Informatics

# Aligning Computer Science Educational Outcomes with Industry Needs Exemplified by a College in Rural India

DIPLOMARBEIT

zur Erlangung des akademischen Grades

**Diplom-Ingenieur**

im Rahmen des Studiums

**Wirtschaftsinformatik**

eingereicht von

**Johannes Hufnagl, BSc**
Matrikelnummer 11918516

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Thomas Grechenig
Mitwirkung: Univ. Ass. Dipl.Ing. Paul Spiesberger, BSc

Wien, 23. Jänner 2026

Unterschrift Verfasser | Unterschrift Betreuung

TU WIEN Informatics

# Aligning Computer Science Educational Outcomes with Industry Needs Exemplified by a College in Rural India

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

**Diplom-Ingenieur**

in

**Business Informatics**

by

**Johannes Hufnagl, BSc**
Registration Number 11918516

to the Faculty of Informatics

at the TU Wien

Advisor: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Thomas Grechenig
Assistance: Univ. Ass. Dipl.Ing. Paul Spiesberger, BSc

Vienna, 23rd January, 2026

Signature Author

Signature Advisor

# Aligning Computer Science Educational Outcomes with Industry Needs Exemplified by a College in Rural India

DIPLOMARBEIT

zur Erlangung des akademischen Grades

**Diplom-Ingenieur**

im Rahmen des Studiums

**Wirtschaftsinformatik**

eingereicht von

**Johannes Hufnagl, BSc**
Matrikelnummer 11918516

ausgeführt am
Institut für Information Systems Engineering
Forschungsbereich Business Informatics
Forschungsgruppe Industrielle Software
der Fakultät für Informatik der Technischen Universität Wien

**Betreuung**: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Thomas Grechenig

Wien, 23. Jänner 2026

# Erklärung zur Verfassung der Arbeit

Johannes Hufnagl, BSc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.
Ich erkläre weiters, dass ich mich generativer KI-Tools lediglich als Hilfsmittel bedient habe und in der vorliegenden Arbeit mein gestalterischer Einfluss überwiegt. Im Anhang „Übersicht verwendeter Hilfsmittel" habe ich alle generativen KI-Tools gelistet, die verwendet wurden, und angegeben, wo und wie sie verwendet wurden. Für Textpassagen, die ohne substantielle Änderungen übernommen wurden, habe ich jeweils die von mir formulierten Eingaben (Prompts) und die verwendete IT-Anwendung mit ihrem Produktnamen und Versionsnummer/Datum angegeben.

Wien, 23. Jänner 2026

Johannes Hufnagl

# Acknowledgements

An Paul,
für die herausragende und intensive Betreuung und Geduld, deine ständige Unterstützung sowie dein offenes Ohr für alle Arten von Problemen und natürlich deinem Engagement im ICT4D-Bereich ohne dessen Engagement diese Arbeit nie zustande gekommen wäre.
An Raoul, Raimund, Enno und Konrad
für die richtigen Fragen zum richtigen Zeitpunkt wertvolles Feedback und den Blick auf das Wesentliche.
An Thomas Grechenig,
für die Betreuung dieser Arbeit sowie die Ermöglichung und Aufrechterhaltung der Kooperation mit unseren indischen Partner:innen.

An Gabriela & Johann,
für die Ermöglichung meiner Ausbildung, eure Unterstützung auf allen Wegen und euren unerschütterlichen Rückhalt.
An Franziska,
für ihre stetige Begleitung und Hilfe auf diesem Weg.
Allen Freunden,
die mit ihrer Motivation und dem entscheidenden Anstoß am Ende unverzichtbar waren.
Danke für die wertvollen Auszeiten und gemeinsamen Erlebnisse, die mir wichtige Atempausen verschafft haben.

Von ganzem Herzen: Dankeschön!

# Abstract

Computer science graduates in India—particularly those from rural affiliated colleges—often enter the labor market with a persistent mismatch between academic preparation and industry expectations. While the Indian software sector demands practical software development skills, exposure to real-world workflows, and professional readiness, higher education curricula at many rural institutions remain strongly theory-oriented. This misalignment poses challenges for graduates' employability and leads companies to invest substantial resources in post-graduate training.

A qualitative research design was employed, combining an integrative literature review and a single-case study in India, including participant observation, semi-structured interviews with graduates and employers, and a thematic analysis following Braun & Clarke. The case study of a college in rural Andhra Pradesh provides nuanced insights into local educational structures and their interaction with industry needs. Findings indicate substantial deficits in practical programming, software engineering processes, and soft skills, alongside strengths in theoretical foundations and motivation to learn.

Drawing on these results, this thesis develops a guidance note for computer science faculty at rural higher education institutions in India, offering actionable recommendations for pedagogical strategies, industry collaboration, and the systematic development of practical and employability-related skills. A focus group evaluation confirms the relevance, clarity, and usability of the proposed guidance.

While the study does not seek statistical generalization, it offers transferable insights for similar rural affiliated colleges facing comparable challenges in aligning computer science education with industry needs.

**Keywords:** Computer Science Education, Skill Gap, Employability, Rural India, Curriculum Alignment, ICT4D, Thematic Analysis, Industry Requirements

# Kurzfassung

Diese Diplomarbeit untersucht, wie die Informatikausbildung an staatlich angebundenen Colleges im ländlichen Indien besser an die Kompetenzanforderungen der Softwareindustrie angepasst werden kann. Sie adressiert ein anhaltendes Missverhältnis zwischen akademischer Ausbildung und beruflichen Erwartungen: Während Studierende überwiegend theoretisches Wissen erwerben, verlangen Unternehmen praktische Programmierfertigkeiten, Kenntnisse von Softwareentwicklungsprozessen sowie Problemlösungs-, Kommunikations- und Teamfähigkeiten. Diese Kompetenzlücken beeinträchtigen die Beschäftigungsfähigkeit von Absolvent:innen und führen dazu, dass Unternehmen erhebliche Ressourcen in nachgelagerte Trainings investieren müssen.

Die Arbeit folgt einem qualitativen Forschungsdesign und kombiniert eine integrative Literaturanalyse mit einer Single-Case-Studie in Indien. Die empirische Datenerhebung umfasste teilnehmende Beobachtung, semi-strukturierte Interviews mit Absolvent:innen und Arbeitgebern sowie eine thematische Analyse. Die Fallstudie eines universitären Partner-Colleges in ländlichem Andhra Pradesh ermöglicht differenzierte Einblicke in lokale Bildungsstrukturen und deren Wechselwirkung mit den Anforderungen der Softwareindustrie. Die Ergebnisse zeigen deutliche Defizite in der praktischen Programmierung, in Softwareentwicklungsprozessen und in Soft Skills, zugleich aber auch Stärken in der theoretischen Ausbildung und in der Lernmotivation der Studierenden.

Auf Basis dieser Erkenntnisse entwickelt die Arbeit praxisorientierte Handlungsempfehlungen für Lehrende und Trainer:innen, die umsetzbare Vorschläge zu didaktischen Strategien, zur Zusammenarbeit mit der Industrie sowie zur systematischen Förderung praktischer und beschäftigungsrelevanter Kompetenzen enthalten. Eine Fokusgruppenevaluierung bestätigt die Relevanz, Verständlichkeit und Anwendbarkeit der vorgeschlagenen Empfehlungen.

Obwohl die Studie keine statistische Generalisierung anstrebt, liefert sie übertragbare Erkenntnisse für vergleichbare universitäre Colleges im ländlichen Raum, die vor ähnlichen Herausforderungen bei der Angleichung der Informatikausbildung an industriebezogene Anforderungen stehen.

**Keywords:** Informatikausbildung, Kompetenzlücke, Beschäftigungsfähigkeit, Ländliches Indien, Lehrplandesign, ICT4D, Thematische Analyse, Anforderungen der Industrie

# Contents

# CHAPTER 1

# Introduction

This chapter discusses the background, motivation, and objectives of this study. It outlines the relevance of aligning computer science education with industry expectations and situates the research within the broader context of educational development in rural India. Furthermore, it defines the problem, specifies the research questions, and summarizes the structure of the thesis to provide an overview of its logical progression.

## 1.1 Problem Description

In 2024, according to the United Nations (UN)[1], 1.1 billion people in 110 observed countries lived in multidimensional poverty, which is calculated considering health, education and the standard of living [1]. The Multidimensional Poverty Index (MPI) is only one of many ways to measure the ongoing inequality in our world. Another key indicator is the number of people living below $1.90 a day, which in 2018 were 8.6% of the world's population. In the last decade, this number was constantly declining and was projected to be at an all-time low of 7.8% in 2021. Due to the COVID-19 pandemic and other crises around the world, the projection for 2022 was to still be at 8.6% [2]. The UN reports such development statistics regularly to track the evolution of their ambitious Sustainable Development Goal (SDG) 1, which aims to end poverty around the globe by 2030. There are currently 17 SDGs with SDG 4 targeting the quality of education: "Ensure inclusive and equitable quality education and promote lifelong learning opportunities for all" [3]. SDG 8 wants to "Promote sustained, inclusive and sustainable economic growth, full and productive employment and decent work for all" which this thesis intends to contribute to.

Countries in the Global South face this challenge and India in particular. While it is experiencing significant economic growth, India still faces stark inequalities in wealth

[1] https://www.un.org/

distribution [4], [5]. Education is seen as a key mechanism for redistributing wealth and promoting equitable development [6], [7], [8]. However, the country faces significant challenges in providing quality education, especially in rural areas [9], [10].

Computer science education in rural India is facing diverse challenges. One outstanding is that graduates are rarely equipped with the skills the private sector desires. In their 4 years of studying, students receive a rather theoretical education and lack practical experiences [11]. After graduation, the private sector trains graduates to become hands-on engineers to complete their education. This results in avoidable costs for the general public as well as for companies [11].

This work is situated at the intersection of software engineering education and empirical software engineering research, with a particular focus on the alignment of university curricula with industry requirements. From a software engineering research perspective, the study is framed as a single case study, which is a well-established empirical research strategy for investigating contemporary phenomena in their real-world context [12]. According to Runeson et al., case study research in software engineering is particularly suitable when the boundaries between a phenomenon and its context are not clearly evident and when multiple sources of evidence are required to gain an in-depth understanding. Accordingly, this study investigates the alignment between computer science educational outcomes and private-sector software industry requirements within a specific institutional and regional context. The empirical focus lies on computer science education at a college in rural India, taking into account the structural, educational, and socio-economic conditions of the region. By applying a single case study design, the research allows for a detailed and context-sensitive analysis of educational practices, graduate skill development, and industry expectations, thereby contributing empirically grounded insights to the field of software engineering education.

To sum up, this master thesis focuses on the following problem statement: *A persistent mismatch exists between the skills taught in rural Indian higher education and those required by the software industry. Prior research identified incomplete student skills, inadequate teaching practices, and limited curriculum evaluation, leading to low employability and costly post-graduate training. This study examines how computer science education at a partner institution can be better aligned with industry needs to enhance graduates' professional readiness.*

Therefore, this master thesis contributes to the knowledge of how to improve computer science education in rural areas in India, more specifically, what requirements do hiring software companies expect from graduates and what the training is like. In semi-structured interviews, both the perspective of graduates and the perspective of companies will be examined and then analyzed by using a thematic analysis by Braun and Clarke to then draw conclusions.

## 1.2 Motivation

The author's interest in computer science originated at an early age and has since developed into a profound dedication to leveraging technology as a means of addressing urgent social and educational issues. Currently pursuing a master's degree in business informatics at the Vienna University of Technology, the author has conducted research in the field of Information and Communications Technology for Development (ICT4D) in his undergraduate thesis. This study addresses a critical gap between the theoretical training provided by computer science programs in rural parts of India and the practical skills required by private sector employers [14], [15]. By exploring this mismatch, the research contributes to the field of educational research with a focus on curriculum development and industry-academia collaboration. The findings will inform policy makers, educators and industry leaders on how to better align educational outcomes with industry needs, potentially leading to improved employability and economic development in rural areas.

Additionally, the author's professional aspirations are informed by a keen interest in leveraging technology to facilitate social change. As a member of ICT4D.at[2], an organization dedicated to leveraging technology for human development, the author has gained valuable insights into the potential of technical solutions to empower communities. The organization's mission to equip individuals with the requisite skills to achieve their full potential is closely aligned with this research, which aims to generate actionable insights for improving computer science education in rural areas [16]. The objective of this thesis is to contribute to the ongoing dialogue between policymakers, educators, and industry leaders with a view to proposing sustainable, practical solutions that enhance employability and foster economic development.

## 1.3 Expected Results

In 2015, the United Nations set forth a key target, 4.3, within SDG 4: "By 2030, ensure equal access for all women and men to affordable and quality technical, vocational, and tertiary education, including university" [17]. Additionally, Target 4.4 emphasizes the importance of equipping individuals with employable skills: "By 2030, substantially increase the number of youth and adults who have relevant skills, including technical and vocational skills, for employment, decent jobs and entrepreneurship". Its associated indicator (4.4.1) measures the "proportion of youth and adults with Information and Communications Technology (ICT) skills[...]" [17]. Similarly, SDG 8 includes target 8.6: "By 2020, substantially reduce the proportion of youth not in employment, education or training" [18]. This master's thesis contributes to the advancement of these goals by providing recommendations to enhance computer science education in rural India. By aligning education outcomes with the needs of the private sector, the research aims to improve access to quality education, increase the employability of graduates, and support sustainable economic growth.

[2] https://www.ict4d.at/

This thesis aims to examine how computer science education in rural India can be improved, with a particular emphasis on identifying strategies that educators can employ to develop sustainable and industry-relevant courses. Specific consideration will be given to the alignment of the curriculum with the skills required by the private sector, with the objective of ensuring that students are adequately prepared for future employment. Furthermore, the thesis will address the distinctive challenges associated with teaching across cultural and infrastructural divides, offering practical guidance for educators to surmount these obstacles and facilitate long-term enhancements in educational outcomes.

This thesis aims to investigate the discrepancy between the skills taught to computer science graduates in rural universities and the requirements of the private sector, with the objective of enhancing employability. The research will concentrate on identifying the particular deficiencies in both hard and soft skills and will offer practical recommendations for addressing these shortcomings. By providing comprehensive guidance, the thesis will assist universities in aligning their computer science curricula with industry requirements, thereby enhancing the preparedness of graduates for the workforce.

With the problem statement from Section 1.1 in mind, this thesis aims to address the following Research Question (RQ):

### 1.3.1 Research Questions

**RQ1** What hard and soft skills do private sector software companies in India require from computer science graduates of rural colleges, exemplified by the region of Andhra Pradesh?

**RQ2** Do the skills of computer science graduates align with the requirements of the private sector in rural India, and if not, what specific skill gaps and skill matches can be identified?

**RQ3** What specific recommendations can help higher education institutions (HEIs) in rural India implement computer science curricula that effectively address the skill gaps identified in RQ2, while also enhancing the identified skill matches?

### 1.3.2 Phases of Answering the Research Questions

The following phases outline the overall research process used to address the research questions. Each phase is grounded in established methodological approaches. Detailed descriptions of the underlying methods, including literature review strategy, interview design and data analysis procedures, are provided in Section 1.4.

**Integrative Literature Review:**

- Identify frameworks and previous research that address the education-employment gap.

- Review existing research on the skills required by the private sector and the current state of computer science education in rural India.

**Action Research:**

- Conduct qualitative, semi-structured interviews and expert interviews with stakeholders, including graduates and employers.
- Observe the situation in the Indian college, as well as investigate private sector companies.

**Data Analysis:**

- Thematically analyze interview transcripts to identify recurring themes and patterns, using a thematic analysis.
- Analyze survey data to quantify the degree of alignment or misalignment between education and industry needs.

**Results and Recommendations:**

- Synthesize findings to identify skill gaps and skill matches as a list.
- Develop guidance notes for college professors/training staff, curricula designers and students to guide them on what is expected from them.
- Evaluate the guidance notes with expert interview.

### 1.3.3 Artifacts

This theses will produce the following results as artifacts:

1. List of skill gaps and matches between graduates and industry
2. guidance notes for Higher Education Institutions (HEIs) faculty and trainers

## 1.4 Methodology

The methodology of this study follows a qualitative-dominant empirical research design grounded in a single case study approach, as established in software engineering research [12]. Multiple sources of evidence (e.g., Integrative Literature Review (ILR) findings, interviews, participant observation, and documents) are combined to enable methodological triangulation and to gain an in-depth understanding of the investigated phenomenon in its real-world context [12].

### 1.4.1 Single Case Study Design

Following Runeson et al., the study is designed as a single case study investigating a contemporary phenomenon in its real-world context. The case is the partner college in rural India, and the unit of analysis is the alignment between educational outcomes in computer science education and private-sector industry requirements. Multiple sources of evidence will be used to support triangulation and to strengthen the credibility of the findings [12].

### 1.4.2 Overview of the Case Study Design

- Integrative Literature Review (ILR)
- Action Research (within the case)
- Participant Observation
- Qualitative Semi-Structured Interviews
  - Employers / training staff
    - Companies employing graduates from the partner college
  - Graduates employed in the software industry (female and male)
- Thematic Analysis following Braun and Clarke
- Development and evaluation of guidance notes (focus group)

### 1.4.3 Theoretical Approach - Integrative Literature Review

Torraco's articles introduce the integrative literature review as a unique form of research that synthesizes existing literature to generate new knowledge [19], [20]. Torraco outlines the key components of an integrative literature review, which include reviewing, critiquing, and synthesizing literature to resolve inconsistencies and provide new perspectives [19], [20]. An integrative literature review provides a comprehensive understanding of a research topic and identifies gaps and inconsistencies in the existing literature. They are particularly useful for dynamic or emerging topics where a holistic view is required. By critically analyzing and synthesizing evidence, integrative reviews contribute to the development of new frameworks and guide future research directions. They also inform policy and practice by synthesizing current knowledge and suggesting practical applications. Overall, integrative reviews advance knowledge by providing new perspectives and resolving inconsistencies in the literature [19], [20].

Conducting an integrative literature review involves several key steps. First, the research question must be clearly defined and justified as appropriate for an integrative review. The objectives and scope of the review should be outlined, with specific inclusion and

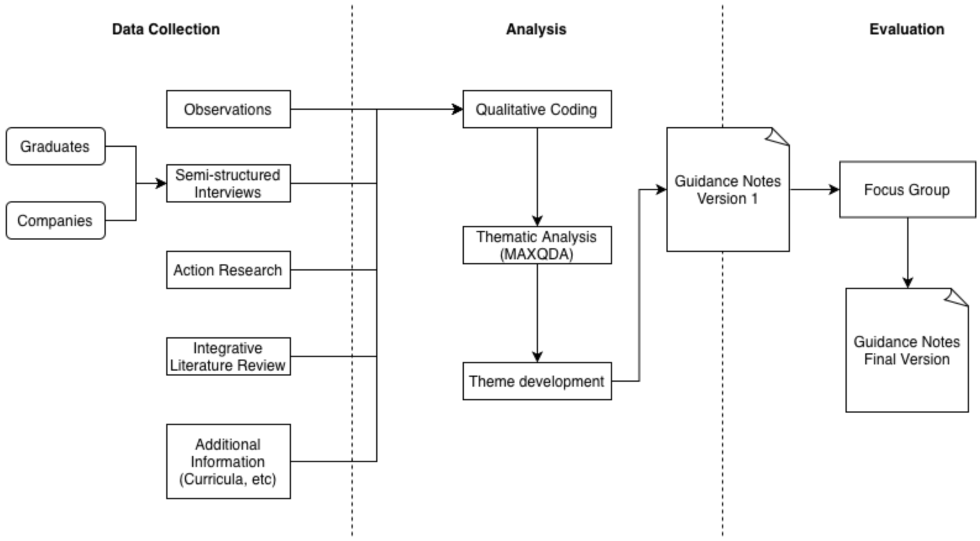


Figure 1.1: Research design of this thesis and how data collection feeds the analysis leading to guidance notes

exclusion criteria. This is followed by a comprehensive search strategy using relevant keywords and databases, with systematic documentation of the search process. Relevant studies are then screened and selected on the basis of their titles, abstracts, and full texts. Data extraction involves the collection of key information from each study, which is then critically appraised for quality and relevance. Synthesis integrates findings from multiple studies to develop new knowledge and conceptual frameworks. Finally, the review is written, organized into clear sections, and revised for clarity and academic rigor [19], [20].

Appropriate research questions for an integrative literature review often focus on identifying the extent of existing research on a topic, the consistency of findings across studies, and the gaps or inconsistencies that need to be addressed. Accordingly, the integrative literature review was guided by a clearly defined set of research-oriented questions focusing on (1) the skills and competencies expected from computer science graduates by the software industry, (2) the extent to which existing curricula in rural India address these expectations, and (3) the gaps and inconsistencies identified in prior empirical and conceptual studies [19], [20]. The full set of guiding review questions and selection criteria is documented in Section 3.

### 1.4.4 Empirical Approach – Action Research

Within the single case study design, this thesis applies Action Research as a primary empirical method to iteratively investigate and address the identified practice-oriented problems [21]. Action research is a well-established methodology that combines systematic inquiry with practical intervention, aiming to generate scientific knowledge while simultaneously improving real-world practice. It is rooted in a post-positivist research philosophy and emphasizes participatory collaboration, iterative learning, and problem-driven research cycles.

In contrast to purely observational fieldwork, action research explicitly integrates researchers and practitioners as co-creators of knowledge. According to Petersen et al., action research is particularly suitable for industry–academia collaboration in software engineering contexts, as it enables close interaction with stakeholders, supports iterative refinement of interventions, and produces outcomes that are both empirically grounded and practically relevant. The methodology follows recurring cycles of diagnosing a problem, planning action, taking action, and reflecting on the outcomes, thereby allowing continuous learning and adjustment.

Within this thesis, action research is operationalized through close engagement with a partner college in rural India, recent computer science graduates, and representatives from the private software industry. The initial diagnostic phase involved identifying skill gaps and mismatches through an integrative literature review, semi-structured interviews, and participatory observations. Based on these insights, an initial intervention in the form of *guidance notes* for higher education institutions was developed, aiming to support educators in aligning curricula with industry needs.

Following the principles outlined by Petersen et al., the intervention was not treated as a final artifact but as an evolving outcome. The guidance notes were evaluated through expert feedback and focus group discussions, leading to refinements across iterative cycles. This reflective phase was essential for validating assumptions, incorporating practitioner perspectives, and improving the applicability of the proposed recommendations.

By adopting an action research approach, this study goes beyond descriptive fieldwork and positions the researcher as an active participant in the change process. This methodology strengthens the practical relevance of the findings while maintaining methodological rigor, as recommended for applied research in software engineering education and industry collaboration [21].

### 1.4.5 Applied Research Approaches

This section outlines the applied research approaches adopted in this study to investigate the alignment between computer science education and industry requirements. It introduces the methodological perspective guiding the empirical phases described in the subsequent subsections.

#### Qualitative Semi-Structured Interviews

Semi-structured interviews with companies and Computer Science (CS) students of the partner institution will be conducted to learn more about their skill requirements and the skill matches and gaps of graduates.

The author selected Semi-Structured Interviews as a suitable interview type for this section of the thesis because of the critical importance of developing a personal connection to the subjects described by Saunders and the exploratory nature of the interviews [22]. The goal is to interview at least three graduates working in the CS sector and different companies.

Saunders et al., Wilde and Hess describe the semi-structured interview as a conversational approach, conducting interviews with one respondent at a time [23], [24]. It utilizes a combination of closed- and open-ended questions, often accompanied by follow-up questions such as "why" or "how". This allows for a more flexible and dynamic dialogue, which may deviate from the topics on the agenda. This approach differs from the standardized survey, which tends to adhere more strictly to a set of predetermined questions. The Semi-Structured Interviews may also explore issues that were not anticipated during the planning stages [23], [24]. The semi-structured interviews will be taped, transcribed, and subjected to a thematic analysis (see 6.1) [13].

### Observations

Observation, particularly participant observation, is a fundamental method for collecting qualitative data about people, processes, and cultures [24]. According to Kawulich, participant observation allows researchers to immerse themselves in the natural setting of their study subjects, providing a rich, contextual understanding of the observed phenomena [24], [25]. This method involves the researcher taking on a dual role as both participant and observer, enabling them to gather data through direct engagement in the activities of the participants.

The observations in this study will be both participatory and uncontrolled. The participatory approach involves the researcher engaging in the daily activities of the study subjects, while the uncontrolled nature of the observations ensures that the behaviors and interactions observed are natural and unaltered by the presence of the researcher. This methodology aligns with the approaches outlined by Petersen et al., who emphasized the importance of observing natural behaviors and interactions [21], see Section 5.2.

### Thematic Analysis

Thematic Analysis is a qualitative research method that is widely recognized for its flexibility and simplicity, making it particularly suitable for novice researchers. According to Braun and Clarke, thematic analysis provides an accessible and systematic technique for identifying, analyzing, and reporting patterns (themes) within data. This method is also appreciated for its collaborative approach, where participants are actively engaged as co-researchers, contributing valuable insights throughout the research process [13].

In the thesis, semi-structured interviews and observations will be analyzed using thematic analysis. The guidelines provided by Braun and Clarke will be followed to ensure a rigorous and systematic application of this technique. Therefore, the software tool such as MAXQDA[3] or QualCoder[4] will be used. They are robust and versatile programs for

[3] https://www.maxqda.com/
[4] https://qualcoder.wordpress.com/

computer-assisted qualitative and mixed-methods data analysis suitable for academic, scientific, and business institutions. They facilitate the systematic analysis of text, multimedia, and other qualitative data by providing a set of tools for coding, categorizing, and interpreting data. The software supports comprehensive data management and visualization, allowing researchers to efficiently identify patterns, themes, and relationships within the data. By leveraging their advanced analytical capabilities, this study will ensure a thorough and rigorous examination of qualitative data collected through interviews and observations, enhancing the validity and reliability of the findings.

### 1.4.6 Development of Guidance Notes

In the final phase of this research, the empirically derived recommendations from **RQ3** will be consolidated into a structured set of *guidance notes* aimed at faculty and trainers at rural higher education institutions.

Following principles of applied qualitative research and educational design research, the purpose of these guidance notes is to translate analytical findings into concise, actionable, and context-sensitive recommendations that can inform curriculum implementation and pedagogical practice [13], [23].

The use of guidance notes as a research output is well established in the field of ICT4D and education policy research, where complex empirical insights are systematically transformed into practitioner-oriented guidance. For example, the ICT4D Collective[5] (formerly known as UNESCO Chair in ICT4D) published the report *“Technology and Education for the Most Marginalised Post-COVID-19”*, which explicitly operationalizes research findings through a series of short, thematically structured guidance notes addressing key implementation challenges faced by governments and educational institutions [26]. This research project was funded by the UK government under the leadership of the Foreign, Commonwealth & Development Office[6] (formally known as Department for International Development) [26].

These guidance notes are designed to be concise, practical, and directly usable by decision-makers, while remaining grounded in rigorous qualitative and consultative research processes. Adopting a similar approach, the guidance notes developed in this thesis aim to support faculty in rural Indian HEIs by providing actionable recommendations that are empirically grounded, context-aware, and aligned with industry requirements.

Each guidance will summarize a problem, outline a recommended intervention, describe feasible implementation strategies, and indicate the expected impact. Should the subsequent evaluation reveal missing perspectives or identify additional categories not covered in the thematic analysis, these insights will be discussed as part of the *Future Work* (see Section 8.2).

[5] https://ict4d.org.uk/

[6] https://www.gov.uk/government/organisations/foreign-commonwealth-development-office

### 1.4.7 Evaluation: Focus Group Discussion

After developing the guidance notes (see Section 7.7), a focus group discussion will be conducted to validate their practical applicability and relevance. This evaluation step will complement the primary data collection and will serve as a quality assurance mechanism to assess whether the proposed recommendations are understandable, feasible, and aligned with the institutional realities of rural higher education.

**Purpose and Rationale.** The focus group will aim to evaluate the clarity, feasibility, and perceived usefulness of the guidance notes derived from the thematic analysis of graduate and employer interviews. While the initial interviews provided empirical insights into the alignment of graduate skills with industry needs, this follow-up evaluation will examine whether the synthesized recommendations reflect practical constraints and can effectively inform institutional decision-making.

**Participants and Sampling.** The focus group will consist of four to five experts with extensive experience in ICT4D, capacity development, and computer science education. Participants will be selected purposively based on their domain expertise and familiarity with educational interventions in low-resource contexts.

**Focus Group Design.** Following the methodological guidelines of Morgan [27], the focus group will follow a discussion format that balances comparability with flexibility. The session will be guided by prompts aligned with three core evaluation dimensions:

1. **Clarity and Comprehensibility:** Are the guidance notes articulated in a way that faculty can easily understand and apply them?
2. **Feasibility and Applicability:** Are the proposed measures realistic within the infrastructural, administrative, and pedagogical constraints of rural HEIs?
3. **Scalability and Transferability:** Can the recommendations be adapted for other affiliated colleges or comparable educational settings?

**Pre-Session Preparation.** To support informed reflection, all participants will receive the complete set of guidance notes several days prior to the focus group discussion. The material will be accompanied by a short briefing document outlining the study context, the purpose of the evaluation, and the three evaluation dimensions. This preparation will enable participants to provide deeper reflections and more targeted feedback during the session.

**Data Collection and Analysis.** The focus group will be conducted on-site at a TU Wien facility. Detailed notes will be taken during the discussion to capture all relevant insights. The collected feedback will be used directly to revise and refine the guidance notes. This evaluative step serves to ensure that the final version of the guidance notes is clear, feasible, and aligned with the practical realities of rural higher education.

If the evaluation introduces perspectives not captured in the original coding, these will be documented as potential directions for future research (see Section 8.2). The validated insights will then be integrated into the final discussion to strengthen the practical relevance and credibility of the study's outcomes.

## 1.5 Limitations

The following restrictions are defined to distinguish this thesis from similar work and to stay within the boundaries of a master's thesis:

1. The outcomes of this study are founded on the analysis of a single curriculum design implemented at a rural Indian college. While this curriculum is centrally prescribed and applied across more than 80 affiliated institutions, the empirical investigation was limited to the college level and did not include direct access to the affiliating university's internal resources or curriculum development processes. Accordingly, the findings remain context-specific and do not allow for statistical generalization.

2. While the findings from this research offer insights into rural education challenges, they are specific to the region of Andhra Pradesh and may not be generalizable to all rural areas in India or other developing countries due to variations in socio-economic and educational conditions.

3. This thesis is specifically focused on the field of computer science education in rural India. It should be noted that the findings may not be directly applicable to other academic disciplines, which may face different challenges related to education and employability.

4. Industry skill requirements were derived from a limited number of employer interviews. A broader and more diverse set of industry participants could potentially reveal additional or differing expectations regarding graduate employability.

5. Cultural dissimilarities between India and the researcher's country of origin may impact the interpretation of data and participant interactions. Despite the researcher's best efforts to approach the study with cultural sensitivity, it is possible that some nuances may not be fully captured.

6. The time constraints of completing this master's thesis constrained the scope of action research and the depth of analysis. Consequently, certain aspects of the curriculum and its impact on employability may not have been fully explored.

7. The graduate sample includes only participants who successfully obtained employment in the software industry. Graduates who did not secure a job after completing their studies could not be reached during action research, and their perspectives are therefore absent from the dataset. This introduces a positive selection bias and limits the study's ability to capture employability challenges experienced by unsuccessful graduates.

## 1.6 Structure

This thesis is structured into eight main chapters, each addressing a distinct phase of the research process while maintaining a coherent logical flow.

**Chapter 1: Introduction** outlines the research motivation, objectives, and relevance of the study. It introduces the central research problem—the skill mismatch between computer science education and industry requirements in rural India—and formulates the guiding research questions. It also summarizes the methodological approach and defines the scope, limitations, and ethical considerations of the study.

**Chapter 2: Related Work** reviews existing literature on the employability gap in computer science education, emphasizing both global and Indian rural contexts. It integrates findings from prior studies conducted within the research collaboration between the Industrial Software (INSO) group and the partner college, as well as related international research on industry–academia collaboration.

**Chapter 3: Integrative Literature Review (ILR)** synthesizes previous academic work on skill development, curriculum design, employability, and higher education in developing countries. Following the principles of Torraco and Whittemore and Knafl, it identifies conceptual patterns and research gaps that inform this study's analytical framework [20], [28].

**Chapter 4: Case Background** situates the research within the socio-economic and institutional setting of rural India. It describes the structure of the Indian higher education system, the affiliating university model, and the characteristics of the partner college, providing the contextual background necessary for interpreting the empirical findings.

**Chapter 5: Case Study Design** explains the research design and action research procedures. It details the process of conducting semi-structured interviews, on-site observations, and related communication activities during the collaboration with the partner college.

**Chapter 6: Case Analysis** outlines the analytical approach based on Braun and Clarke's six-phase model of thematic analysis [13]. It describes the use of software tools such as MAXQDA for coding and categorization and presents the derived themes that address the research questions.

**Chapter 7: Results and Discussion** integrates the empirical findings from graduates, employers, and observations. It discusses the (mis)alignment between educational outcomes and industry expectations, supported by thematic evidence and cross-references to the ILR. The chapter also evaluates the developed *guidance notes* through an expert interview, assessing their clarity, feasibility, and transferability.

**Chapter 8: Conclusion and Future Work** summarizes the key findings, theoretical and practical implications, and contributions of this research. It concludes with reflections on methodological limitations and outlines avenues for future work in curriculum development and ICT4D-oriented educational research.

A list of references, web sources, and appendices—including interview guides, coding schemes, an example of the transcribed interviews and the complete ILR—complement the main chapters and ensure transparency and reproducibility.

## 1.7 Relevance to the Curriculum

The scientific relevance of this thesis lies in its focus on addressing the persistent discrepancy between the skills taught in computer science programs at rural universities and the actual demands of the private sector. By meticulously delineating the deficiencies in both hard and soft skills that impede graduate employability, this research establishes a foundation for educational reform that is firmly rooted in the actual demands of the industry. Moreover, this study contributes to the broader discourse on improving education in underdeveloped regions, specifically within the context of emerging economies like India, where education is a key mechanism for reducing inequality and promoting sustainable development. The practical recommendations generated for aligning curricula with industrial expectations advance academic understanding and offer tangible solutions that can inform policy-making and curriculum design. Ultimately, this work supports global development efforts, particularly those outlined in the United Nations' SDG 4 on quality education and SDG 8 on decent work and economic growth. It does so by fostering an education system that equips students with the relevant skills for modern labor markets.

### 1.7.1 Relevance to the Business Informatics Curricula

The Business Informatics curricula at the TU Wien state that it qualifies its graduates to work in consulting companies and the public sector. Furthermore, Business Informatics is a multidisciplinary scientific discipline with a focus on the socioeconomic context. The University Act of 2002 also states in § 1 that "Universities are called upon to serve scientific research and teaching, the development and advancement of the arts and the teaching of the arts, and thus to contribute responsibly to the solution of human problems and to the successful development of society and the natural environment." Education is a key factor to raise a generation of engineers to solve humanity's problems and tackle poverty around the globe, which aligns with the SDGs stated by the UN. In the past, the author of this work has been engaged in educational programs in India and has been a member of the Non-Governmental Organization (NGO) ICT4D.at[7], which has the goal to use knowledge and technology to empower people.

### 1.7.2 Ethical Considerations

Ethical considerations are a critical component of this research to ensure integrity and respect for all participants involved. When conducting action research and collecting data through semi-structured interviews and surveys in rural India, it is essential to obtain informed consent from all participants by clearly explaining the purpose of the research, the procedures involved, and their right to withdraw at any time without penalty. Ensuring confidentiality and anonymity of participants is paramount, personal identifiers are removed or anonymized in all reports and publications. In addition, sensitivity to cultural norms and practices will be maintained to build trust and rapport with local communities. The potential power dynamics between the researcher and participants will be carefully managed to avoid any form of coercion or undue influence. All data will be stored securely to protect the privacy and rights of the participants. Furthermore, it is imperative to approach this research with a sense of colonial responsibility, ensuring that the work is not conducted from a position of superiority but rather as a service to the community. The research will be demand-driven, guided by the needs and interests of the local participants, ensuring that the outcomes are beneficial to them and that their voices and perspectives are central to the research process. These ethical considerations are vital to uphold the credibility and ethical integrity of the research.



[7] https://www.ict4d.at/

CHAPTER 2

# Related Work

This chapter presents the theoretical and conceptual foundations that inform this research. It introduces key concepts in computer science education, curriculum design, employability, and pedagogical theory, and situates them within the context of developing and rural higher education systems. These fundamentals provide the analytical basis for understanding how curricular structures and teaching practices influence the alignment between educational outcomes and industry needs.

## 2.1 Previous Educational Research in Rural India

This thesis contributes to the research collaboration between the INSO research group and the partner college, which started in 2017. In the past, two research papers were published during this collaboration and the author has written his bachelor thesis in this context [14], [29]. One being Spiesberger et al.'s research paper, a Design-Reality Gap Analysis (DRGA) was used to identify the main gaps between the curriculum design by an Indian university and its implementation in one of the affiliated university's partner colleges in Andhra Pradesh (AP). Their work's results are summarized within five gaps [14]:

**Gap 1:** Incomplete students skills

**Gap 2:** Unmet employability goal

**Gap 3:** Improperly implemented teaching methods

**Gap 4:** Missing curriculum evaluations

**Gap 5:** Inadequate professors skill

Spiesberger et al. also identified three major challenges that students and professors face while implementing the curriculum. They differ from gaps, as they have no design counterpart in the curriculum:

**Challenge 1:** Difficulties in self-learning

**Challenge 2:** Excessive student workload

**Challenge 3:** Ever-changing course assignments

In the previous research by Spiesberger et al. leading to the paper *Identifying Higher Software Engineering Education's Design-Reality Gaps in Rural India*, Buerstmayr gathered data, with students and teachers participating in semi-structured interviews and observations [30]. The OPTIMISM dimension was used to thematically evaluate the collected data [31]. In his thesis Buerstmayr, gave a summary of the abilities that (software) engineering graduates are lacking or have insufficient skills according to the software industry [30]:

- Communication skills (oral & written)
- Ability to analyze and solve problems
- Collaboration, teamwork and social skills
- Reliability
- Willingness to learn, take new directions
- Technical skills
- Use of modern (software) tools

In their paper, „Teaching Global Software Engineering in a Remote Customer Environment“ Vallon et al. describe a course designed to emulate the conditions encountered in the real world of Global Software Engineering. The objective of this course, which was developed in collaboration between a European university and an Indian college, was to address the key challenges inherent to Global Software Engineering, including communication, cooperation, and coordination. European students assumed the role of developers, while Indian students acted as remote customers, introducing authentic cultural, linguistic, and temporal obstacles [29]. The for this research relevant results and insights by Vallon et al. are summarized as follows:

**Methodology and Challenges** The course by Vallon et al. employed an agile methodology, incorporating techniques such as Scrum and Extreme Programming (XP). The Indian students were responsible for formulating the software requirements, while the European students were tasked with the development of the projects. Significant communication barriers emerged due to language differences, cultural differences in conflict avoidance, and coordination issues resulting from time zone differences and a lack of transparency [29].

Despite the challenges encountered, the experience was highly valued by the students. The Indian students expressed appreciation for the opportunity to enhance their communication abilities and gain insight into the requirements process. However, they indicated a preference for involvement in the development phase. The European students were able to gain an appreciation for the realistic customer interaction [29].

It was recommended that the course duration be extended, direct communication be increased, and Indian students be involved in development roles in order to enhance their practical skills. The aforementioned adjustments are designed to better prepare students for the modern distributed workplace [29].

## 2.2 Employability of Indian Graduates

The employability crisis among engineering graduates in India can be illustrated by a case study from the Indian region of Tamil Nadu. This study highlights the critical skills gap between the training provided by academic institutions and the needs of industry. Many graduates lack the practical skills and industry readiness needed to succeed in the workforce [32].

In Tamil Nadu, India, several initiatives have been taken to address this issue. These include incorporating internships, live projects and industry interactions into the curriculum. The emphasis on Outcome-Based Education (OBE) ensures that students not only acquire theoretical knowledge, but also develop practical skills that are directly applicable to industry requirements. This approach has significantly improved the employability and work readiness of graduates, leading to higher placement rates and better career prospects [32].

The study concludes that these experiential learning initiatives have a profound impact on student outcomes. By providing students with real-world experience and opportunities to apply their knowledge, these programs help bridge the gap between academia and industry. The success of these initiatives in Tamil Nadu serves as a model for other regions seeking to improve the employability of their engineering graduates [32].

The employability of India's engineering graduates remains a significant problem, with only 25% deemed employable by the Indian National Association of Software and Services Companies (NASSCOM). Nair study explores innovative ways to bridge this gap through experiential learning and competitive programming [11].

The Live-in-Labs program at Amrita Vishwa Vidyapeetham[1] is highlighted as a key initiative. This program is hands-on and student-centered, allowing students to apply their engineering knowledge in real-life village settings. It emphasizes human-centered

[1] https://www.amrita.edu

design and sustainable development, and develops both technical and soft skills such as teamwork and communication. Students engage directly with community issues and translate theoretical concepts into practical solutions [11].

Competitive programming platforms such as CodeChef[2], HackerRank[3], and the International Collegiate Programming Contest (ICPC) are also essential. These platforms provide a competitive environment that sharpens coding and problem-solving skills. Graduates who participate in these initiatives often secure high-paying jobs at leading technology companies [11].

The Live-in-Labs program has led to innovative solutions for rural communities, and each project results in a student-published paper. These initiatives significantly enhance the employability of graduates and prepare them to meet industry needs [11].

In conclusion, the integration of experiential learning and competitive programming into the curriculum is critical to improving employability. These initiatives ensure that graduates have the skills necessary to succeed in the technology industry [11].

Given the large amount of relevant literature found, a systematic and integrative approach was chosen to consolidate and critically evaluate these findings. The adoption of an ILR (see Section 3) enables this thesis to synthesize evidence from multiple studies and contexts, offering a comprehensive overview of existing knowledge on employability, curriculum alignment, and skill development in engineering education.

[2] https://www.codechef.com
[3] https://www.hackerrank.com

CHAPTER 3

# Integrative Literature Review

This study employed an ILR approach to establish the conceptual foundation for examining the alignment between computer science education and industry expectations. The ILR serves as the theoretical backbone that situates the field study findings within global evidence on skill gaps, curriculum alignment, and employability. An ILR enables the integration and synthesis of both empirical and theoretical contributions, incorporating diverse methodologies to develop a comprehensive and contextually grounded understanding of the research phenomenon [20], [28], [33]. Compared to systematic reviews, which emphasize exhaustive coverage and narrowly defined inclusion criteria, and narrative reviews, which often provide descriptive and less replicable summaries, the integrative approach combines methodological rigor with interpretive flexibility [33]. This duality enables researchers to not only summarize existing knowledge but also to build new conceptual perspectives that link evidence across disciplines and methods [20]. In this study, the ILR therefore functions as more than a preparatory step—it provides a theoretical framework that contextualizes the field study results and informs the subsequent thematic analysis (see 6.1) by connecting global research evidence to the empirical realities of computer science education in rural India. The process of the ILR can be seen in Figure 3.1.

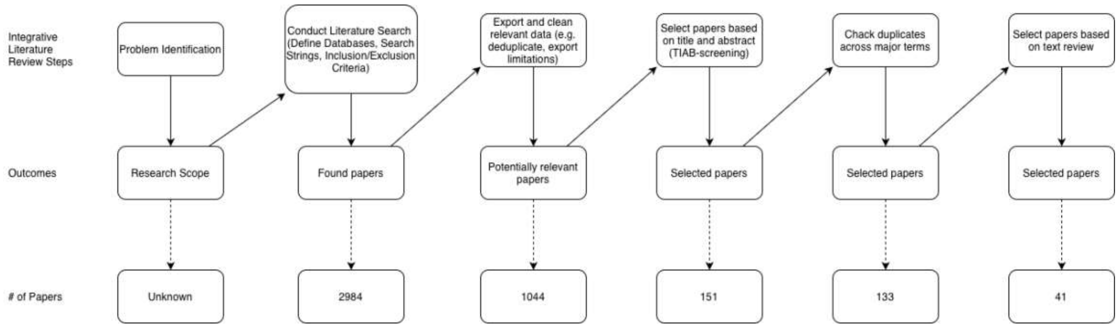


Figure 3.1: Overview of the Integrative Literature Review workflow, including database search, deduplication, and screening steps.

**Five-Stage Review Process.** Following the five-stage model proposed by Whittemore and Knafl, the ILR process consisted of the following steps [28]:

1. **Problem identification** — defining the research purpose and framing the central questions guiding the review.

2. **Literature search** — conducting a structured and transparent search across major academic databases, including the ACM Digital Library[1], IEEE Xplore[2], Scopus[3], and Web of Science[4].

3. **Data evaluation** — assessing the quality, scope, and relevance of the identified studies according to predefined inclusion and exclusion criteria.

4. **Data analysis** — synthesizing and categorizing key insights from the selected works to identify recurring themes and conceptual relationships.

5. **Presentation** — organizing and reporting the findings in a transparent, replicable manner that connects them to the research questions and broader theoretical context.

Search strings were derived from the RQs and combined core terms such as *computer science education*, *industry expectations*, *curriculum development*, and *employability*. Table 3.2 provides an overview of all search queries, filters, and databases used. Where possible, search results were filtered by relevance, otherwise, the default ranking of each database was applied.

[1] https://dl.acm.org
[2] https://ieeexplore.ieee.org
[3] https://www.scopus.com/
[4] https://www.webofscience.com/

**Project Management and Audit Trail.** To ensure methodological transparency and reproducibility, all search, screening, and synthesis activities were systematically documented and monitored within a structured project management framework. This framework provided a clear overview of every stage of the ILR process, ensuring that all decisions and procedures remained traceable throughout the review. Comprehensive records were maintained for each major step, including detailed database queries (see Table 3.2), inclusion and exclusion criteria (see Table 3.1), title and abstract screening logs, and extracted data. In addition, the documentation incorporated Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) counts, which record the number of studies identified, screened, included, and excluded at each stage of the review in accordance with the PRISMA guidelines [34]. Although originally developed for systematic reviews, the PRISMA framework is widely applied in integrative reviews as well, as it enhances clarity and transparency by visualizing the flow of records through the search and selection process [33]. The complete search history—including all queries, applied filters, and record counts before and after deduplication—is summarized in Appendix C.1. This structured documentation forms the audit trail of the ILR, providing verifiable evidence of the rigor and transparency applied during literature selection and synthesis.

**Scope and Eligibility Criteria.** The evaluation focused on research examining **skill gaps**, **curriculum–industry alignment**, and **employability** within higher education in CS and software engineering. In line with the ILR framework, both empirical (qualitative, quantitative, and mixed-methods) and conceptual studies were considered relevant if they contributed to understanding graduate competences, curriculum design, or industry expectations [20], [28]. To ensure a contemporary and comparable evidence base, the scope of the review was limited to peer-reviewed, English-language studies published since 2005, reflecting the modern development of computing curricula and employability frameworks.

**Manual recording of Scopus results** Due to export limitations in the Scopus database, the metadata of the first thirty search results for each query were manually transcribed into structured Excel spreadsheets. Each record included the publication title, author names, year, venue, and—where available—the abstract or summary. These manually compiled tables were integrated into the overall documentation of the ILR, ensuring that all Scopus queries were transparently recorded and verifiable within the review's audit trail. This approach preserved methodological consistency across databases and maintained full traceability of the search process.

## 3.1 Database Coverage and Record Management

Appendix C.1 provides a comprehensive overview of the database search history, including all queries, filters, and export metadata from Scopus, Web of Science, IEEE Xplore, and the ACM Digital Library. The appendix table lists all twenty distinct

| ID | Type | Criterion | Rationale |
|---|---|---|---|
| 1 | Inclusion | Focus on Computer Science or Software Engineering higher education. | Ensures disciplinary relevance to CS/SE. |
| 2 | Inclusion | Addresses at least one dimension: skill gaps, curriculum alignment, employability, or employer expectations. | Captures studies relevant to research questions. |
| 3 | Inclusion | Empirical (qualitative, quantitative, or mixed) or conceptual/theoretical studies with actionable insights. | Allows integration of diverse evidence. |
| 4 | Inclusion | Published 2005 or later, in English, with full text accessible. | Ensures contemporary and readable evidence base. |
| 5 | Exclusion | Studies on K–12 or vocational training without higher-education relevance. | Outside scope of higher education. |
| 6 | Exclusion | Articles outside Computer Science (e.g., general education). | Irrelevant discipline or too general. |
| 7 | Exclusion | Pure opinion or commentary papers without methodological basis. | Lack of empirical or conceptual rigor. |
| 8 | Exclusion | Non-English or published before 2005. | Language/time-frame filter. |
| 9 | Exclusion | Duplicates or inaccessible full texts. | Practical exclusion. |

Table 3.1: Inclusion and exclusion criteria for the integrative literature review.

searches—representing the five conceptual domains of this ILR across four databases—allowing readers to review both raw and deduplicated record counts, as well as the respective export formats. This documentation offers a transparent account of the breadth and depth of the literature search and forms part of the overall audit trail of the review.

### 3.1.1 Deduplication and Screening

Following the database searches, all retrieved records were exported in BibTEX format and imported into the reference management software Zotero[5] for centralized organization and evaluation. In accordance with Whittemore and Knafl, the ILR process emphasizes methodological transparency and the iterative refinement of data inclusion [28]. Within this environment, duplicate entries originating from overlapping databases (e.g., Scopus and Web of Science) were identified and removed through manual validation to ensure data integrity and consistency across sources.

**Title and Abstract Screening** Subsequently, all remaining records underwent Title and Abstract Screening (TIAB) to assess their overall relevance and methodological suitability. During this step, particular attention was given to whether the studies focused on higher education in CS or software engineering, offered conceptual or empirical insights related to the review's scope, and demonstrated sufficient methodological transparency. Publications that clearly fell outside this scope—such as those limited to K–12 (primary

[5] https://www.zotero.org

| Major Term | Core Search Structure (pseudocode) |
|---|---|
| **Skill Gaps** | (skill gap* ∨ competence gap* ∨ knowledge gap* ∨ skills mismatch) ∧ (computer science ∨ software engineering ∨ IT graduate*) ∧ (higher education ∨ universit* ∨ engineering education) ∧ ¬ (K-12 ∨ school ∨ teacher) |
| **Curriculum Development** | (curriculum alignment ∨ curriculum design ∨ curriculum develop* ∨ education reform ∨ industry-academia collaboration ∨ project-based learning) ∧ (computer science ∨ software engineering) ∧ (higher education ∨ universit* ∨ engineering education) |
| **Employability** | (employability ∨ graduate employability ∨ job readiness ∨ career readiness ∨ workplace skills ∨ soft skills) ∧ (computer science ∨ software engineering ∨ computing education) ∧ (higher education ∨ universit* ∨ engineering graduates) ∧ ¬ (K-12 ∨ vocational) |
| **Industry Expectations** | (employer expectations ∨ industry requirements ∨ skills demand ∨ employer perception ∨ labor market needs) ∧ (computer science ∨ software engineering ∨ ICT sector) ∧ (higher education ∨ universit* ∨ engineering graduates) |
| **Developing Countries** | (developing countries ∨ emerging economies ∨ Global South ∨ rural India) ∧ (computer science ∨ software engineering ∨ ICT education) ∧ (higher education ∨ universit* ∨ college ∨ tertiary education) |

Table 3.2: Simplified overview of core search terms and logical structures used across scientific databases (ACM, IEEE, Scopus, Web of Science) for the integrative literature review.

and secondary school) or vocational education, or purely technical case studies without educational relevance—were excluded.

**Fulltext Screening** Approximately 130 studies that passed the TIAB screening were reviewed in full text to verify their methodological rigor, contextual fit, and contribution to the overall conceptual framework of the ILR. At this stage, theoretically redundant works and studies lacking transparency or empirical depth were excluded. The remaining selection of around 40 high-quality publications formed the empirical and theoretical foundation for the synthesis presented in this thesis.

**Integration with Code System.** To ensure conceptual continuity between the literature review and the field study, representative insights from each of the five major terms of the ILR were synthesized into overarching analytical themes. As illustrated in Figure 1.1, the results of the ILR formed a preliminary analytical and informed summary, which was subsequently merged with the empirically derived code system from interviews and observations. Instead of importing all forty reviewed publications into MAXQDA, the synthesized ILR themes were integrated at a higher conceptual level into the empirical code hierarchy. This process ensured that both theoretical and field-based perspectives were systematically aligned and jointly informed the final thematic structure of the analysis. The resulting structure—summarized in Tables 3.3–3.7 and consolidated in the cross-domain overview (Table 3.8)—captures the key conceptual themes that informed the subsequent thematic analysis. This procedure enabled a cross-level comparison between theoretical frameworks identified in the literature and empirical patterns emerging from the data, thereby preserving conceptual coherence across both analytical phases [13], [28]. In doing so, the ILR not only informed the thematic analysis but also provided a theoretical anchor that guided the interpretation of findings in alignment with the research questions (see Section 1.3.1).

## 3.2 Major Term Findings: Skill Gaps

The reviewed literature reveals broad consensus that computer science and software engineering graduates remain insufficiently prepared for the practical and organizational realities of professional software development. Across large-scale meta-analyses, recurring gaps emerge in *software configuration management*, *process literacy*, and *testing*, while competencies in *requirements engineering*, *design*, and *architecture* are regarded as the most critical for professional readiness [35], [36], [37]. These findings underscore a persistent imbalance between theoretical instruction and hands-on proficiency.

Survey- and interview-based research corroborates this gap. Employers frequently report that new hires struggle with version control systems, continuous integration pipelines, and systematic testing practices [38], [39]. The inability to apply theoretical knowledge in real-world contexts—particularly within agile or DevOps environments—is often attributed to limited project-based experience and insufficient exposure to collaborative development settings [40]. Dobslaw et al. [39] further demonstrate that higher education curricula tend to emphasize conceptual understanding over emerging technologies such as Docker or Kubernetes, thereby widening the technology-specific skill gap.

Onboarding studies provide qualitative evidence of these shortcomings. Pham et al. [41] found that novice developers frequently perceive automated testing as a detached activity rather than an integral part of software development. This misconception leads to overreliance on mentoring and formal onboarding programs, which in turn raises organizational costs and delays productivity. Practitioners consequently call for stronger integration of testing and quality assurance throughout university curricula.

Complementary studies highlight that employability depends not only on technical mastery but also on *teamwork*, *communication*, and *problem-solving*—skills that remain underdeveloped among many graduates [40], [42]. Thus, the notion of "skill gaps" extends beyond technical literacy to encompass interpersonal and adaptive competencies necessary for professional collaboration.

These insights directly inform this study's research questions by:

| Label | Key Findings from Literature | Resulting Theme | Key Sources |
|---|---|---|---|
| ILR–T1 | Persistent gaps in testing, configuration management, and process literacy, limiting graduates' readiness for professional software development. | *Industry–Academia Skill Gap* | [35], [36], [37] |
| ILR–T2 | Insufficient project-based experience and limited exposure to collaborative or agile development environments. | *Lack of Practical Training* | [38], [39] |
| ILR–T3 | Extended onboarding phases and overreliance on mentoring due to missing integration of testing and teamwork in university curricula. | *Onboarding Dependency* | [40], [41], [42] |

Table 3.3: Themes derived from the literature on **Skill Gaps**.

- explaining employers' skill expectations and hiring priorities (**RQ1**),
- identifying the specific discrepancies between graduates' competencies and industry needs (**RQ2**), particularly in testing, configuration management, and process literacy, and
- linking these gaps to employability barriers and curricular implications (**RQ3**), emphasizing the necessity of practice-oriented and collaborative learning environments.

## 3.3 Major Term Findings: Curriculum Development

There is a widespread consensus in the literature that curriculum design in CSE and software engineering must continually evolve in response to industrial changes, national development goals, and local socio-economic realities. A well-aligned curriculum should combine theoretical foundations with applied, project-based, and communicative components that mirror authentic software engineering practice [43], [44]. Such integration requires continuous interaction between academia and industry to ensure that educational outcomes remain relevant and graduates are adequately prepared for professional work [45], [46].

Collaborative approaches—such as joint curriculum committees, structured internship programs, and regular industry feedback loops—have been shown to narrow the persistent gap between academic training and professional requirements [14], [45]. These models emphasize that curriculum alignment is not a one-time reform but a sustained process of co-design and evaluation. The Namibian and Indian case studies, in particular, illustrate that even in low-resource environments, mutual engagement between higher education institutions and industry can significantly enhance employability and contextual relevance [14], [45].

However, several structural and contextual barriers continue to impede effective implementation. In many institutions, courses still rely on outdated syllabi, theoretical delivery, or rote-based assessment methods (memorization technique based on repetition). Studies conducted in Sudan and rural India identify recurring obstacles, including limited faculty expertise, inadequate laboratory infrastructure, and highly centralized university regulations that restrict local curriculum adaptation [14], [47]. These systemic barriers often prevent meaningful reform even when awareness of industry needs exists.

Capacity-building research stresses that curriculum models transferred from high-income contexts often fail when local adaptation and faculty development are neglected [48], [49]. Sustainable reform therefore depends on empowering educators to interpret, contextualize, and iteratively update curricula. Without such local ownership, imported frameworks tend to remain superficial and unsustainable in the long term.

In sum, the reviewed evidence primarily informs **RQ2** and **RQ3**. It contributes to **RQ2** by revealing structural and pedagogical misalignments between curricula and industry requirements—especially in resource-constrained, regulated settings—and informs **RQ3** by identifying actionable strategies for reform, including industry collaboration, ongoing curriculum evaluation, and faculty capacity building that can enhance the employability and contextual fit of computer science education in rural higher education systems.

## 3.4 Major Term Findings: Employability

The reviewed literature consistently shows that employability in computer science and software engineering extends beyond technical proficiency. Employers emphasize hybrid competence profiles that combine disciplinary expertise with transferable skills such as communication, teamwork, adaptability, and self-directed learning [50], [51]. While most graduates secure employment shortly after graduation, many require extensive onboarding before becoming fully productive, reflecting a persistent mismatch between academic preparation and workplace expectations [39], [50].

Recent studies underline the growing importance of experiential learning and professional identity formation as key mechanisms for improving employability. Lundberg conceptualizes employability through the lens of *Communities of Practice*, suggesting that higher education should cultivate students' pre-professional identity and understanding of their future work roles [52]. Similarly, Lawrence-Fowler et al. argue that early engagement with authentic work contexts, internships, and reflective learning activities supports smoother transitions into employment [51].

| Label | Key Findings from Literature | Resulting Theme | Key Sources |
|---|---|---|---|
| ILR–T4 | Effective collaboration between academia and industry through internships, curriculum committees, and continuous feedback mechanisms strengthens employability and curriculum relevance. | *Curriculum–Industry Alignment* | [43], [44], [45], [46] |
| ILR–T5 | Faculty training, professional development, and pedagogical empowerment are prerequisites for sustainable curriculum reform. | *Faculty Capacity Building* | [48], [49] |
| ILR–T6 | Curriculum reforms must be context-sensitive and locally adapted to overcome structural constraints such as rigid affiliation systems and resource limitations. | *Contextual Curriculum Reform* | [14], [47] |

Table 3.4: Themes derived from the literature on **Curriculum Development and Alignment**.

However, definitions and expectations of employability vary across regions and industries. Comparative analyses across Europe and sub-Saharan Africa reveal cultural and contextual differences in which skills are valued most highly, yet adaptability, communication, and ethical awareness consistently emerge as universal expectations [53], [54]. From a theoretical standpoint, employability is increasingly understood as a social and developmental process rather than a static set of skills [55]. Graduates become employable by internalizing professional norms and developing a “feel for the game” within workplace communities.

Furthermore, professional certifications and micro-credentials have gained traction as alternative or complementary markers of competence, helping to bridge visible gaps between academic learning and industrial readiness [56]. These credentials provide tangible validation of technical and applied skills, particularly in rapidly evolving technology domains.

Overall, the reviewed studies clarify **RQ1** by specifying the combination of technical and non-technical skills most valued by employers in computer science and software engineering. They also inform **RQ3** by highlighting mechanisms through which employability can be strengthened—most notably through experiential learning, identity development, and structured certification pathways that align education with professional realities. Collectively, these insights affirm that employability is not merely the outcome of technical

| Label | Key Findings from Literature | Resulting Theme | Key Sources |
|---|---|---|---|
| ILR–T7 | Employability depends on a hybrid skill profile combining technical expertise with communication, teamwork, and self-management. | *Hybrid Skill Profiles* | [50], [51] |
| ILR–T8 | Experiential learning and professional identity formation are central to graduates' transition from academia to work. | *Experiential Learning and Professional Identity* | [51], [52], [55] |
| ILR–T9 | Employability expectations vary globally, but adaptability, communication, and ethical awareness are universally valued, micro-credentials offer practical means to demonstrate readiness. | *Global and Lifelong Employability* | [53], [54], [56] |

Table 3.5: Themes derived from the literature on **Employability**.

training but the product of an integrated system linking academic rigor, reflective practice, and sustained engagement with industry.

## 3.5 Major Term Findings: Industry Expectations

Across global and regional contexts, studies converge on the view that industry expectations for computer science and IT graduates encompass both technical expertise and professional competence. Employers consistently demand graduates who can combine solid programming and systems knowledge with communication, teamwork, and problem-solving abilities [38], [45], [57], [58]. Yet empirical evidence indicates that higher education programs often emphasize theoretical mastery over applied and interpersonal capabilities, leaving graduates underprepared for collaborative and dynamic work environments [59].

Radermacher et al. found through interviews with hiring managers in the U.S. and Europe that graduates frequently struggle with configuration management tools, communication, and practical testing experience [38]. Similar gaps have been identified in diverse contexts: Thai employers prioritize initiative, teamwork, and professional self-efficacy over academic achievement [57]. Turkish companies emphasize leadership, agile collaboration, and client communication—skills rarely formalized in university curricula [58], and North American employers report that insufficient project-based experience limits graduates' readiness for workplace integration [60].

Complementary research from Asia and the Global South provides parallel insights.

| Label | Key Findings from Literature | Resulting Theme | Key Sources |
|---|---|---|---|
| ILR–T10 | Employers consistently value teamwork, communication, and adaptability as core graduate competencies across regions. | *Workplace Readiness* | [38], [57], [59] |
| ILR–T11 | Professional competence increasingly requires agile collaboration, leadership, mentoring, and self-efficacy within project environments. | *Professional Competence* | [58], [61], [63] |
| ILR–T12 | Structured industry partnerships and experiential programs, such as internships or mentoring schemes, strengthen alignment and employability. | *Collaborative Learning Frameworks* | [60], [61], [62] |

Table 3.6: Themes derived from the literature on **Industry Expectations**.

In Sri Lanka, structured mentoring programs by industry professionals demonstrably improved students' readiness and alignment with employer expectations [61]. In India, Balakrishna observed that state-led initiatives such as the Andhra Pradesh Information Technology Academy (APITA) successfully enhanced graduates' employability by offering targeted technical and communication training, though challenges in scaling such programs persist [62]. Furthermore, Singh and Singh found a marked disconnect between academic preparation and industry needs in safety-critical software domains, emphasizing the need to embed reliability and systems engineering topics in undergraduate curricula [63].

Despite broad consensus on the desired skill mix, the reviewed literature highlights methodological and contextual limitations. Most studies focus on urban or industrialized environments, leaving rural and resource-limited regions underrepresented. Few investigations adopt longitudinal or mixed perspectives that integrate academic and employer viewpoints, which constrains understanding of how sustained industry involvement affects employability outcomes [60], [61].

These findings directly inform **RQ1** by clarifying the competencies and professional traits that employers consistently expect from computer science graduates—chiefly teamwork, adaptability, and communication. They also contribute to **RQ2** by exposing the persistent misalignment between academic training and workplace demands, particularly regarding project-based learning, agile methodologies, and customer-oriented collaboration. Finally, they support **RQ3** by underscoring the importance of structured partnerships, mentoring initiatives, and experiential frameworks as means of achieving sustainable curriculum–industry alignment.

## 3.6 Major Term Findings: Developing Countries

Research on computer science and software engineering education in developing countries reveals that contextual, infrastructural, and institutional factors significantly impact educational outcomes. Although global frameworks such as ACM/IEEE curricula promote standardization, their practical implementation in Africa and Asia often diverges due to resource constraints, differing pedagogical traditions, and governance structures [14], [45]. Empirical evidence consistently shows that graduates in these regions face systemic challenges such as outdated syllabi, limited laboratory access, and insufficiently prepared teaching staff [64], [65].

Studies from rural India and Namibia demonstrate that the key challenge lies less in curricular design than in implementation. In India, Spiesberger et al. identify a pronounced *design–reality gap* between nationally mandated course objectives and local teaching practices, primarily caused by examination-driven pedagogy and weak quality assurance mechanisms [14]. Similarly, Ntinda et al. report that Namibian institutions struggle to translate industry requirements into curricular content due to faculty skill shortages and limited opportunities for practical collaboration [45]. In both contexts, strengthening university–industry linkages and enabling faculty exposure to industry environments are seen as crucial for improving graduate employability.

Comparable patterns emerge in Mozambique, where Juvane et al. [15] observe that sustainable innovation ecosystems depend on continuous collaboration among universities, local companies, and international development partners. Their findings emphasize that without mechanisms for co-creation and market-oriented projects, national software industries remain dependent on imported technologies and external expertise.

More broadly, research from the Global South cautions that the integration of digital learning and ICT-enhanced education can unintentionally deepen existing inequalities if socio-economic realities are overlooked [64]. Sutinen and Tedre argue that in low-resource contexts, ICT4D initiatives should follow a participatory and culturally adaptive approach that privileges local ownership, capacity building, and contextual understanding over one-way technology transfer [65].

Overall, the reviewed evidence informs both **RQ2** and **RQ3**. It contributes to **RQ2** by illustrating how infrastructural limitations, rigid governance systems, and underqualified faculty impede the translation of curricular intentions into effective learning outcomes. Simultaneously, it informs **RQ3** by highlighting reform strategies that emphasize local curriculum ownership, participatory design, and long-term institutional capacity building. These insights bridge global frameworks and local realities, underscoring that sustainable educational transformation in developing contexts requires context-sensitive and inclusive approaches to curriculum and system design.

| Label | Key Findings from Literature | Resulting Theme | Key Sources |
|---|---|---|---|
| ILR–T13 | Infrastructural limitations, rigid governance, and insufficient faculty preparation constrain effective curriculum implementation. | *Contextual Constraints* | [64], [65] |
| ILR–T14 | Misalignment between national curriculum frameworks and their rural implementation due to examination-focused pedagogy and missing quality assurance. | *Design–Reality Gap* | [14], [45] |
| ILR–T15 | Sustainable reform depends on participatory, locally owned approaches that integrate universities, industry, and development partners. | *Participatory Reform* | [15], [65] |

Table 3.7: Themes derived from the literature on **Developing Countries**.

## 3.7 Findings Across All Major Terms

Across all five major terms, the reviewed literature reveals a persistent misalignment between higher education and industry requirements. While the thematic emphases vary—from skill gaps and curriculum alignment to employability, industry expectations, and the contextual challenges of developing regions—the underlying issue remains the same: the limited translation of academic knowledge into professional competence.

A broad range of studies demonstrates that technical expertise alone is insufficient for graduates to succeed in modern software and IT environments. Employers consistently highlight the equal importance of transferable and professional skills such as communication, teamwork, critical thinking, and adaptability [38], [50], [59]. Yet higher education institutions across both advanced and developing contexts struggle to embed these competencies within curricula often constrained by rigid accreditation standards, outdated pedagogical models, or insufficient industry engagement.

Three overarching insights emerge from this synthesis. First, **collaboration between academia and industry** is repeatedly identified as the most effective mechanism for aligning education with the dynamic demands of the labor market. Structured partnerships—including internships, advisory committees, and mentoring initiatives—consistently yield measurable improvements in graduate preparedness and institutional responsiveness [45], [60], [61]. Second, **experiential and project-based learning** functions as the primary conduit between theoretical instruction and practical competence, fostering technical proficiency alongside autonomous and reflective learning [14], [52]. Third, the **contextual dimension**—encompassing institutional capacity, local industry structures,

| **Major Term** | **Key Findings from Literature** | **Resulting Themes (Labels)** |
|---|---|---|
| Skill Gaps | Persistent gaps in testing, configuration management, and process literacy. Lack of project-based experience and practical training. | *ILR–T1* Industry–Academia Skill Gap, *ILR–T2* Lack of Practical Training, *ILR–T3* Onboarding Dependency. |
| Curriculum Development | Continuous academia–industry collaboration, faculty capacity building, and locally adapted curriculum design are crucial for sustainable alignment. | *ILR–T4* Curriculum–Industry Alignment, *ILR–T5* Faculty Capacity Building, *ILR–T6* Contextual Curriculum Reform. |
| Employability | Employability depends on hybrid skill profiles, experiential learning, and professional identity development. | *ILR–T7* Hybrid Skill Profiles, *ILR–T8* Experiential Learning, *ILR–T9* Global and Lifelong Employability. |
| Industry Expectations | Employers emphasize teamwork, agile practice, adaptability, and mentoring-oriented learning cultures. | *ILR–T10* Workplace Readiness, *ILR–T11* Professional Competence, *ILR–T12* Collaborative Learning Frameworks. |
| Developing Countries | Educational outcomes are constrained by infrastructural deficits, governance rigidity, and limited local ownership. | *ILR–T13* Contextual Constraints, *ILR–T14* Design–Reality Gap, *ILR–T15* Participatory Reform. |

Table 3.8: Summary of key findings and derived analytical themes across all major terms of the Integrative Literature Review. For detailed subthemes, see Tables 3.3–3.7.

and socio-economic inequality—determines whether alignment efforts are sustainable. Without faculty development, local ownership, and long-term capacity building, reforms risk remaining symbolic [48], [65].

The findings from the ILR are closely connected. Knowing what employers expect (**RQ1**) helps to understand how curricula address these needs and where structural gaps exist (**RQ2**). These insights, in turn, inform strategies to improve employability and guide curriculum reform (**RQ3**). The literature thus supports a systemic perspective in which employability is not an isolated outcome but the product of coherent interaction between educational design, institutional capacity, and stakeholder collaboration. This synthesis establishes the conceptual foundation for the subsequent thematic analysis, where empirical evidence from rural India will be interpreted in relation to these global patterns of convergence and divergence.

Across all five thematic domains, the ILR findings form a coherent conceptual model linking the demand for industry-relevant competencies (**RQ1**) with the identification of educational misalignments (**RQ2**) and actionable strategies for sustainable reform and employability enhancement (**RQ3**). This framework informs the subsequent thematic analysis in Chapter 6.1, ensuring that empirical findings are interpreted within a globally

| Theme Label | Theme Title | Related Research Question(s) |
|---|---|---|
| **ILR–T1** | Industry–Academia Skill Gap | **RQ1**, **RQ2** |
| **ILR–T2** | Lack of Practical Training | **RQ2** |
| **ILR–T3** | Onboarding Dependency | **RQ2**, **RQ3** |
| **ILR–T4** | Curriculum–Industry Alignment | **RQ2**, **RQ3** |
| **ILR–T5** | Faculty Capacity Building | **RQ3** |
| **ILR–T6** | Contextual Curriculum Reform | **RQ3** |
| **ILR–T7** | Hybrid Skill Profiles | **RQ1**, **RQ3** |
| **ILR–T8** | Experiential Learning | **RQ3** |
| **ILR–T9** | Global and Lifelong Employability | **RQ1**, **RQ3** |
| **ILR–T10** | Workplace Readiness | **RQ1**, **RQ2** |
| **ILR–T11** | Professional Competence | **RQ1**, **RQ3** |
| **ILR–T12** | Collaborative Learning Frameworks | **RQ3** |
| **ILR–T13** | Contextual Constraints | **RQ2**, **RQ3** |
| **ILR–T14** | Design–Reality Gap | **RQ2** |
| **ILR–T15** | Participatory Reform | **RQ3** |

Table 3.9: Mapping of ILR-derived themes (ILR–T1–T15) to the corresponding Research Questions (**RQ1**–**RQ3**).

validated conceptual structure.

CHAPTER 4

# Case Background

This chapter outlines the current state of CSE at the partner college and positions the institution within the broader Indian higher education landscape. The partner college was selected based on its scientific relevance rather than the existence of an institutional collaboration, as it represents the dominant higher-education model in India: more than 90% of higher education institutions in 2021 were private, university-affiliated colleges implementing curricula prescribed by nationally recognized state universities, commonly referred to as affiliating universities [66], [67]. It describes the structural, curricular, and pedagogical conditions under which computer science education is currently delivered, as well as the institutional dependencies that shape these processes. The purpose of this chapter is to provide a contextual understanding of the environment in which the field study was conducted, including governance structures, teaching practices, and systemic challenges that affect curriculum implementation and graduate employability. The description draws on official documentation from the affiliating university, national education policy reports such as the *National Education Policy (NEP 2020)*, and prior academic collaborations between TU Wien and the partner institution, which investigated curriculum implementation and design–reality gaps in rural Indian engineering education [14], [30], [68]. These previous studies provide a foundation for situating this thesis within the ongoing research collaboration between Austria and India, focusing on enhancing the alignment between academic curricula and private-sector requirements in computer science education.

## 4.1 Computer Science Education

CSE is a multidisciplinary field that integrates the theoretical foundations of computation with applied approaches to software development, problem-solving, and the use of digital technologies across diverse domains. Its objective extends beyond technical proficiency: it aims to cultivate the cognitive, analytical, and social competencies required to apply computational thinking to real-world challenges [69], [70]. As such, CSE fulfils both an academic and an economic function—fostering innovation, technological progress, and national development through a skilled and adaptable workforce [71]. At the global level, CSE curricula are primarily guided by the frameworks of the Association for Computing Machinery (ACM) and the IEEE Computer Society. These guidelines specify learning outcomes in areas such as algorithms, data structures, programming languages, software engineering, and professional practice [70]. They emphasize the integration of theoretical understanding with practical experience, encouraging institutions to adopt project-based and experiential learning approaches that promote creativity, teamwork, and problem-solving. Nevertheless, the extent and quality of implementation differ widely, influenced by institutional autonomy, resource availability, and pedagogical traditions. In developing and emerging economies—including India—CSE has become a strategic priority due to its perceived potential to drive economic transformation and enhance employability [11], [65]. However, persistent obstacles such as outdated curricula, insufficiently trained faculty, and limited infrastructure continue to constrain learning outcomes [64]. Particularly in rural contexts, a predominant focus on rote learning and examination-driven assessment limits opportunities for students to engage in creative problem-solving, teamwork, and hands-on software development. The Indian higher education system exemplifies this structural tension between rapid expansion and uneven quality. Since the 1990s, the number of engineering colleges has increased dramatically in response to the global demand for technical graduates [66]. Yet, national assessments repeatedly highlight a persistent skills mismatch between graduate competencies and industry expectations [72]. To mitigate this gap, national initiatives and accreditation frameworks—such as the implementation of outcome-based education (OBE) and mandatory curriculum updates—promote closer collaboration between higher education institutions and industry. Despite these reforms, adoption and quality assurance remain inconsistent, particularly among affiliated and rural institutions where structural and resource-related constraints persist [14]. In summary, CSE constitutes the academic foundation for preparing graduates to participate in knowledge-driven economies. However, the effectiveness of these programs ultimately depends on how global educational frameworks are interpreted and enacted within specific institutional settings. Understanding the realities of local implementation—exemplified by the partner college in Andhra Pradesh—is therefore crucial for assessing how curricular design, pedagogy, and institutional context shape the alignment between education and industry requirements.

## 4.2 Higher Education Context in India

India's higher education system is characterized by a dual structure that distinguishes between affiliating universities and their affiliated colleges. More than 90% of HEIs in 2021 were private, affiliated institutions that implement curricula developed by nationally recognized state universities, commonly referred to as affiliating universities [66], [67]. These universities prescribe detailed syllabi, course structures, and learning outcomes for each academic program, and they oversee standardized end-of-semester examinations across their network of colleges [11]. Affiliated colleges are required to follow these curricula without modification. Their institutional autonomy is largely restricted to administrative and pedagogical aspects—such as scheduling, classroom management, and internal assessments—while core curricular decisions remain under the authority of the affiliating university. According to the University Grant Commission (UGC) Annual Report 2018–2019, India counted 40,489 such colleges, of which more than 60% were located in rural areas [67]. This structure ensures nationwide standardization and quality control, yet it also limits the capacity of individual institutions to tailor programs to local contexts or industry needs, particularly in rural and resource-constrained environments.

## 4.3 Case Context: Partner College

This section provides contextual background on the partner college and its institutional environment, outlining structural, curricular, and pedagogical conditions that frame the empirical investigation.

### 4.3.1 Status Quo at the Indian University

The partner college, established in 2015, is one of more than 80 affiliated institutions under the governance of the affiliating university. As an affiliated college, it follows the official Jawaharlal Nehru Technological University Anantapuramu (JNTUA) curriculum for engineering programs, including all course contents, credit structures, and examination schemes. Students are evaluated through standardized exams administered by the university, while the partner college is responsible for delivering the prescribed lectures, laboratory sessions, and internal assessments. This institutional arrangement guarantees uniform academic standards across the university network but restricts the college's ability to introduce localized or practice-oriented teaching innovations. Previous collaborations between the partner institution and INSO have revealed that such centralized governance often results in a design–reality gap—a discrepancy between the intended curriculum and its implementation in everyday classroom practice, particularly regarding applied learning and project-based instruction [14]. These systemic constraints typify the situation of many rural engineering colleges across India and define the broader educational environment in which this research was conducted. In mid-2025, the partner college was granted autonomous status, marking a major institutional transition. This autonomy allows the college to design and revise its curricula independently of its affiliating university, fostering greater flexibility in aligning course content with local industry needs and modern pedagogical approaches. The forthcoming years will show whether this structural shift enables sustained curriculum innovation and bridges the persistent gap between academic training and employability.

Figure 4.1 illustrates the geographical location of the study site in the state of Andhra Pradesh, as well as other relevant urban centers mentioned.


Figure 4.1: Geographical location of the study region and key related cities in Southern India. Highlighted in yellow is the state of Andhra Pradesh, with marked cities including Bengaluru, Hyderabad, and Pune. Source: Created with `ultimaps.com`

### 4.3.2 Current Teaching Practices

The CSE program at the partner college follows the standard four-year Bachelor of Technology (B.Tech) structure prescribed by the affiliating university, consisting of eight semesters. The first two years emphasize foundational subjects such as mathematics, physics, and introductory programming, while advanced semesters focus on specialized modules including database systems, software engineering, and artificial intelligence. Programming languages typically change across semesters (e.g., C, Python, Java), exposing students to multiple paradigms and syntaxes. Theoretical classes are complemented by laboratory sessions intended to provide hands-on experience in programming and system design. In practice, however, teaching remains predominantly lecture-based with limited opportunities for active student participation. Laboratory exercises frequently involve replicating predefined examples from laboratory manuals rather than developing original or open-ended solutions. Such teacher-centred practices and the dominance of rote learning are widely reported characteristics of engineering education in India, particularly in affiliated and rural institutions [11], [14], [30], [64]. Large class sizes, restricted access to computers, and time constraints during exam periods further limit the scope for experimentation and peer learning. Faculty members often teach several subjects per semester while managing administrative tasks, leaving little capacity for updating materials or integrating contemporary technologies into coursework [14], [66]. As a result, both instructors and students tend to prioritize examination preparation and syllabus completion over exploratory or project-based learning—an imbalance also documented in national reviews of engineering education [67], [72].

### 4.3.3 Institutional Challenges

The challenges observed at the partner college mirror structural issues prevalent across India's affiliating university system, where standardized curricula and resource disparities constrain local innovation [11], [14]. The most significant institutional constraints include:

- **Infrastructure and Resource Limitations:** Computer laboratories frequently rely on outdated hardware, limited licensed software, and unstable internet connectivity, restricting opportunities for practical training and modern tool usage [64], [66].
- **Pedagogical Constraints:** Instruction remains largely teacher-centred and examination-driven, with few opportunities for students to engage in critical discussion, problem-solving, or teamwork—patterns well documented in Indian higher education research [11], [14].
- **Faculty Workload and Development:** Academic staff typically manage multiple courses and administrative duties per semester, leaving limited capacity for continuous pedagogical training or research engagement [30], [67].
- **Assessment System:** The standardized examination framework of affiliating universities emphasizes theoretical recall and grading uniformity rather than practical application, which narrows the pedagogical flexibility of local institutions [11], [14].

These structural and pedagogical limitations collectively shape the context in which CSE is taught at the partner college. They also underscore the persistence of a gap between curricular objectives and actual teaching practice—an issue central to this study's analysis of curriculum–industry alignment in rural higher education settings.

## 4.4 Regional and Policy Context

This section situates the case study within its broader regional, socio-economic, and policy environment, outlining contextual factors that shape higher education and graduate employability in Andhra Pradesh.

### 4.4.1 Socio-Economic Context of Andhra Pradesh

India's economy has undergone a rapid transformation over the past two decades, driven by the expansion of the ICT and service sectors. While metropolitan regions such as Bengaluru, Hyderabad, and Pune concentrate the majority of high-value digital employment, much of rural India continues to depend on agriculture and low-productivity services [73], [74]. The state of AP exemplifies this structural dualism between industrial growth and agrarian dependence. According to the *Socio-Economic Survey 2024–25* published by the Government of Andhra Pradesh, the state's Gross State Domestic Product (GSDP) reached approximately 16 trillion Indian Rupees (INRs) in 2024, which—based on the mid-2024 exchange rate of 1 INR $\approx$ 0.011 EUR—corresponds to roughly 176 billion EUR. For comparison, Austria's GSDP in 2023 amounted to about 480 billion EUR, while the combined GDP of the EU-27 reached approximately 14 trillion EUR. Thus, Andhra Pradesh's economy represents around 37 percent of Austria's GDP and roughly 1.3 percent of the EU's total economic output [75], [76], [77], [78]. Services and industry jointly contribute around two-thirds of total output, while agriculture still employs more than half of the workforce. Recent development initiatives—such as port-led industrialization around Visakhapatnam and the expansion of Software Technology Parks in Visakhapatnam and Tirupati—reflect the state's gradual integration into India's digital economy [79]. However, most CS graduates from rural colleges in AP compete in labour markets centred in neighbouring metropolitan hubs. Hyderabad alone accounts for Information Technology (IT) exports exceeding 2.7 trillion INRs and employs nearly one million professionals [80], while Bengaluru represents India's largest technology cluster with over 1.5 million employees in the ICT sector [81]. This geographic and skills imbalance between the location of training and the location of employment constitutes a key structural challenge for graduate employability—a recurring theme throughout this study.

### 4.4.2 Educational Reforms and the National Education Policy 2020

The National Education Policy 2020 (NEP 2020) represents the most comprehensive reform of India's higher education system since 1986. It envisions a flexible and holistic academic structure with four-year degree programmes that include multiple exit options, interdisciplinary learning pathways, and an emphasis on experiential and technology-enhanced instruction [68]. A central pillar of the policy is the promotion of closer connections between academia and industry, ensuring that graduates acquire both technical expertise and transferable skills relevant to contemporary labour markets. For the field of CSE, the NEP 2020 advocates curriculum models grounded in Project-Based Learning (PBL), internships, and the integration of digital learning platforms—approaches designed to foster creativity, innovation, and applied problem-solving [11]. These measures are particularly relevant for rural engineering colleges, where infrastructural constraints and limited access to modern learning environments have traditionally hindered the effective delivery of practice-oriented education. The policy also promotes institutional autonomy and alignment with national skills frameworks, including OBE standards, to enhance both quality assurance and employability outcomes [66]. Despite this progressive framework, implementation remains uneven across India's states and university systems. Regions with extensive networks of affiliated private colleges—such as AP—continue to face substantial challenges in adapting policy recommendations to local realities. Persistent deficits in infrastructure, faculty development, and industry engagement have slowed the transition from prescriptive curricula to competency-based and experiential models [14], [82]. This ongoing divergence between policy intent and institutional reality reflects the broader curriculum–industry gap observed in this study's action research at the partner college and other rural higher education institutions in southern India.

CHAPTER 5

# Case Study Design

This chapter describes the empirical data collection process conducted as part of the single-case study. It outlines the preparatory activities undertaken before action research, the on-site observations carried out at the partner college, and the semi-structured interviews conducted with key stakeholders. In addition, the chapter addresses ethical considerations and procedural safeguards applied throughout the data collection phase to ensure methodological rigor, transparency, and compliance with established qualitative research standards.

## 5.1 Preparation

The preparatory phase of the action research was crucial to ensure methodological rigor and to establish the feasibility of conducting interviews and observations within the context of rural Indian higher education. In addition to the ILR in Section 3, the researcher began preparations for on-site data collection at the partner college in AP, India. The field visit, carried out between mid-October and mid-December 2024, aimed to deepen the understanding of local teaching practices and institutional structures through direct interaction with professors, students, and alumni.

The researcher's physical presence on-site enabled the implementation of semi-structured interviews, participatory observations, and active pedagogical engagement within the college environment. This proximity enabled authentic insights into the socio-cultural and organizational dynamics that shape teaching and learning processes.

The planning of interviews and observations followed the empirical procedures introduced in Section 1.4.4 and Section 1.4.5. An interview guide was developed in close consultation with this thesis' advisor before action research and is included in the Appendix A. Separate guidelines were developed for graduates and industry representatives to facilitate a nuanced examination of perceptions regarding employability, curriculum relevance, and skill requirements. Before data collection commenced, all participants were informed about the study's objectives and procedures. They were provided with written consent forms that covered audio recording, voluntary participation, and the anonymized use of their statements.

This procedure ensured compliance with ethical research standards and reflected the principles of informed consent, transparency, and voluntary participation as emphasized in qualitative research ethics frameworks [23], [83].

## 5.2 Observations

The observations were conducted as uncontrolled and participatory, as described in Section 1.4.5. This approach ensured that teachers and students behaved as naturally as possible, allowing authentic challenges to emerge rather than being masked by altered conduct [14], [23]. Consistent with the action research framework outlined in Section 1.4.4, observations took place during the on-site visit between October 2024 and January 2025 in AP, India. The author adopted the role of a participant-as-observer during workshops, discussions, and teaching activities. This role involves active participation while openly disclosing the researcher's identity, as illustrated in Figure 5.1. Alternative roles, such as the complete observer or complete participant, either omit observation or conceal the researcher's identity and were therefore unsuitable for this study's ethical and practical setting.

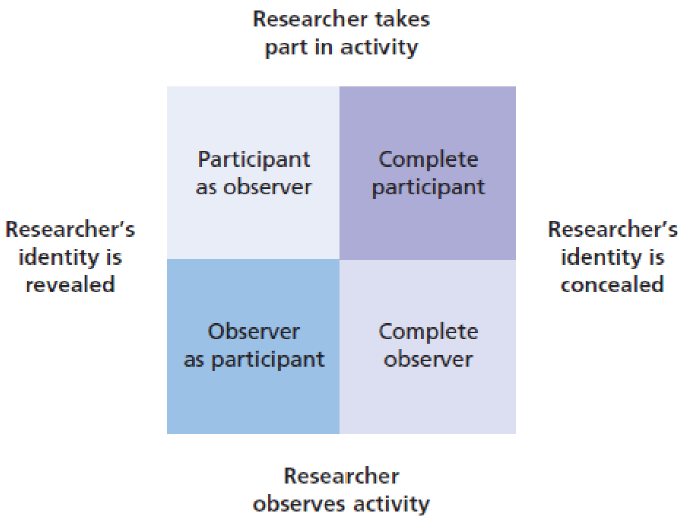


Figure 5.1: Typology of researcher roles in participant observation [23].

The observations aimed to gain insights into teaching practices and institutional routines from an insider's perspective, complementing the semi-structured interviews. They provided valuable understanding of classroom interaction, student engagement, and faculty behaviour within the rural higher education context of the partner college. The resulting notes informed the interpretation of how professors conduct classes and how students experience learning on a daily basis, thereby contributing to answering the research questions of this thesis.

**Data Recording and Reflexivity** After each session, observation notes were documented in a structured research diary. Each entry included the date, location, brief contextual description, and preliminary analytical reflections. Given the dual role of the researcher as both lecturer and observer, reflexivity was maintained throughout the process to mitigate potential bias. Separate sections in the diary distinguished factual observations from subjective interpretations, increasing transparency and reliability [23].

**Overview of Observations** Table 5.1 presents a summarised overview of selected observations recorded during action research, while the complete log is provided in Appendix F.1. The data reveal recurring institutional and infrastructural challenges that shaped students' learning experiences.

### 5.2.1 Document Analysis: Affiliating University Curriculum (R20–R23)

In addition to the field observations (O1–O6), a document analysis was conducted to contextualise the observed teaching practices within the formal curricular framework prescribed by the affiliating university. This analysis included both the previous R20 and the revised R23 versions of the Bachelor of Technology (B.Tech.) programme in CSE, which define the academic and pedagogical scope for all affiliated colleges in the region of AP. As the partner college is one of more than eighty institutions implementing these curricula, the analysis supports the understanding of how curriculum design and implementation interact within this regulated context.

**Purpose and Method** The analysis examined how designed learning outcomes, course sequencing, and skill orientation evolved from the R20 to the R23 curriculum. Both syllabus documents were reviewed regarding structure, thematic composition, degree of practice orientation, and employability focus. The findings were summarised thematically and aligned with the results of the interviews and observations (see Chapter 6.1). In reviewing both curricula, particular attention was also given to the cognitive level of the intended learning outcomes, drawing on Bloom's Taxonomy as a reference framework [84] . The taxonomy distinguishes between lower-order processes such as remembering and understanding, and higher-order competencies such as applying, analysing, and creating (see Figure 5.2). This allows the curriculum to be assessed not only in terms of content but also in terms of the depth of cognitive engagement it promotes.

**Findings** The analysis indicates a gradual modernisation of the curriculum, while the overall structure remains strongly theory-oriented. Table 5.2 presents an overview of the key findings identified during the analysis.

| ID | Location | Theme | Description and Reflexive Note |
|---|---|---|---|
| **O1** | Campus | Power outages | Frequent power and internet outages interrupted lectures and lab work. Reflexive: Infrastructure failures significantly disrupted the learning flow. |
| **O2** | Lab Room | Lack of laptops | Students often shared computers or borrowed laptops from peers. Reflexive: Unequal access limited practical learning opportunities. |
| **O3** | Faculty Office | Faculty values | Faculty emphasised discipline and loyalty more than creativity or enjoyment in learning. Reflexive: Hierarchical culture dominated classroom dynamics. |
| **O4** | Library | Financial burden | Several students mentioned that their families took out loans to pay tuition fees. Reflexive: Financial stress increased performance pressure. |
| **O5** | Campus | Career ambitions | Many students expressed dreams of working for global technology companies. Reflexive: Aspirations were idealised; career guidance was lacking. |
| **O6** | Lab Room | Communication behaviour | Students rarely asked questions and hesitated to engage in discussion. Reflexive: Hierarchical norms suppressed participation. |
| **O7** | Classroom | Admission test irregularities | Reports of cheating and informal handling of entrance tests. Reflexive: Weak institutional control mechanisms. |
| **O8** | Classroom | Teaching quality | Faculty relied on rote memorisation and dictation. Reflexive: Limited interactive teaching practices. |
| **O9** | Campus | Gender interaction | Male and female students sat separately and rarely interacted. Reflexive: Cultural norms shaped communication and collaboration. |
| **O10** | Lab Room | Use of AI tools | Students experimented with generative AI (e.g., ChatGPT) for code generation. Reflexive: Indicates growing self-learning initiative despite limited guidance. |

Table 5.1: Selected observations recorded during action research at the partner college (Oct 24-Jan 25).

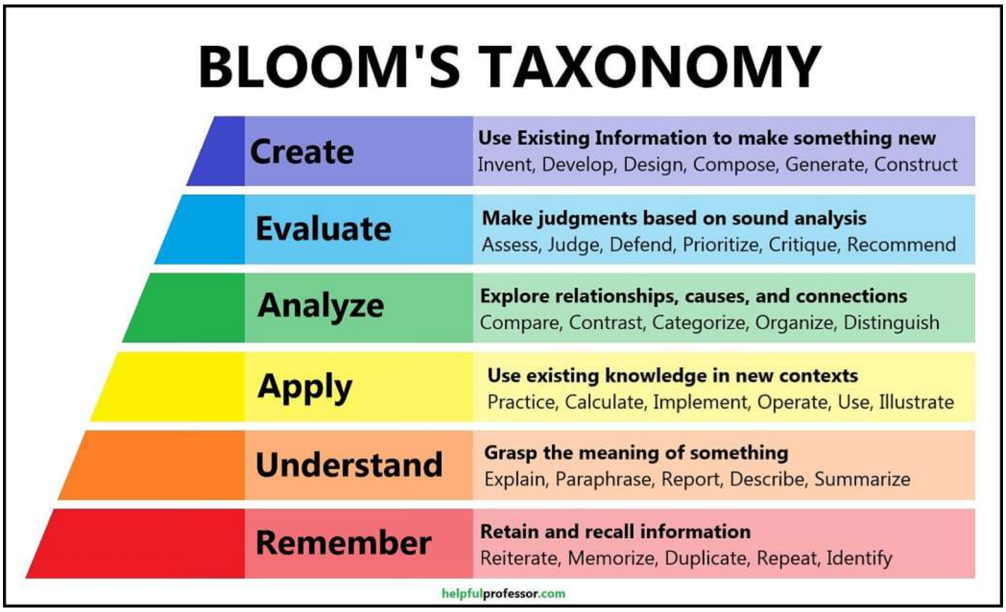


Figure 5.2: Bloom's Taxonomy of cognitive learning objectives [85].

**Interpretation** The transition from R20 to R23 reflects an incremental alignment with the national OBE framework prescribed by the All India Council for Technical Education (AICTE), introducing employability-related modules and interdisciplinary exposure. However, the reform remains largely formal and does not fundamentally alter the pedagogical approach. The curriculum continues to rely on summative assessment and compartmentalised lab exercises rather than integrated, project-based learning. This design reality mirrors the classroom observations (O3–O6), where teaching remained mainly lecture-driven and theoretical, with little evidence of active or experiential learning.

Overall, the findings confirm the persistent design–reality gaps identified in related research [14], [30], [86], particularly regarding curriculum evaluation, teacher training, and employability alignment. The identified curriculum features (C1–C5) will therefore serve as reference points for interpreting and triangulating the qualitative results in the findings chapter.

**Analytical Integration** The observational data were triangulated with interview findings and integrated into the thematic analysis (see Section 6.1). This enabled the identification of converging and diverging patterns between reported experiences and observed practices, thereby enhancing the study's overall credibility and validity.

## 5.3 Semi-Structured Interviews

Semi-structured interviews were conducted following the approach outlined in Section 1.4.5, which enables guided yet flexible conversations. They were performed with graduates

| ID | Label | Summary of Findings |
|---|---|---|
| **C1** | Theoretical orientation (R20) | Curriculum dominated by lecture-based courses (3-0-0); limited project-based learning. Practical labs exist but are separated and low in credit weight, reinforcing teacher-centred instruction. |
| **C2** | Fragmented practical components | Laboratory courses (e.g., Programming, Data Structures, AI Lab) are isolated from theory and not embedded in larger project work. No continuous assessment or portfolio-based evaluation is foreseen. |
| **C3** | Skill-oriented additions (R23) | New modules introduced, such as Full Stack Development, Design Thinking & Innovation, and Human Values & Professional Ethics. These reflect compliance with national OBE guidelines but remain minor in scope and credit weight. |
| **C4** | Missing industry linkages | Neither R20 nor R23 establish structured industry collaboration, extended internships, or formal curriculum evaluation mechanisms. No evidence of advisory boards or systematic employer feedback. |
| **C5** | Cognitive learning focus | Learning outcomes are formulated mainly at lower Bloom levels (remember, understand), with limited emphasis on application or synthesis [84]. |

Table 5.2: Overview of key findings from the curriculum analysis (R20–R23).

and CS industry experts to explore perceived skill gaps and matches, industry expectations, and the lived experiences of teachers and students at a rural HEI. The interviews also covered hiring, onboarding, and on-the-job training practices, thereby contributing directly to answering this thesis's research questions (see Section 1.3.1).

Semi-structured interviews were chosen as the most suitable method due to the critical importance of establishing personal rapport with participants and the exploratory nature of the research [23]. This approach allowed for guided yet flexible conversations that encouraged participants to elaborate on experiences and perceptions while enabling the researcher to follow up on relevant issues as they arose.

**Interview Preparation and Setting** When designing and conducting the interviews, several methodological aspects outlined by Saunders et al. [23, pp. 383] were considered to ensure validity, reliability, and ethical integrity:

**Level of knowledge about the context and culture of the organization** Prior to data collection, the researcher possessed substantial contextual understanding of the partner college through earlier collaborations in the TU Wien–Indian-College partnership and the preceding bachelor's thesis conducted at the same site. This prior engagement offered deep insights into institutional structures, hierarchical communication patterns, and the socio-economic background of students and faculty. Familiarity with local norms facilitated trust-building and culturally sensitive communication. Nevertheless, the researcher remained reflexive about his dual role as both lecturer and investigator, continuously documenting reflections to minimize bias.

**Level of information provided to the interviewees** To promote openness and credibility, participants were informed in advance about the study's objectives and the interview topics, which included: the transition from graduation to employment, everyday academic life, and the hiring and onboarding processes of companies. In line with ethical research principles [23], participants received written information sheets and consent forms that detailed the voluntary nature of their participation and the anonymized handling of their statements.

**Interview location** Interviews were conducted at locations where participants could speak comfortably and without disturbance [23]. Most sessions took place in informal settings such as cafés, offices, or quiet campus areas in rural Andhra Pradesh and Bangalore between November and December 2024. The relaxed atmosphere encouraged open discussion and authentic responses.

**Opening comments and introduction** As emphasized by Saunders et al., the initial moments of an interview substantially influence its outcome. Therefore, participants were welcomed with a brief introduction that explained the research purpose, data confidentiality, and the intended use of the results. The following key points were consistently addressed:

- Clarification of the study's aim and participants' potential contributions.
- Assurance of anonymized data handling and restricted access to recordings (limited to the author, supervisor, and co-supervisor).
- Explicit request for permission to record the interview (audio and/or video).
- Notification that data would be securely stored and deleted after project completion.

**Questioning approach** Given that English was not the mother tongue of either the interviewer or participants, questions were formulated in simple and clear language, avoiding technical jargon or abstract phrasing [23]. When misunderstandings occurred, questions were rephrased or repeated to ensure comprehension. All questions were open-ended to avoid bias and focused on participants' real-world experiences rather than theoretical constructs. Two categories of questions were developed: one addressing specific aspects of the Java lab courses, and another covering general experiences with college routines and professional life. Sensitive questions were reserved for the later stages of each interview to allow rapport and trust to develop, following the guidance of Healey and Rawlinson [87].

**Accurate and complete data recording** In accordance with Saunders et al., all interviews were audio recorded with prior consent. Supplementary field notes were taken to capture non-verbal cues and contextual details. Each transcript was labeled with date, time, location, and participant type (graduate or industry expert) to ensure systematic organization and traceability.

**Sampling and Conduct** Participants were selected through purposive sampling to ensure representation from both academic and professional contexts. Graduates and industry experts were identified via personal contacts established during the researcher's teaching period at the partner college and through professional networking platforms such as LinkedIn[1]. A combination of convenience and snowball sampling was employed, the latter referring to a method where existing participants recommend further potential interviewees with relevant backgrounds [23]. This facilitated access to diverse perspectives across rural and urban employment contexts. Interviews lasted between 30 and 90 minutes and were primarily conducted in English.

The author's prior engagement as a visiting lecturer at the partner institution enabled an atmosphere of trust and openness, yet also required continuous reflexivity to prevent potential bias—particularly when participants referenced courses or programs in which the researcher was involved.

All participants voluntarily consented to the recording and transcription of their interviews, which were later analyzed using thematic analysis (see Section 1.4.5 and Section 6.1). Pseudonyms are consistently used throughout this thesis to safeguard anonymity.

[1] https://www.linkedin.com

| Code | Role / Affiliation | Gender | Years of Experience | Description |
|---|---|---|---|---|
| G1 | Graduate (Software Developer) | Male | 2 | Employed at a start-up private IT company in Bangalore. |
| G2 | Graduate (Associate Software Developer) | Female | 2 | Works at a multinational IT consulting and software development company. |
| G3 | Graduate (Associate Software Developer) | Female | 3 | Responsible for backend development in a subsystem of a large enterprise application. |
| E1 | Employer (CEO) | Male | 10+ | Oversees software teams and recruitment processes in his IT company. |
| E2 | Employer (Workforce Manager) | Male | 29 | Responsible for hiring of technical staff at an international firm. |

Table 5.3: Overview of Interview Participants, Gender, and Years of Experience

CHAPTER 6

# Case Analysis

## 6.1 Thematic Analysis

According to Braun and Clarke, thematic analysis is a versatile and straightforward method suitable for researchers with minimal or no expertise in qualitative research. Moreover, it is described as an advantageous research methodology in which people are engaged as collaborators [13]. Consequently, thematic analysis was employed to examine semi-structured interviews and observations as described in Section 1.4.5.

The guidelines established by Braun and Clarke for executing a theme analysis have been considered in the application of the thematic analysis technique. They assert that this procedure has six stages [13]:

1. Familiarize yourself with the collected data: Transcribe the interviews, organize the observation notes, analyze them, and document your preliminary reflections.
2. Create initial codes: Develop a systematic coding framework for the compelling data attributes within the comprehensive dataset, then aggregate the information pertinent to each code.
3. Search for themes: Potential themes must be summarized, with each theme accompanied by a compilation of relevant information.
4. Review themes: A thematic map of the study is constructed by examining and refining the themes through the initial coded data extraction, followed by the comprehensive data collection.
5. Defining and naming themes: Provide a comprehensive explanation of each theme, identify its fundamental concepts and elements, and formulate concise titles and definitions for each.

6. Write a verdict: Identify prominent parts from the themes and thereafter provide an accurate evaluation of those essential discoveries. Establish a correlation between the analysis and the research question.

The Braun and Clarke methodology, which is described in detail in Section 6.1, is used to thematically analyze the conducted interviews, observations and findings from the ILR in the part that follows.

### 6.1.1 Data Familiarization and Coding

All interview transcripts, observation notes, curriculum findings and selected visual materials from the action research, such as photographs and short video recordings, were imported into MAXQDA. In line with Braun and Clarke's six-phase framework for thematic analysis, the researcher first engaged in a phase of data familiarization by repeatedly reading the material and noting initial ideas [13].

An inductive coding approach was used to stay close to the participants' views, while ideas from the ILR supported the interpretation. Codes were created line by line and refined several times until a clear and structured codebook was developed. Each code was linked to its original source (graduate, employer, or observation) to ensure transparency and traceability.

### 6.1.2 Theme Development and Validation

The generated codes were then grouped into broader themes according to their similarity in meaning and their relevance to the research questions (RQ1–RQ3). After connecting the findings to the Research Questions the themes were reviewed several times to make sure that the data within a theme were consistent and that the themes were clearly distinct from each other, following the principles by Braun and Clarke [13]. A code frequency table (Table 6.1) was created to show how often each code occurred.

| ID | Code System | Summary | Freq. |
|---|---|---|---|
| **I1** | **Infrastructure** | Infrastructure-related constraints and enablers. | 11 |
| **I2** | **Remote Work / Hybrid Work** | Non-onsite work patterns and their effects. | 2 |
| **I3** | **Workplace Culture & Environment** | Team norms, climate, and values at work. | 8 |

*Continued on next page*

| ID | Code System | Summary | Freq. |
|---|---|---|---|
| **I4** | **Collaboration with Academia** | University–industry collaboration formats. | 15 |
| **I5** | **Skill Gaps Observed** | Mismatches between graduate skills and industry needs. | 7 |
| **I6** | **Industry Expectations** | Employer views on desired competencies. | 4 |
| **I7** | **Skills useful/required** | Skills explicitly mentioned as helpful. | 2 |
| I7.1 | Hardskills | Technical capabilities and tooling. | 9 |
| I7.2 | Softskills | Communication, teamwork, and interpersonal skills. | 8 |
| **I8** | **Gender differences** | Gender-based variations in access and participation. | 4 |
| **I9** | **Hiring process** | Stages and criteria during recruitment. | 21 |
| I9.1 | Recruitment Sources | Candidate sourcing channels. | 1 |
| I9.2 | Job offer revoked due to market downturn | Offer withdrawal cases. | 1 |
| I9.3 | Self-initiated job search on startup platforms | Graduate-driven applications. | 1 |
| I9.4 | Aptitude Test | Analytical and reasoning screening. | 8 |
| I9.5 | Soft skills assessment in hiring | Evaluation of communication or motivation. | 12 |
| I9.6 | Hard skills assessment in hiring | Practical or technical skill checks. | 15 |
| I9.6.1 | Hands-on hiring process with real coding tasks | Work-like coding projects. | 7 |
| **I10** | **Worklife** | Early career experiences and workplace adaptation. | 17 |
| I10.1 | Training on the job | Mentoring and practical upskilling after entry. | 18 |

*Continued on next page*

| ID | Code System | Summary | Freq. |
|---|---|---|---|
| I10.1.1 | Exams | Internal evaluations or certifications. | 2 |
| I10.1.2 | Hardskill Training | Formal technical training modules. | 7 |
| I10.1.3 | Softskill Training | Workshops on teamwork and communication. | 3 |
| I10.2 | Product/Service based companies | Distinction between firm types. | 5 |
| I10.3 | Technology used in job | Tools, stacks, and frameworks in daily use. | 11 |
| **I11** | **Use of AI tools** | Adoption and perceived benefits of AI in work. | 6 |
| **I12** | **RIT Experience** | Experiences from RIT modules during study. | 7 |
| **I13** | **Educational Preparedness** | Perceived job readiness resulting from education. | 25 |
| I13.1 | Learning | Learning experiences and methods. | 4 |
| I13.1.1 | Self-learning through online platforms | MOOCs, YouTube, and self-directed learning. | 14 |
| I13.1.2 | Online certifications to supplement knowledge | Certificates and micro-credentials. | 3 |
| I13.2 | College Experience | Perceptions of college processes and quality. | 54 |
| I13.2.1 | Rural/Urban differences | Contextual and resource disparities. | 13 |
| I13.2.2 | Teaching | Effectiveness of teaching and lecturers. | 22 |
| I13.2.3 | Placement in college | Career support and placement quality. | 9 |
| **I14** | **Suggestions** | Improvement ideas and feedback from participants. | 7 |
| I14.1 | Tips for students | Practical advice for future cohorts. | 6 |
| I14.2 | College Improvement suggestions | Teaching, curriculum, and infrastructure reforms. | 18 |

Table 6.1: Overview of Codes and Subcodes from MAXQDA

### 6.1.3 Cloud Visualizations

To complement the qualitative coding and thematic analysis, visualizations in the form of word clouds were created from the interview transcripts and coded data in MAXQDA. These word clouds provide an accessible overview of the most frequently used terms in the dataset, highlighting linguistic patterns and offering a first impression of recurring themes. While they do not replace systematic coding or interpretation, they serve as a useful visual tool to illustrate how participants expressed and emphasized key concepts.

Figure 6.1 visualizes the most frequent words mentioned by a graduate (G3). Prominent terms such as *college*, *code*, *work*, and *exam* reflect general references to learning, programming, and students' perceptions of their educational experience. This overview provides a basic impression of recurring topics discussed by graduates in relation to RQ2. Figure 6.2 presents the most frequent words used by an employer (E2). Common terms include *skill*, *training*, *student*, *industry*, and *tier*, indicating a focus on practical competencies and the educational background of applicants. It offers a brief overview of key topics raised by employers relevant to RQ1.

Figure 6.3 shows the aggregated code cloud created in MAXQDA, visualizing the frequency of thematic codes across all interviews and observations. The relative size of each code represents how often it appeared in the dataset, offering a concise overview of the main analytical categories identified during the thematic analysis.

### 6.1.4 Visualizing Code and Word Frequencies

To complement the tabular overview of codes, the following figures visualize the relative distribution of coded themes across interview participants and the frequency of key terms within the dataset. These visualizations help to identify which topics were most prominent and how the focus areas differed between graduates and employers.

As illustrated in Figures 6.4 and 6.5, the interviews revealed a strong focus on themes related to *skills*, *training*, and *curriculum*. Codes such as *Educational Preparedness* and *Hiring Process* show consistently high coverage across participants, while frequent words like *skill* and *aptitude* indicate the prominence of employability-related discussions. Together, these visualizations reinforce the analytical insights summarized in Table 6.1.

### 6.1.5 Theme Definitions

After coding and reviewing all interview transcripts and observation notes, recurring patterns were identified across the dataset. Codes with similar meanings or conceptual links were grouped into higher-level analytical categories through repeated comparison and refinement in MAXQDA. This iterative process led to the identification of four overarching themes that capture the central ideas emerging from the data.

Each theme functions as a higher-level category that combines several related codes and reflects key aspects of the participants' experiences. As summarized in Table 6.2, these themes correspond directly to the research questions (RQ1–RQ3) and illustrate how

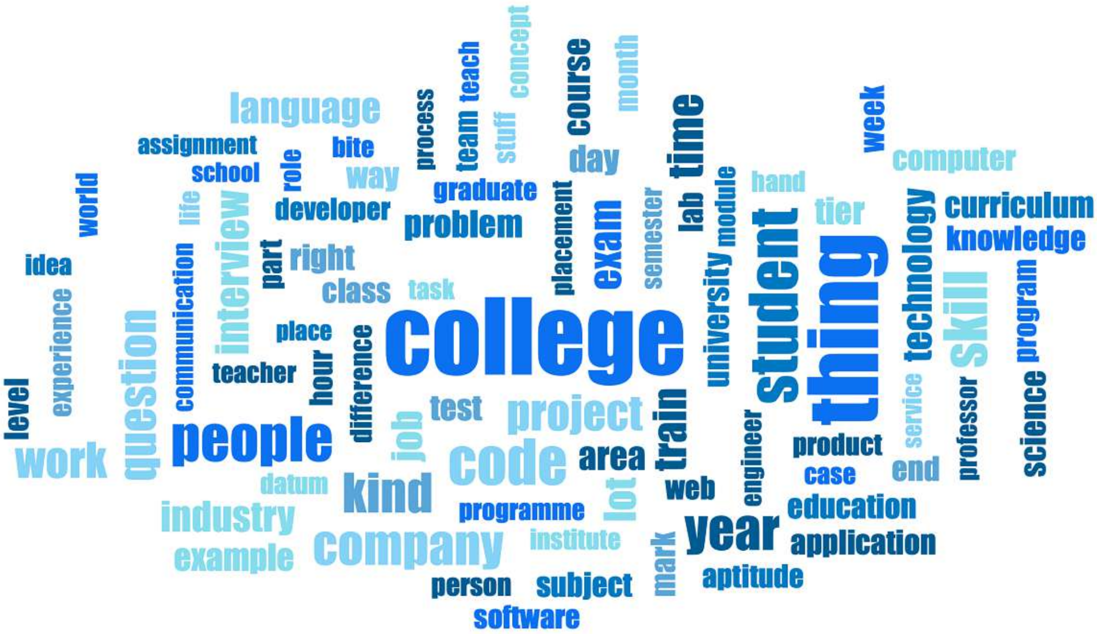


Figure 6.1: Word cloud from a graduate interview (Participant G3), illustrating themes related to college experience, self-learning, and improvement suggestions.

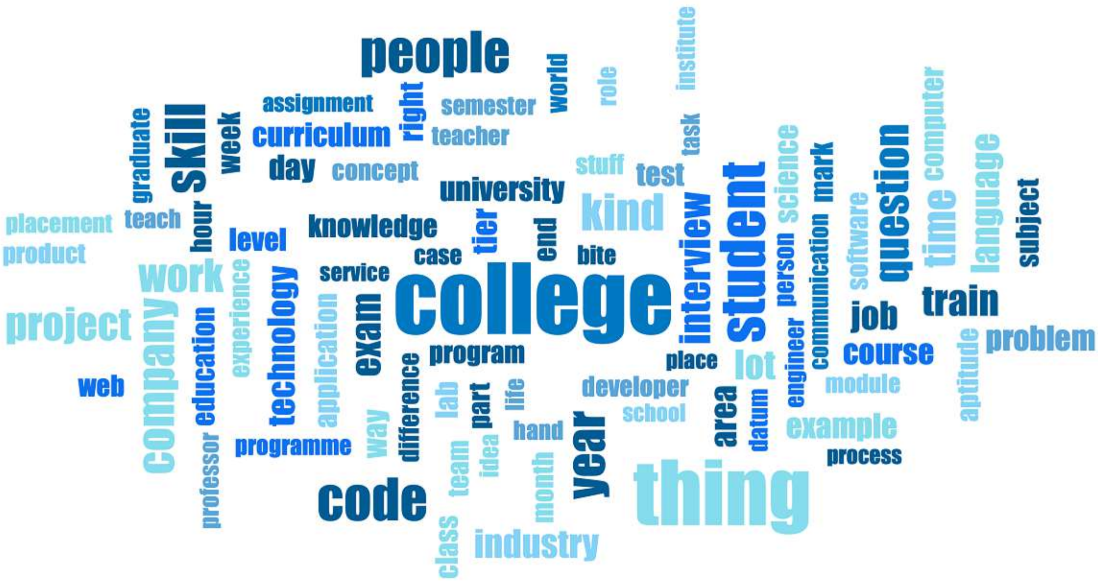


Figure 6.2: Word cloud from an employer interview (Participant E2), emphasizing terms linked to hiring process, skills, and workplace training.


Figure 6.3: Code cloud visualizing the frequency of analytical codes across all interviews and observations.

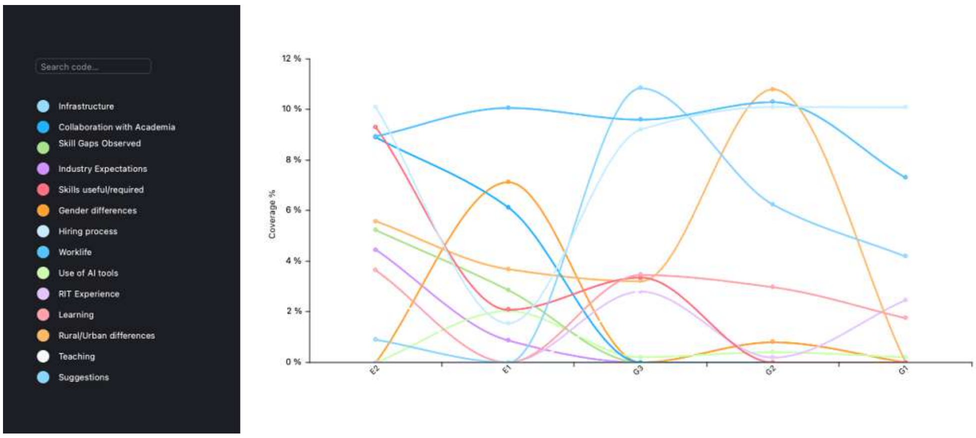


Figure 6.4: Code coverage across interview participants and data sources. Each line represents one code (e.g., *Skill Gaps Observed*, *Industry Expectations*, *Hiring Process*) and shows its percentage of coded text segments per participant. The figure highlights which themes were emphasized most strongly across the interviews.

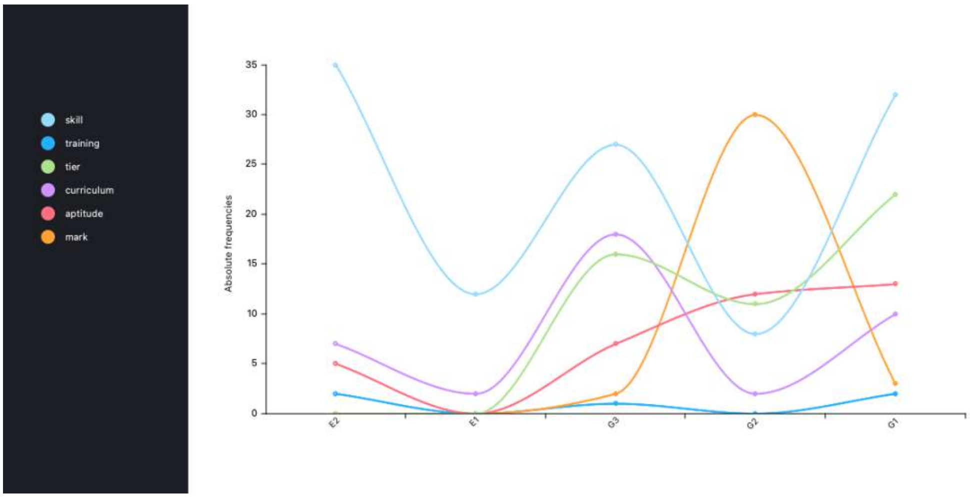


Figure 6.5: Word frequency visualization of recurring keywords across all interview transcripts. Words such as *skill*, *training*, *curriculum*, and *aptitude* appeared most frequently, underscoring their central role in discussions on employability and education. This visualization complements Figure 6.4 by showing the linguistic emphasis within the dataset.

industry expectations, educational experiences, and institutional factors interact within the rural Indian context. The following sections provide a detailed discussion of each theme, supported by representative quotations, field observations, and connections to existing literature.

### 6.1.6 Thematic Map and Interrelations of Themes

After grouping the codes into higher-level categories, four connected themes were developed to represent the main ideas that emerged from the data. As shown in Table 6.2, these themes relate directly to the research questions (RQ1–RQ3) and show how industry expectations, education, and institutional factors interact in the rural Indian context.

Figure 6.6 visualizes the relationships between these themes. The map shows how employer expectations (Theme 1) overlap with graduates' preparedness (Theme 2) within the broader institutional and social context (Theme 3). The integrative theme, *Bridging Academia and Industry* (Theme 4), connects all three and forms the basis for the curriculum recommendations discussed in Chapter 7.

This figure illustrates that the alignment between education and employability is not a simple process, but a dynamic interaction between people, institutions, and contextual conditions.

Together, these four themes form a clear analytical structure that connects the interviews and observation results with the insights from the ILR and the curricula insights. They

| Theme | Description and Analytical Focus | Associated Codes and References |
|---|---|---|
| **1. Industry Skill Expectations** | Represents the technical and interpersonal competencies most valued by employers. Shows how private-sector companies define employability, balance hard and soft skills, and evaluate candidates through aptitude tests or practical coding tasks. This theme directly relates to RQ1, outlining what the industry expects from computer science graduates in rural India. | **MAXQDA:** I6, I7, I9, I10, I3; **Observations:** O5, O8; **Curriculum:** C5; **ILR:** ILR–T1, ILR–T7, ILR–T9, ILR–T10, ILR–T11. |
| **2. Educational Preparedness and Skill Alignment** | Examines how graduates perceive their learning outcomes, self-directed skill development, and how well their competencies align with industry needs. Links classroom learning, online self-learning, and project experience to real-world expectations. Addresses RQ2 by identifying both skill matches and mismatches between education and employment. | **MAXQDA:** I13, I12, I1, I5, I11; **Observations:** O1, O2, O3, O6, O10, O12, O13, O14; **Curriculum:** C1, C2, C3; **ILR:** ILR–T2, ILR–T3, ILR–T6, ILR–T14. |
| **3. Institutional and Contextual Barriers** | Captures systemic, infrastructural, and socio-cultural factors that hinder effective curriculum implementation and graduate readiness. Includes issues such as outdated syllabi, rural–urban disparities, limited faculty development, and gender-related access differences. Relates to RQ2 and provides contextual input for RQ3. | **MAXQDA:** I8, I14, I1; **Observations:** O1, O2, O4, O7, O8, O9; **Curriculum:** C4; **ILR:** ILR–T5, ILR–T6, ILR–T13, ILR–T14ILR–T15. |
| **4. Bridging Academia and Industry (Integrative Theme)** | Serves as a cross-cutting theme that highlights collaboration mechanisms for closing identified skill gaps. Focuses on practical recommendations such as internships, joint curriculum reviews, and continuous dialogue between employers and educators. Provides the empirical foundation for RQ3, guiding the formulation of recommendations. | **MAXQDA:** I4, I12, I14; **Observations:** O11, O12; **Curriculum:** C3, C4; **ILR:** ILR–T4, ILR–T8, ILR–T12. |

Table 6.2: Refined Themes and Associated Codes, Linked Across Data Sources (MAXQDA, Observations, Curriculum, and ILR).

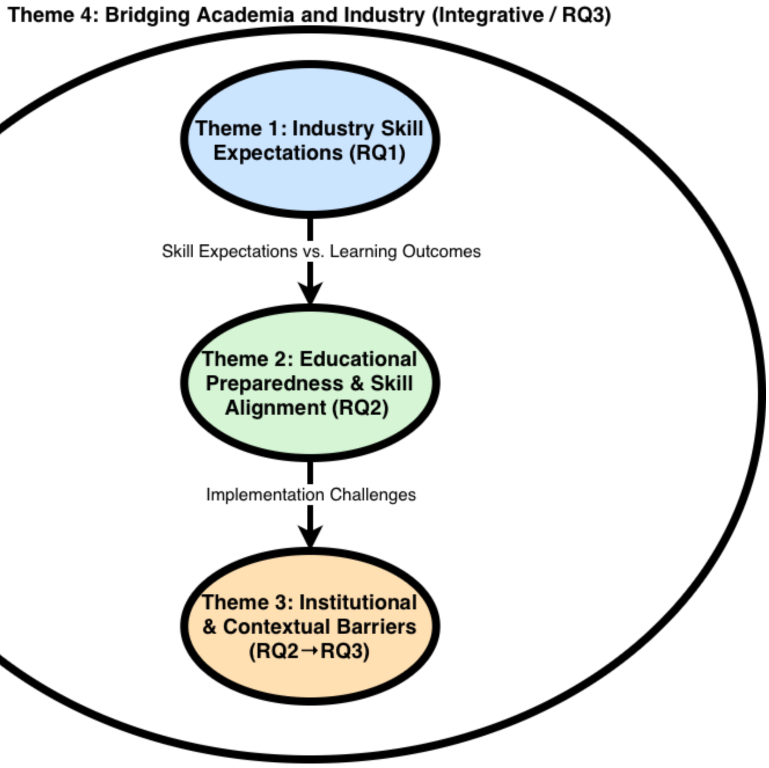


Figure 6.6: Thematic Map showing how the four themes relate to each other and to the research questions (RQ1–RQ3). Theme 1 (*Industry Skill Expectations*) describes employer demands, Theme 2 (*Educational Preparedness and Skill Alignment*) reflects graduates' skills and learning outcomes, Theme 3 (*Institutional and Contextual Barriers*) highlights the challenges that affect this alignment, and Theme 4 (*Bridging Academia and Industry*) links all themes and suggests ways to improve curricula.

help to bridge the gap between the raw data and the discussion of the research questions in Chapter 7.

## 6.2 Findings from Thematic Analysis

The thematic analysis resulted in four interconnected themes that directly relate to the three research questions (RQ1–RQ3). Each theme brings together evidence from interviews, field observations, the curricula insights, and the ILR, offering a broad understanding of how skill expectations, educational preparedness, and institutional factors shape graduate employability in the rural Indian context.

As shown in Figure 6.7, the theme *Educational Preparedness and Skill Alignment* accounts for the largest share of coded segments (57 %), followed by *Industry Skill Expectations* (27 %) and *Institutional and Contextual Barriers* (9 %). Only a smaller portion of codes

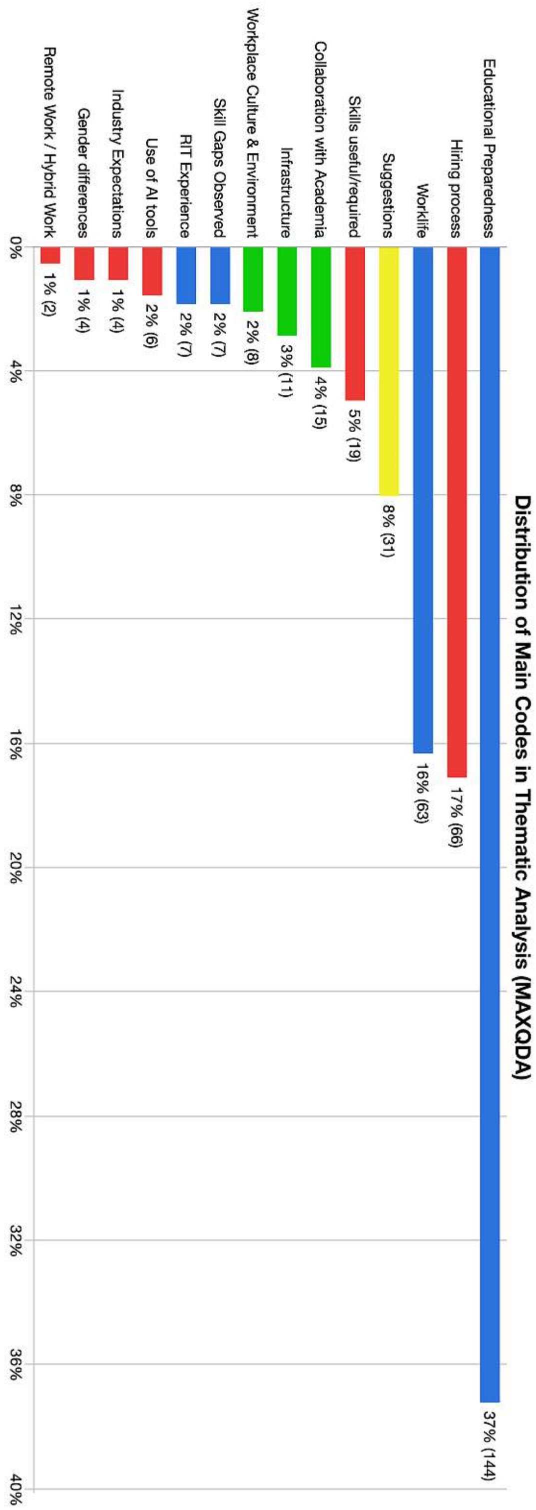


Figure 6.7: Distribution of main codes in the thematic analysis, color-coded according to the four overarching themes: Industry Skill Expectations (red), Educational Preparedness and Skill Alignment (blue), Institutional and Contextual Barriers (green), and Bridging Academia and Industry (yellow). The percentages indicate the relative frequency of coded segments across all interviews and observations.

(8 %) relates to *Bridging Academia and Industry*, indicating that collaborative and reform-oriented efforts are less frequently discussed among participants. This quantitative overview supports the qualitative interpretation that most employability challenges originate within the educational and institutional domains rather than external industry conditions.

This thematic synthesis connects the individual, institutional, and industry perspectives into a coherent framework for strengthening computer science education in rural contexts. The following sections explore each theme in greater depth, supported by illustrative quotations, coded examples, and cross-references to the corresponding ILR findings.

## 6.3 Detailed Analysis of Themes

This section presents a detailed thematic analysis of the empirical data, organizing the findings into core themes that capture recurring patterns across interviews, observations, and curricular documents.

### 6.3.1 Theme 1: Industry Skill Expectations

This theme represents the technical and interpersonal competencies most valued by employers. Consistent with ILR–T10 (*Workplace Readiness*) and ILR–T11 (*Professional Competence*), companies emphasized that employability depends not only on coding proficiency but also on soft skills such as adaptability, teamwork, and communication. However, when compared to the current curricular structure, a misalignment becomes visible. The analysis of the affiliating college curriculum revealed that learning outcomes remain focused on lower Bloom level (see Figure 5.2) (C5), with limited emphasis on the development of behavioural or interpersonal competencies [84]. Although the revised curriculum introduced several skill-oriented elements such as *Design Thinking & Innovation* and *Full Stack Development* (C3), these modules carry minor credit weight and are rarely embedded in a broader employability framework.

**Illustrative Quotes (edited for readability)** The overview of interview participants can be found in Table 5.3.

> “We look for graduates with strong fundamentals — data structures, databases, software engineering. Knowing two or three programming languages is enough if the conceptual base is strong. Industry expects people who can adapt to new tools like cloud or AI technologies quickly.” – *Employer E2*

> “Soft skills are very important in India because of our diverse workplaces. A positive attitude, open-mindedness, and good communication are key. We evaluate these during interviews rather than training them later.” – *Employer E2*

> "The hiring process had multiple rounds — aptitude, coding, and HR. They tested data structures and problem-solving, but in HR they checked how we would handle real situations, teamwork, and stress." – *Graduate G3*

> "In my interview, they gave me a small hands-on coding task with a short deadline. It was about logic and clarity of approach rather than just syntax." – *Graduate G1*

**Observation Notes** Observations O5 and O8 showed that many students aspire to work for international companies and that classroom teaching remains strongly teacher-centred, with limited opportunities for collaborative or problem-based learning. This environment reinforces the focus on exam-oriented performance rather than the active development of professional communication and teamwork abilities that employers demand.

**Interpretation** Employers consistently highlight the need for *hybrid skill profiles* that integrate solid technical foundations with interpersonal competence and adaptability. Learning many different programming languages was not seen as necessary — logical reasoning and solution-based thinking are prioritised by recruiters. This mirrors findings from ILR–T1 (*Industry–Academia Skill Gap*), ILR–T7 (*Hybrid Skill Profiles*) and ILR–T9 (*Soft Skill Integration*), which emphasise that employability emerges from a combination of knowledge, performance, and personal attributes. At the curricular level, the prevalence of theory-driven objectives (C5) and the absence of structured soft-skill assessment explain why graduates may excel in technical tests but underperform in behavioural evaluations. Even recent reforms (C3) have yet to translate into integrated, project-based approaches that build workplace readiness. Overall, this theme addresses **RQ1** by clarifying what the private sector values in computer science graduates and by exposing where the current academic design falls short of these expectations.

### Theme 2: Educational Preparedness and Skill Alignment

This theme explores graduates' perceptions of their learning outcomes and the degree to which their education prepared them for professional roles. It reflects on formal instruction, self-learning, and practical exposure, consistent with ILR–T2 (*Lack of Practical Training*), ILR–T6 (*Contextual Curriculum Reform*), and ILR–T14 (*Design–Reality Gap*). At the curricular level, these issues are mirrored in the strong theoretical orientation of the syllabus (C1) and the fragmentation of practical components (C2), where lab exercises are detached from real-world applications. Although the revised regulation introduced skill-oriented elements such as *Design Thinking & Innovation* and *Full Stack Development* (C3), their minor credit weight and limited pedagogical integration prevent them from compensating for the overall lack of experiential learning.

**Illustrative Quotes (edited for readability)** The overview of interview participants can be found in Table 5.3.

> "Our college teaching was mostly theoretical. We covered units, prepared for exams, and rarely applied anything to real projects. It gave me a degree that was necessary to get a job, but most of my actual knowledge I taught myself." – *Graduate G1*

> "Students were not encouraged to make mistakes or experiment. Coding practice was limited to short lab sessions once a week, which didn't help us learn effectively." – *Graduate G1*

> "So I would say that college classes were not that helpful, that they were only being taught or the classes were held only for the sake of the college examinations." – *Graduate G3*

> "In rural colleges, teaching is mostly theoretical. Companies expect practical knowledge, but students are not given enough opportunities to practice coding. It's like a recipe book — you can understand the recipe by reading it, but only when you actually cook do you get it right." – *Graduate G2*

> "In the first and second year, classes were often irregular — some teachers came, some didn't. Most of them weren't serious until the third year, when they suddenly pushed us to prepare for placements and learn coding languages like Python and Java. But even then, they just wanted us to memorize the code line by line instead of understanding it." – *Graduate G2*

> "The so-called hands-on training in college was very basic. Working on real industry projects would have been much more valuable." – *Graduate G3*

**Observation Notes** Frequent power and internet outages disrupted laboratory sessions, and students often shared or borrowed laptops to complete assignments. Faculty tended to emphasise discipline and attendance over creativity or experimentation, while students hesitated to ask questions or challenge instructors. Students reacted enthusiastically to mentoring and real-world projects, but were unable to complete simple practical coding assignments without direct assistance from mentors. Lectures and lab sessions focused predominantly on theoretical content, with limited opportunities for applied programming. (Observations O1, O2, O3, O6, O12, O13, O14, Campus and RIT Sessions).

**Interpretation** The interview and observational data reveal a strong emphasis on theoretical knowledge, memorization-based learning, and examination performance. Graduates report limited opportunities for hands-on experimentation, creative problem-solving, or failure-based learning. These conditions foster dependency and passivity rather than confidence and technical adaptability. While some students compensate through self-learning on platforms such as YouTube[1], Coursera[2], or HackerRank[3], this reflects individual resilience rather than institutional preparedness. This pattern also reflects ILR–Tt3 (*Onboarding Dependency*), where insufficient practical preparation during the degree program shifts the burden of skill development to employers' onboarding and training processes. The findings also align with ILR–T2 (*Lack of Practical Training*) and ILR–T14 (*Design–Reality Gap*), confirming that the curriculum's structure (C1–C3) reinforces rather than resolves this misalignment. Ultimately, the evidence underscores that skill acquisition depends more on students' self-initiative than on the formal academic design, directly addressing **RQ2** by evidencing the persistent gap between educational delivery and employability requirements. Moreover, students' capacity for self-learning is strongly shaped by their socio-economic background, as access to personal devices, stable internet, and quiet study environments is far more attainable for those from higher-income households, whereas students from lower-income families often lack these essential prerequisites.

#### Theme 3: Institutional and Contextual Barriers

This theme captures systemic and socio-cultural challenges that hinder effective education delivery and graduate readiness. It includes infrastructural limitations, hierarchical learning environments, and social norms that restrict participation and progress. The findings align with ILR–T13 (*Contextual Constraints*) and ILR–T14 (*Design–Reality Gap*), demonstrating that institutional capacity, faculty culture, and regional inequality influence how learning outcomes are achieved. At the structural level, these barriers are reinforced by the absence of formal evaluation and industry linkage mechanisms within the curriculum (C4), which limits continuous feedback and institutional learning. While policies such as the NEP 2020 call for stronger academia–industry collaboration, affiliated colleges remain dependent on top-down regulations from the affiliating university, leaving little room for local innovation or adaptation.

**Illustrative Quotes (edited for readability)** The overview of interview participants can be found in Table 5.3.

> "In rural colleges, students fear speaking up because internal grades depend on faculty approval. Questioning teachers or management can result in mark deductions." – *Graduate G2*

> "In urban colleges, students can challenge teachers and engage openly, but in rural colleges, hierarchical and emotional relationships prevent open discussion." – *Graduate G2*

[1] https://www.youtube.com/
[2] https://www.coursera.org/
[3] https://www.hackerrank.com/

> "My cousin studied in Chennai, one of the metropolitan cities in India. There, students are not afraid to speak up — even if they tell a teacher something is wrong, it does not affect their grades. But in rural colleges, it's the opposite. If you correct a teacher, they take it personally, as if you are being disrespectful. There's an inferiority complex and ego issues between teachers and students."
> – *Graduate G2*

> "Previously, companies hired only from accredited urban institutions. Since the National Qualifier Test was introduced, rural students also have access to national recruitment — though exposure gaps remain." – *Employer E2*

> "Rural students often lag behind in communication and confidence, but they are loyal, motivated, and eager to learn if guided properly." – *Employer E2*

> "Our software team consists entirely of married women working remotely. Flexibility helps them balance family duties while contributing effectively — they are highly reliable." – *Employer E1*

**Observation Notes** Multiple classes at the rural campus were interrupted by power or internet outages, forcing students to share laptops or complete assignments in groups. Faculty interactions often reflected a hierarchical tone—teachers emphasized discipline, attendance, and obedience over creativity and innovation. Students rarely questioned instructions and often avoided open disagreement to protect internal marks and avoid criticism to prevent offending teachers or peers. Several responses appeared overly favourable or adjusted to align with what students/faculty perceived authorities would want to hear. Gender norms were visible during group work and lab sessions, with male students taking initiative more often, while female students participated quietly (Observations O1, O2, O4, O7, O8, O9, Campus and Faculty Office).

**Interpretation** The data reveal a multilayered set of institutional and contextual barriers that extend beyond curriculum design. Limited infrastructure, unreliable power supply, and outdated equipment constrain practical learning. Hierarchical academic culture — where faculty authority is seldom questioned — discourages inquiry and experimentation, especially among rural students. Social hierarchies, including gendered expectations, further restrict equal participation, while institutional governance prioritises control and reputation over innovation and creativity. These barriers are compounded by the structural limitations of the affiliating university system, which lacks a formal mechanism for curriculum evaluation or industry feedback (C4). Such conditions mirror the broader design–reality gap identified in ILR–T14 (*Design–Reality Gap*) and the contextual inequities highlighted in ILR–T13 (*Contextual Constraints*). The theme intersects with **RQ2** and **RQ3**, emphasising that sustainable improvement requires not only curricular reform but also institutional empowerment, faculty development, and the implementation of inclusive, context-sensitive policies.

#### Theme 4: Bridging Academia and Industry (Integrative Theme)

This integrative theme synthesizes opportunities for collaboration between educational institutions and industry actors. It emphasizes practical cooperation mechanisms such as internships, curriculum co-design, joint training programs, and faculty engagement. The theme builds on ILR–T4 (*Curriculum–Industry Alignment*) and ILR–T12 (*Collaborative Learning*), addressing **RQ3** by identifying actionable strategies to close the skill gap between academia and the private sector. At the curricular level, the revised regulation introduced new modules such as *Design Thinking & Innovation* and *Full Stack Development* (C3), which reflect an initial attempt to incorporate employability-oriented content. However, the absence of structured industry partnerships or formal feedback mechanisms (C4) limits the long-term impact of these reforms. Without systemic coordination between academia and employers, such initiatives often remain isolated and dependent on individual faculty efforts rather than institutional frameworks.

**Illustrative Quotes (edited for readability)** The overview of interview participants can be found in Table 5.3.

> "I regularly visit rural universities to conduct sessions for faculty and students, sharing what the industry expects from graduates." – *Employer E2*

> "Collaborations with rural institutes have proven highly effective. We train students in advance and help them prepare for industry assessments, which increases their employability." – *Employer E2*

> "Through BITES, industry professionals volunteer to train students and faculty in emerging technologies, ensuring that curricula remain relevant." – *Employer E2*

> "Our collaboration with a local university allows us to mentor students on real projects. It's mutually beneficial — we get skilled support, and they gain practical exposure." – *Employer E1*

> "I suggest including real-world or open-source projects in the curriculum to better prepare students for corporate expectations." – *Graduate G3*

**Observation Notes** Faculty expressed interest in continuing such collaborations but cited time constraints and limited institutional funding. Students responded enthusiastically to external mentors, noting that exposure to real-world projects motivated them to improve their technical skills (Observations O11, O12, Campus and RIT Sessions).

**Interpretation** The data illustrate a strong willingness from both academia and industry to collaborate, yet structural barriers and uneven coordination limit the depth of engagement. Industry representatives emphasize early-stage partnerships, faculty training, and curriculum alignment as key to ensuring graduate readiness. Programs such as BITES[4] demonstrate scalable models where industry experts volunteer expertise across rural institutions, filling a gap left by formal curricular structures (C4). For graduates, practical learning opportunities — through internships, open-source participation, and corporate-led workshops — significantly enhance confidence and employability. While the inclusion of skill-oriented modules in the new curriculum (C3) reflects responsiveness to industry needs, their limited credit weight and lack of integration reduce effectiveness. The evidence thus confirms that effective bridges require institutional commitment, resource allocation, and sustained dialogue between educators and employers, rather than one-time interventions. Ultimately, these insights provide actionable guidance for higher education institutions to address the challenges highlighted in **RQ3**.

### 6.3.2 RQ1 – Industry Skill Requirements: Hard and Soft Skills Expected

Employers consistently emphasized that technical competence alone does not guarantee employability. In addition to solid *hard skills*—such as programming proficiency in Java, Python, or C++, as well as database management, and problem-solving through algorithmic thinking. Companies expect graduates to demonstrate a range of *soft skills*. These include effective communication, teamwork, adaptability, and self-motivation, which were often described as equally important during hiring decisions.

Interview excerpts from employers highlighted that candidates with practical experience, such as internships or project work, were perceived as better prepared to adapt to real work environments. This reflects the broader theme of *Industry Skill Expectations and Employability* (see Figure 6.6), which shows a clear consensus that employability relies on a balanced combination of technical expertise and professional behavior.

### 6.3.3 RQ2 – Alignment and Skill Gaps Between Graduates and Industry Requirements

Graduates expressed confidence in their theoretical knowledge but acknowledged difficulties in applying it to real-world situations. Employers confirmed this gap, pointing to limited experience with industry tools and insufficient problem-solving skills.

Codes such as *Educational Preparedness*, *RIT Experience*, and *Teaching Quality* (Table 6.2) indicate that many students rely on self-learning through online platforms to compensate for curricular shortcomings. Observation notes from classroom sessions showed a strong focus on lectures, with only limited opportunities for project-based learning or teamwork.

[4] https://www.bites.org.in

This misalignment results in recurring *skill gaps* in programming practice, debugging, and collaboration. At the same time, several positive *skill matches*—such as analytical thinking and persistence—were evident, reflecting a solid theoretical foundation.

The overlap between *Graduate Learning* and *Industry Skill Expectations* suggests that curriculum relevance remains the central link between education and employability. These findings are consistent with the results of the ILR, which identified similar patterns of hybrid skill expectations across industry-oriented studies.

### 6.3.4 RQ3 – Institutional and Curricular Recommendations

Based on the integrative theme *Bridging Academia and Industry*, three core recommendations emerged from the data:

1. **Strengthen experiential learning**
2. **Enhance faculty development**
3. **Foster structured industry–academia partnerships**

Participants agreed that sustainable improvement depends on both institutional flexibility and active student engagement. When combined with the insights from the ILR, these findings suggest that adapting global education models to local needs—rather than directly adopting them—is essential for overcoming the contextual challenges faced by rural institutions in India.

### 6.3.5 Synthesis Across Themes

The thematic analysis revealed four interrelated dimensions of graduate employability that together explain the alignment gap between higher education and industry needs in rural India. Theme 1 (*Industry Skill Expectations*) and Theme 2 (*Educational Preparedness and Skill Alignment*) illustrate a clear divergence between what employers value and what graduates acquire, particularly in the balance between technical proficiency and professional behaviour. Theme 3 (*Institutional and Contextual Barriers*) situates this misalignment within the broader socio-cultural and infrastructural realities of rural colleges, while Theme 4 (*Bridging Academia and Industry*) demonstrates pathways toward more sustainable collaboration.

Taken together, these findings suggest that skill gaps are not merely a product of individual performance or curriculum design but are embedded in structural, pedagogical, and relational factors. The interaction of these dimensions reinforces the conclusions drawn in the ILR, particularly those related to ILR–T10 (*Workplace Readiness*), ILR–T11 (*Professional Competence*), and ILR–T14 (*Design–Reality Gap*). The subsequent chapter integrates these insights to present the overall empirical *Results* and to connect them to the study's research questions (**RQ1**–**RQ3**).

CHAPTER 7

# Results

This chapter presents the empirical findings of the study, structured along the three research questions. Drawing on interview data, observation notes, and insights from the integrative literature review, the results first summarize industry skill requirements (RQ1), then assess the alignment between expected and observed graduate skills (RQ2), and finally derive recommendations and the validated guidance notes (RQ3).

## 7.1 RQ1: Industry Skill Requirements

**Research Question:** *What hard and soft skills do private sector software companies in India require from computer science graduates of rural colleges, exemplified by the region of Andhra Pradesh?*

### 7.1.1 Overview of Industry Expectations

Employer interviews, observations and literature reveal that employability is conceived as a combination of three interdependent dimensions:

1. *Technical competences / Hard Skills* — proficiency in programming, debugging, and understanding system logic,
2. *Professional behaviour / Soft Skills* — reliability, discipline, and client-oriented work ethics, and
3. *Adaptability and communication / Soft Skills* — the ability to integrate into diverse, distributed teams and learn new technologies quickly.

Both *Employer E1* and *Employer E2* emphasized that coding proficiency alone does not determine employability; rather, it must be complemented by collaboration, communication, and self-directed learning. These findings correspond to ILR–T10 (*Workplace*

*Readiness*) and ILR–T11 (*Professional Competence*), showing that successful graduates demonstrate both cognitive and behavioural readiness for modern workplaces. This observation echoes international literature identifying a demand for hybrid professionals who integrate technical and interpersonal capabilities [50], [51].

The graduate interviews provide additional insights into how these expectations manifest during job search and early career experiences. Participants described that the transition from college to employment exposed substantial gaps between academic preparation and real-world demands. Graduate G1, for instance, reported that recruiters evaluated candidates primarily through practical coding rounds on problems which look like tasks from platforms such as HackerRank[1] and LeetCode[2], focusing on data structures, algorithms, and debugging under time pressure. Others highlighted that successful placement required demonstrable projects or portfolio work beyond what the college curriculum provided. Graduate G2 recalled that even during onboarding, companies expected new hires to already possess hands-on skills in frameworks and version control systems, whereas college training had remained largely theoretical. Similarly, G3 emphasized that the ability to communicate progress, collaborate within a team, and adapt to company workflows was as critical as technical skill.

These graduate reflections align with the employers' perspective that formal education alone rarely suffices to prepare graduates for immediate productivity. The early job experiences of participants illustrate that companies expect newcomers to exhibit not only fundamental programming competence but also initiative, resilience, and a capacity for rapid, independent learning. This is consistent with the coded data in MAXQDA (I9.5, I9.6, I10.1), confirming that employers systematically evaluate both technical and behavioural dimensions of employability in parallel rather than separately.

### 7.1.2 Industry Skill Framework

As summarized in Table 7.1, the competencies most frequently mentioned by employers and graduates were synthesized into an extended skill model. It combines empirical insights from the interviews with ILR findings to illustrate how industry requirements extend beyond technical mastery.

### 7.1.3 Interpretation of Findings

The results demonstrate a consistent industry preference for graduates who combine solid technical foundations with effective communication and adaptive learning capacity. Practical exposure to real projects and teamwork scenarios was repeatedly cited as a strong indicator of employability. These insights empirically reinforce ILR–T7 (*Hybrid Skill Profiles*), showing that employability depends on the interplay of technical competence, behavioural maturity, and social collaboration.

[1] https://www.hackerrank.com
[2] https://leetcode.com

| Category | Skill / Competence | Source(s) |
|---|---|---|
| Hard Skills | Programming fundamentals (C, Java, Python) | E1, E2, G1–G3, O13, O12, ILR–T10, ILR–T11 |
| Hard Skills | Software testing and debugging | E2, G3, ILR–T1, ILR–T3 , ILR–T11 |
| Hard Skills | Applied tools and frameworks (Git, React, Angular, Docker) | G1–G3, E2, ILR–T7 |
| Hard Skills | Analytical and problem-solving ability | E1–E2, G2–G3, ILR–T2,ILR–T7 |
| Hard Skills | Understanding of software architecture and system logic | E2, ILR–T11 |
| Soft Skills | Communication and teamwork | E1–E2, O6, ILR–T10 |
| Soft Skills | Adaptability and willingness to learn new technologies | E2, G1–G3, O10, ILR–T7 |
| Soft Skills | Self-learning and continuous upskilling | G1–G3, E2, O10, O12, ILR–T7 |
| Soft Skills | Professional attitude and client interaction | E1, O6, ILR–T11 |
| Soft Skills | Reliability, discipline, and accountability | E1–E2, O3, O10, ILR–T11 |

Table 7.1: Industry Skill Expectations Derived from Employer and Graduate Interviews

Beyond the expected technical and communication competencies, three additional areas emerged strongly from the interviews:

1. **Applied Tools and Framework Literacy:** Graduates highlighted the need for hands-on familiarity with modern development environments such as Git, React, or Docker (G1–G3, E2). Their absence in curricula leads to extended onboarding times.
2. **Analytical and Problem-Solving Ability:** Employers stressed logical reasoning and structured thinking during hiring tasks—more than perfect code output (E1–E2, G3).
3. **Self-Learning and Continuous Upskilling:** All graduates and also Industry (E2) emphasized that success in industry depends on autonomous learning habits, consistent with ILR theme ILR–T7 (*Hybrid Skill Profiles*).

These findings are consistent with Spiesberger et al.'s identification of **Gap 1 (In-**

**complete Student Skills)** and **Gap 2 (Unmet Employability Goal)** in the same educational context [14]. They also resonate with Buerstmayr's recommendations advocating for project-based learning and continuous exposure to industry mentors [30].

**Reflection** The emphasis on employability as a blend of hard and soft skills mirrors findings by Spiesberger et al., who identified incomplete programming competence and communication deficits as primary design–reality gaps in the same educational context [14]. These results further substantiate ILR–T10 (*Workplace Readiness*) and ILR–T11 (*Professional Competence*), indicating that rural graduates face similar constraints despite modest institutional improvements. In line with Buerstmayr, who proposed increased exposure to project-based learning and industry mentoring, this study confirms the need for hybrid skill formation mechanisms that foster both technical proficiency and workplace adaptability [30].

Overall, private-sector employers in rural India seek "ready-to-contribute" graduates — those who can immediately participate in production environments with minimal supervision, bridging both knowledge and communication divides. These findings and the overview in Table 7.1 answer **RQ1** and provide the empirical foundation for the subsequent analysis of educational alignment in Section 7.2.

## 7.2 Findings Related to RQ2: Alignment of Skills

**Research Question:** *Do the skills of computer science graduates align with the requirements of the private sector in rural India, and if not, what specific skill gaps and skill matches can be identified?*

### 7.2.1 Comparison of Expected and Observed Skills

Graduate and employer perspectives were systematically compared to assess the extent to which the skills acquired in higher education align with private-sector expectations. Observation notes, interview data, and evidence from the ILR were triangulated to contextualize these findings within the institutional environment of rural Andhra Pradesh. To ensure analytical consistency, the same skill categories as in **RQ1** were applied and can be found in Table 7.1. The strength of alignment (*high, medium, low*) was assessed through qualitative weighting based on:

1. Recurrence across independent sources (graduates, employers, observations),
2. depth and emphasis of discussion in coded segments, and
3. cross-references with relevant ILR themes.

This interpretive procedure follows the principles of thematic salience described by Braun and Clarke and Fereday and Muir-Cochrane, where prevalence reflects analytic importance rather than numerical frequency [13], [88]. The resulting synthesis is summarized in Table 7.2. Table 7.3 defines the weighting levels applied during the thematic analysis, while Table 7.4 explains the alignment labels used to assess the correspondence between graduate skills and industry expectations.

### 7.2.2 Interpretation of Findings

The comparative analysis reveals a limited alignment between educational outcomes and industry expectations. Contrary to initial assumptions, programming fundamentals are not well established through the regular curriculum. Graduates (e.g. G1, G2, G3) reported that college labs covered only very basic exercises such as adding or multiplying two numbers, which did not prepare them for real-world development tasks. Most students who later succeeded in programming-oriented roles did so through intensive self-learning or by participating in a external training program, which provided the only structured hands-on exposure to real coding projects. This finding confirms O13 and O14, indicating that theoretical instruction dominates classroom learning while practical competence must be acquired independently.

**Skill gaps** remain most evident in applied programming, analytical problem-solving, and software testing, where graduates require external coaching or job-based training to reach employable standards. Employers highlighted that graduates possess conceptual awareness but lack the ability to translate it into executable code (E2). Observation data likewise show that laboratories are under-equipped and exercises are repetitive rather than exploratory, reinforcing rote learning instead of applied problem-solving (O13, O14).

**Skill matches**, by contrast, were found in self-learning motivation and teamwork, where students exhibit strong perseverance and adaptability. Several graduates completed online certifications and external courses (G1, G3), demonstrating initiative to bridge institutional shortcomings.

Overall, these results underscore that employability in rural India is co-determined by individual self-learning efforts and supplementary programs like the RIT initiative, rather than by the college curriculum itself. This substantiates ILR–T2 (*Lack of Practical Training*) and ILR–T14 (*Design–Reality Gap*), highlighting that systemic curricular deficits, rather than student ability, cause the persistent misalignment between education and industry needs.

Taken together, the comparison of industry expectations and graduates' demonstrated competencies reveals substantial misalignment, particularly in core programming, problem-solving, and practical application skills. Table 7.2 synthesizes these gaps and highlights only limited areas of partial alignment in soft skills. These findings answer RQ2 and form the empirical basis for the subsequent development of recommendations and guidance in Section 7.3.

| Skill | Expected by Industry | Demonstrated by Graduates | Alignment | Evidence Source(s) |
|---|---|---|---|---|
| Programming fundamentals (C, Java, Python) | High | Low[1] | Gap | G1–G3, E2, O13, O12, ILR–T10, ILR–T11 |
| Software testing and debugging | High | Low | Gap | E2, G3, ILR–T1, ILR–T3, ILR–T11 |
| Applied tools and frameworks (Git, React, Angular, Docker) | Medium–High | Low–Medium | Partial Gap | G1–G3, E2, ILR–T7 |
| Analytical and problem-solving ability | High | Low | Gap | E1–E2, G2–G3, ILR–T2, ILR–T7 |
| Understanding of software architecture and system logic | Medium–High | Low | Gap | E2, ILR–T11 |
| Communication and teamwork | High (communication), Medium (teamwork) | Partial (communication), Medium (teamwork) | Partial Gap | E1–E2, O6, ILR–T10 |
| Adaptability and willingness to learn new technologies | Medium | High | Match | E2, G1–G3, O10, ILR–T7 |
| Self-learning and continuous upskilling | Medium | High | Match | G1–G3, E2, O10, O12, ILR–T7 |
| Professional attitude and client interaction | Medium | Medium | Match | E1, O6, ILR–T11 |
| Reliability, discipline, and accountability | Medium | Medium–High | Match | E1–E2, O3, O10, ILR–T11 |

[1] Interviewed graduates in the sample had participated in external practical programming training which is not representative of the typical graduate profile observed. This is a limitation when assessing overall graduate preparedness.

Table 7.2: Skill Gap and Match Matrix for Graduates and Industry (**RQ2**)

| Weighting Level | Description |
|---|---|
| **High** | Frequently mentioned, strongly emphasized, and triangulated across multiple sources (graduates, employers, observations, ILR themes). |
| **Medium** | Mentioned regularly but with lower interpretive depth or fewer supporting sources. |
| **Low** | Rarely mentioned or limited to a single source group. |
| **Partial Match** | Competency is present, but insufficiently developed in depth, consistency, or transferability relative to industry expectations. |

Table 7.3: Definition of weighting levels used in the thematic analysis

| Alignment Label | Description |
|---|---|
| **Match** | The skill demonstrated by graduates aligns well with industry expectations in both level and consistency. |
| **Partial Gap** | Graduates show elements of the expected skill, but with notable limitations that reduce workplace readiness. |
| **Gap** | The expected industry skill is largely absent or insufficiently demonstrated by graduates, indicating a substantive misalignment. |

Table 7.4: Alignment labels describing the degree of correspondence between graduate skills and industry expectations

## 7.3 RQ3: Recommendations for Curriculum Enhancement

**Research Question:** *What specific recommendations can help higher education institutions (HEIs) in rural India implement computer science curricula that effectively address the skill gaps identified in RQ2, while also enhancing the identified skill matches?*

### 7.3.1 Synthesis of Themes and Gaps

Building on the integrative theme *Bridging Academia and Industry*, the findings indicate that improving employability in rural contexts requires coordinated action at three levels: institutional, pedagogical, and individual. The recommendations derived from both empirical data and the ILR converge on the following priority areas:

- **Curriculum design and pedagogy** — make courses more practical, problem-oriented, and teamwork-focused, replacing purely theoretical or exam-driven instruction;

| Theme | Recommendation | Level | Stakeholder(s) |
|---|---|---|---|
| Curriculum Alignment | Redesign/Teach courses to include practical, problem-based assignments and collaborative projects that simulate real software development tasks. | Course | Faculty |
| Soft-Skill Integration | Integrate soft-skill components—such as communication, teamwork, and project coordination—into technical subjects rather than as standalone modules. | Course | Faculty + Students |
| Industry Collaboration | Establish partnerships with IT firms for joint workshops, short-term internships, and work-and-study initiatives that enhance applied learning opportunities. | Regional | HEI + Industry |
| Continuous Learning Culture | Encourage participation in online courses, peer-learning groups, and open-source projects to strengthen self-directed learning habits. | Individual | HEI + Students |
| Infrastructure Support | Upgrade computer labs and ensure stable internet connectivity to support hands-on coursework and virtual collaboration. | Institutional | HEI + Government |

Table 7.5: Overview of Recommendations for Curriculum Enhancement

- **Soft-skill development** — embed communication, collaboration, and client-interaction training directly into technical courses;
- **Institutional collaboration with industry** — promote regular workshops, work-and-study programmes, and joint skill-development initiatives between HEIs and employers.

As summarized in Table 7.5, the following set of recommendations integrates the empirical findings from graduate and employer interviews with insights from the ILR.

*Derivation logic:*

- Recommendations integrate empirical insights from **Theme 4** (*Bridging Academia and Industry*) and ILR findings (ILR–T4, ILR–T8, ILR–T12, ILR–T13).
- The classification by *level* reflects the scale of implementation (institutional, course, or individual level).
- The *stakeholder* column identifies the primary actors responsible for each recommendation.

Overall, these recommendations emphasize practical relevance, collaborative learning, and sustained engagement between academia and industry, rather than large-scale infrastructural reforms. They recognize that effective curriculum enhancement in rural India depends less on adopting international models and more on developing **locally adapted, partnership-based solutions** and keep contextual constraints in mind as found in ILR–T13.

### 7.3.2 Discussion and Implications

These recommendations operationalize **Theme 4** by linking educational reform directly to sustained collaboration with employers. They show that long-term improvement in employability depends on institutional flexibility, faculty development, and active student participation. The findings align with integrative insights from the ILR, particularly ILR–T4 (*Curriculum–Industry Alignment*) and ILR–T12 (*Collaborative Learning*), which emphasize that curricular reform must proceed alongside industry co-creation and continuous professional development. This perspective also corresponds to India's *National Education Policy (NEP 2020)*, which promotes experiential learning, flexibility, and interdisciplinarity in higher education [68].

Empirical evidence from employer interviews (E1–E2) and graduate reflections (G1–G3) supports these strategies: employers expressed willingness to collaborate with rural colleges, while graduates emphasized the importance of hands-on and project-based learning. Observation data from external teaching sessions further confirmed that student motivation increased noticeably when external mentors and real-world tasks were included.

A central output of this research is the translation of the recommendation set in Table 7.5 into actionable guidance notes for CSE faculty at the end of this chapter 7.7. Table 7.5 served as the analytical foundation from which each Guidance was derived, prioritised, and operationalised. While the table summarises the core strategic directions emerging from the findings, the guidance notes transform these directions into concrete, context-sensitive instructions for faculty and curriculum designers at rural affiliated colleges.

**Answer to RQ3** In summary, employability-oriented curriculum enhancement in rural India requires a **three-level strategy**: (1) *Institutional partnerships* that link HEIs with industry, (2) *pedagogical reforms* emphasizing project-based and collaborative learning, and (3) *individual capacity-building* that strengthens continuous self-learning. Together, these measures **address the skill gaps identified in RQ2** while reinforcing existing strengths in programming knowledge and adaptability. This conclusion is grounded in the patterns across the interviews, observations, and the ILR. Employers and graduates consistently highlighted that meaningful improvements arise when reforms reflect local labour-market realities, involve stakeholders directly, and are introduced in small, feasible steps rather than large structural changes. These converging insights support the emphasis on locally grounded, participatory, and incremental reform approaches.

Out of these recommendations guidance notes for CSE faculty was developed, evaluated in a focus group and refined. These guidance notes can be found at the end of this chapter 7.7.

## 7.4 Evaluation of the Guidance Notes (Focus Group Discussion)

To assess the practical relevance, clarity, and applicability of the guidance notes, a focus group discussion as described in Section 1.4.7 with experts in ICT4D and higher education was conducted. The purpose of this evaluation was not to collect new empirical evidence for the core research questions, but to validate and refine the proposed Guidance Note before finalization.

### 7.4.1 Focus Group Procedure

The focus group took place at TU Wien and brought together five experts with significant experience in ICT4D initiatives, curriculum development, and computer science education in low-resource environments. The participants were selected purposively based on their domain expertise and familiarity with the challenges faced by institutions similar to the case study college. The group consisted of individuals who were already acquainted with one another, which supported an open and constructive atmosphere.

The session lasted approximately two hours and followed a semi-structured format. After a short introduction, the discussion proceeded along three evaluative dimensions:

- **Clarity:** Are the guidance notes conceptually clear, well structured, and understandable for faculty at rural colleges?
- **Feasibility:** Are the recommendations realistic given the typical resource constraints, staffing structures, and pedagogical practices?
- **Transferability:** Are the suggested measures applicable to other affiliated or rural institutions beyond the case study context?

The author moderated the discussion, introduced each Guidance, and facilitated the exchange of views while ensuring that all participants had the opportunity to contribute.

No audio recordings were made. Instead, detailed written notes were taken directly in the working draft of the guidance notes to document comments, suggestions, and illustrative examples.

### 7.4.2 Findings

**(1) Clarity and Comprehensibility.** Participants agreed that the overall structure of the guidance notes is clear and logically ordered, while all agreed that scientific references are too technical for the context of a Guidance Note and were therefore removed. The modular organisation into discrete recommendations was perceived as intuitive and helpful for practitioners who may consult only specific sections. The examples embedded after each recommendation were highlighted as particularly valuable, as they translate abstract ideas into concrete, classroom-oriented actions, more examples emerged in the discussion and were integrated into the final version. Some participants noted, however, that a few formulations—especially those involving pedagogical terminology such as flipped classroom or project-based learning—may require more explanation in form of resources in further reading to ensure accessibility for faculty with limited formal training in educational methods. It was also recommended to further tighten the introduction by explicitly stating the purpose, target audience, and intended use of the guidance notes as well as possible problems which can be tackled by the Guidance Note. Overall, the guidance notes were considered readable and appropriate for educators in rural and low-resource contexts.

**(2) Feasibility and Applicability.** The experts generally agreed that the recommendations are feasible within the typical constraints of affiliated rural colleges, provided they are implemented incrementally. Changes had to be made in the language in regards who is adressed and what changes can faculty make, because many topics rely on managment or institutional level. Low-cost and no-cost elements—such as integrating short hands-on tasks, facilitating peer presentations, or adopting Git for simple class exercises—were regarded as realistic first steps. Concerns were raised about the variability of infrastructural conditions across institutions. Participants emphasised that some suggestions (e.g. extended computer lab access, alumni collaboration, micro-projects with industry) require administrative support that may not be uniformly available. Nevertheless, they appreciated that the guidance notes maintain a pragmatic tone and repeatedly emphasise adaptation to local constraints.

**(3) Scalability and Transferability.** The group agreed that the guidance notes address challenges that may be common across many rural Indian affiliated colleges, such as large class sizes, limited practical exposure, and restricted access to industry networks but the group was limited to their experience in only one college. Therefore, the content was considered only transferable to institutions facing the problems which are listed in the target audience section. In summary, the experts viewed the guidance notes as scalable across diverse affiliated colleges, provided that local adaptation remains an integral part of their implementation.

### 7.4.3 Implications for the Guidance Notes

Based on the feedback, several implications emerged for refining the guidance notes:

- **Strengthen accessibility:** Add explanatory literature for pedagogical terms and reduce abstract wording where possible.
- **Emphasise limitations for faculty:** Adapt guidances to clarify which actions are feasable to implement by faculty and which steps can only be encouraged and rely on the management
- **Highlight local adaptation:** Explicitly reinforce that all recommendations should be tailored to institutional realities.
- **Clarify and expand examples:** Strengthen the examples in sections on practical programming and soft-skill development to emphasise low-cost, actionable classroom practices.

## 7.5 Changes to first Version of Guidance Notes

Following the focus group evaluation, several targeted changes were made to improve clarity, feasibility, and transferability:

- Refined the introduction to more clearly outline the purpose, scope, problems and target audience.
- Restructured Further Reading section, by grouping resources by topic
- Expanded examples in the programming, soft skills, and industry collaboration sections to illustrate concrete low-cost implementation steps.
- Added clarification that all recommendations should be adapted to local infrastructural and organisational realities.

Together, these adjustments enhanced the usability and practical relevance of the guidance notes for educators in rural and resource-constrained higher education environments.

## 7.6 Synthesis of Findings and Discussion

The integrated analysis across **RQ1–RQ3** reveals that the skill gap in rural computer science education is multidimensional and shaped by several interacting factors:

- **Graduate Level:** Strong theoretical knowledge but limited practical and communication skills.
- **Institutional Level:** Outdated curricula, insufficient faculty training, and rural–urban disparities in resources and exposure.
- **Industry Level:** Growing demand for hybrid skill profiles that combine technical ability with professional readiness.
- **Bridging Level:** Emerging opportunities through external teaching programs, online learning platforms, and early-stage collaborations with industry.

Taken together, the findings show that employability is influenced not by individual skills alone, but by the interaction between curriculum design, institutional culture, and industry engagement. This synthesis highlights the interdependence of the four themes and forms the empirical basis for the concluding chapter, which discusses the study's key contributions, practical implications, and directions for future research.

## 7.7 Guidance Note for CSE Faculty and Trainers

The **final version** of the guidance notes is presented in this section as an integral part of the results for RQ3. For transparency of the evaluation process, the **initial version (v1)** and a **diff document visualizing the changes between v1 and the final version** are provided in the Appendix (see Appendix I and Appendix J).

The diff document visualizes the changes between the initial and final versions of the guidance notes. Due to layout reflow and extensive textual revisions, the overlay-based visualization may appear dense and is therefore not intended for continuous reading. Its purpose is to deliberately highlight the extent and nature of the evaluation-driven revisions and to provide a transparent visual trace of the refinement process, rather than to serve as a line-by-line readable comparison.

Therefore, the following pages contain the validated and application-ready version of the guidance notes.

**INSO - Industrial Software**
Institute of Computer Aided Automation | Faculty of Informatics | Vienna University of Technology



# Guidance Note

## Guidance Note for Computer Science Faculty at Rural Higher Education Institutions in India

### Enhancing Employability through Curriculum and Teaching Alignment

Developed by: Johannes Hufnagl, Paul Spiesberger, Thomas Grechenig
December 2025

# 1 Context

Computer science graduates from rural Indian colleges continue to enter the job market with significant skill gaps—particularly in programming, problem solving, communication, and practical software engineering. This mismatch between academic training and industry expectations results in low employability and high post-hire training costs for companies.

**Caveat:** Solutions to problems differ and heavily depend on the context of each Higher Education Institution and rely on factors such as technical resources as well as faculty know-how. The here stated solutions can only be seen as proposals and require constant on site evaluation and adaptation.

## 1.1 Computer Science Education Faculty and Target Audience

The primary audience of these notes are lecturers, trainers, and assistant professors teaching Computer Science Education courses in affiliated colleges under state universities. Although a curriculum is predefined, faculty members have the autonomy in terms of teaching methods, assessments, and student engagement. Their pedagogical decisions strongly influence the employability and confidence of graduates.
These Guidance Notes aim to help educators:

- strengthen the link between theoretical instruction and practical application;
- foster employability skills and professional behaviour;
- establish sustainable collaboration with industry partners;
- and want to teach programming in a more practical way

Lecturers, trainers, and assistant professors teaching Computer Science who face the following challenges may find practical solutions in these Guidance Notes:

- struggling to translate theoretical content into hands-on programming exercises and meaningful lab work;
- noticing that students lack basic coding proficiency and problem-solving skills needed for industry-style tasks;
- having limited exposure to modern software engineering tools and workflows (e.g. Git, testing, collaborative development);
- seeking simple, actionable ways to foster communication, teamwork, and professional behaviour in class.

## 1.2 Scope and Adaptation

The recommendations are based on an affiliated college in rural Andhra Pradesh, India as well as insight from industry and literature and reflect the conditions of similar Higher Education Institutions in India with limited resources and large class sizes. Therefore, they should be adapted to local institutional realities and regularly reviewed with students, faculty, and industry partners.

## 2 Guidance

The following recommendations focus on feasible, low-cost actions that Computer Science Education faculty can implement within existing institutional constraints.

1. **Strengthen hands-on programming and applied learning.** Faculty should systematically connect theory to practice by integrating small programming projects, group exercises, and open-ended assignments into laboratory work. In doing so, faculty are encouraged to move away from purely lecture-based (frontal) teaching and adopt more student-centered approaches such as flipped classroom and project-based learning, or at minimum a blended combination of these methods.

   *Example:* Short exercises (e.g. loops, conditionals, arrays) should be introduced with small hands-on tasks that students implement during the class and then be integrated into a single cumulative semester project.

   *See further reading: 1.1, 1.2, 1.4, 1.8, 1.10 .*

2. **Show students modern tools and agile workflows.** To better prepare them for industry expectations, classes should additionally incorporate essential software engineering tools and workflows, including version control (e.g. Git[1]), collaborative development practices (e.g. pair programming and code reviews), basic testing, and the use of issue trackers or simple ticket systems.

   *Example:* During the semester integrate agile workflows and a version control system into the single cumulative semester project.

   *See further reading: 1.11, 2.1, 2.2, 2.3 .*

3. **Foster soft skills and professional readiness.** Communication, teamwork, and adaptability are key hiring criteria. Include presentations, peer feedback, and mock interviews to simulate professional environments and give students repeated practice in speaking and collaborating.

   *Example I:* At the end of each hands-on programming lesson, ask student teams to give a 5-minute project pitch followed by short peer feedback on clarity, structure, and teamwork.

   *Example II:* Let students present/teach certain topics to their peers as in a flipped classroom scenario.

   *See further reading: 1.3, 1.4, 1.8, 1.12 .*

4. **Promote industry collaboration.** Advocate partnerships with local IT firms, alumni, or tech experts to organize guest lectures, workshops, and mentoring. Small collaborations can already increase institutional visibility and student employability.

   *Example I:* Students may work in small teams on *applied mini-projects* for local companies ("micro research projects"), giving firms early access to talent while providing students with real-world experience.

   *Example II:* Faculty can facilitate *micro consulting sessions*, where external experts review student work and help build a basic expert network around the institution.

   *Example III:* Invite one alumnus per semester to discuss recruitment and skill expectations and combine this with a small, company-inspired student project.

[1] `https://git-scm.com`

*See further reading: 1.4, 1.11, 2.3 .*

5. **Continue self-learning.** Faculty development strengthens teaching quality and keeps subject knowledge up to date. In addition to online courses (e.g. NPTEL[2], Coursera[3]) and peer exchange, faculty are encouraged to not only teach theory, but also implement all assigned tasks themselves. Share teaching materials and know-how between other faculty members. Familiarize yourself with Open Educational Resources[4] to use and share learning material with your peers.

   A combined *learning and software development circle* can support this: faculty jointly complete online modules and also implement the same programming exercises and projects given to students.

   *Example:* Establish a small faculty circle that completes one NPTEL course per year and regularly meets to review code, discuss challenges, share material and align teaching practices.

   *See further reading: 1.5, 1.6, 1.14, 2.1 .*

6. **Support student self-learning.** Motivate students to engage in self-paced online learning such as HackerRank[5] or LeetCode[6] to improve their applied problem-solving skills and recognize certificates or project work as part of internal assessments where appropriate. Students learn most effectively when they can experiment and try things out themselves; this requires adequate *time* and *access to resources*. Ask your institutions if it is possible that computer labs are open beyond class hours so students have the opportunity to explore, practice, and "play" with software development tools, but also allow this to happen in your classes. Provide guidance on selecting high-quality resources and balancing extra learning with workload.

   *Example I:* Allow students to submit one online course or coding-platform project per semester (plus a short reflection) as a small share of their internal marks.

   *Example II:* Encourage senior students to interact and teach their juniors.

   *See further reading: 1.5, 1.6, 1.7, 1.9, 2.1 .*

7. **Integrate ethical and social awareness.** Discuss ethics, data privacy, and the societal implications of technology in class, using examples that are locally relevant to students' lives. Promote a *Free and Open-Source Software (FOSS) mindset* by encouraging students to use open-source tools, share code openly, and contribute to or start small FOSS projects; this not only supports ethical and transparent development practices but also simulates real collaborative "company-like" workflows and can have positive impact on the placement quote. Encourage student participation in technical local digital literacy or community projects to strengthen social responsibility.

   *Example:* Show students an open source project (on e.g. GitHub[7]) and solve a simple issue together.

   *See further reading: 1.11, 1.13, 1.14 .*

[2] `https://nptel.ac.in`
[3] `https://www.coursera.org`
[4] `https://en.wikipedia.org/wiki/Open_educational_resources`
[5] `https://www.hackerrank.com`
[6] `https://leetcode.com`
[7] `https://github.com`

8. **Use AI tools as supportive, transparent learning aids.** Treat AI-based tools (e.g. code assistants or chatbots) as complements to, not replacements for, core learning activities. Explain to students that AI tools are especially helpful for understanding code, debugging, and exploring alternative solutions—but not for replacing hands-on practice. Clearly define when AI use is allowed, ask students to document prompts and outputs, and emphasise that AI-generated results must always be checked and understood.

   *Example:* In programming courses, allow students to use an AI assistant for debugging, but require them to submit their prompts, explain which suggestions they kept or rejected, and verify the final code with tests.

   *See further reading: 3.1, 3.2, 3.3 .*

# 3 Further Reading

For additional context and theoretical background, grouped into thematic fields:

## 3.1 Teaching Programming and Course Structure

1.1 **Raspberry Pi Foundation – Teaching Resources.** A large collection of free, classroom-ready programming projects (Python, Scratch, physical computing), including full units and step-by-step project sheets suitable for resource-constrained environments. `https://www.raspberrypi.org/teach/`

1.2 **Experience-CS Curriculum.** A project-based computing curriculum with concrete examples of semester projects, learning trajectories, and rubrics for CS education. Helpful for designing integrated, cumulative programming projects. `https://experience-cs.org/`

1.3 **What schools don't teach (Code.org).** A short and influential video motivating why programming education matters; ideal for motivating students and faculty. `https://www.youtube.com/watch?v=0xS68sl2D70`

1.4 **Teaching Methods Overview (Teach.com).** A practical introduction to common teaching strategies including collaborative learning, inquiry-based learning, flipped classroom, and project-based approaches. `https://teach.com/what/teachers-know/teaching-methods/`

1.5 **W3Schools Academy.** Beginner-friendly, interactive tutorials for HTML, CSS, JavaScript, Python, SQL, and more. Ideal for foundational programming labs. `https://www.w3schools.com/academy/academy_get_started.php`

1.6 **FreeCodeCamp.** A comprehensive, fully free, project-based curriculum covering full-stack development. Includes certificates and large practical assignments. `https://www.freecodecamp.org`

1.7 **CS Unplugged.** A widely used collection of activities to teach computing concepts without using computers—especially useful for limited lab availability. `https://www.csunplugged.org/en/`

1.8 **Flipped Classroom Explained Simply (YouTube).** A short and clear visual explanation of how the flipped classroom model works, why it improves student engagement, and how faculty can gradually adopt it even with limited resources. `https://www.youtube.com/watch?v=qdKzSq_t8k8`

1.9 **Scratch, PocketCode and visual programming tools.** Beginner-friendly environments that help students internalise algorithmic thinking, sequencing, logic, and event-driven programming before transitioning to text-based languages. Useful for first-year students with limited prior exposure. https://scratch.mit.edu/, https://catrob.at/pocketcode

1.10 **MIT App Inventor.** A block-based app development environment that enables students to build Android apps in weeks. Ideal for mini-projects and practical labs. https://appinventor.mit.edu

1.11 **GitHub Education.** A globally accessible programme providing free access to developer tools, cloud platforms, and classroom management features (GitHub Classroom). Extremely relevant for integrating professional workflows (Git, repositories, reviews) into teaching. https://github.com/education

1.12 **Essential Soft Skills for Software Engineers (GeeksforGeeks).** A concise overview of communication, teamwork, adaptability, problem-solving, and time management skills that are crucial for employability.
https://www.geeksforgeeks.org/blogs/essential-soft-skills-for-software-engineers/

1.13 **What is Free and Open-Source Software? (OpenProject Blog).** An accessible explanation of FOSS principles, why open-source software matters, and how sustainable business models around FOSS work. Supports teaching ethical, transparent, and collaborative software development practices. https://www.openproject.org/blog/what-is-free-and-open-source-software-and-why-it-costs-money/

1.14 **Open Educational Resources (OER).** OERs are learning material licensed under the Creative Commons License[8] and allow everyone to use, study, adapt and share learning materials. See UNESCO's recommendations for OERs: https://www.unesco.org/en/legal-affairs/recommendation-open-educational-resources-oer Connect to international and local OER repositories and communities such as https://openverse.org/ and https://oerhub.at/en.

## 3.2 Programming Tools and Workflows

2.1 **LabEx Online Coding Playgrounds.** Browser-based practice environments for Git, Linux, Python, Java, SQL, and algorithm problems—no installation required. https://labex.io

2.2 **GeeksforGeeks – Software and Tools A–Z List.** A broad catalogue of commonly used software tools across programming, DevOps, databases, cloud platforms, and more. Helpful for faculty looking to introduce students to industry-relevant tools. https://www.geeksforgeeks.org/websites-apps/software-and-tools-a-to-z-list/

2.3 **What is Agile Software Development? (YouTube).** A short animated introduction to Agile principles, Scrum roles, sprints, and iterative development. Useful for faculty who want to integrate simple Agile workflows into classroom projects. https://www.youtube.com/watch?v=vYI7-UD9tEQ

[8] https://creativecommons.org/

## 3.3 Artificial Intelligence

3.1 **Why students still need to learn to code in the age of AI.** A blog entry and position paper by the Raspberry Pi Foundation offering clear arguments for coding education, including hands-on learning, computational literacy, and the importance of empowering young people in an AI-driven world. Highly recommended for understanding *why* hands-on programming should remain central to teaching strategies. `https://www.raspberrypi.org/blog/why-kids-still-need-to-learn-to-code-in-the-age-of-ai/`

3.2 **Veritasium: What Everyone Gets Wrong About AI and Learning.** A clear and accessible explanation by Derek Muller (Veritasium) about how humans learn, why effort and confusion matter, and why AI tools do not replace the need for active learning. Highly relevant for understanding why hands-on coding practice remains essential. `https://www.youtube.com/watch?v=0xS68sl2D70`

3.3 **OpenAI Example Prompts for Coding Support.** Demonstrates how students can use AI tools responsibly for debugging, learning concepts, reviewing code, generating test cases, and documenting solutions. `https://platform.openai.com/docs/examples`

CHAPTER 8

# Conclusion & Future Work

This chapter synthesizes the central findings of the thesis and reflects on their theoretical and practical implications. It first summarizes the study's contributions to understanding the alignment between computer science education and industry needs in rural India, and then outlines directions for future research and continued curriculum development.

## 8.1 Conclusion

This thesis set out to explore how computer science education in rural India can be better aligned with industry needs. Through an integrative literature review (Chapter 3), semi-structured interviews with graduates and employers (Section 5.3), observations and a thematic analysis (Section 6.1), it identified persistent mismatches between academic preparation and professional expectations.

The findings related to RQ1 demonstrate that while graduates possess solid theoretical foundations, industry actors primarily expect practical competencies such as problem-solving, teamwork, communication, and the ability to work with real-world codebases (Section 7.1). In addressing RQ2, the analysis revealed concrete skill gaps in applied development practices and professional skills, alongside partial alignments in foundational knowledge and technical understanding (Section 7.2).

Building on these insights, RQ3 was addressed through the development of actionable recommendations for curriculum enhancement (Section 7.3). These recommendations were consolidated into the *Guidance Note for Faculty and Trainers*, which translate empirical findings into practical guidance for teaching methods, curriculum design, and institutional collaboration with industry.

The Guidance Note was evaluated in a focus group with domain experts (Section 7.4), who confirmed their clarity, relevance, and practical feasibility. The feedback from this evaluation was systematically incorporated into the final version, enhancing the usability and transferability of the proposed guidance (Section 7.7).

While this research is limited to a single case study (Section 1.5), it contributes to the growing body of work on aligning educational outcomes with labor market demands in developing contexts. By linking educational research with sustainable development objectives, this thesis provides both conceptual insights and practical tools for improving employability in rural regions.

In conclusion, aligning higher education with real-world industry requirements is not solely a pedagogical task but a developmental imperative. The findings synthesized in Chapter 7 and operationalized through the guidance notes highlight the transformative potential of context-sensitive, industry-informed curricula in empowering students and strengthening local economies.

## 8.2 Future Work

While this study provides empirical insights into the alignment between computer science education and industry needs in rural India, several directions remain open for future research and practical application.

**Extension to multiple colleges as a primary next step.** First and foremost, future research should extend the present single-case design by including multiple colleges across different rural and urban contexts. Investigating additional affiliated and autonomous institutions is a necessary prerequisite for assessing whether the patterns identified in this study reflect broader structural characteristics of computer science education or are specific to the investigated case. Only on this basis can systematic comparisons—such as rural versus urban institutions, different governance models, or regional ecosystems—be meaningfully conducted.

**Longitudinal validation of findings.** Future studies should extend the current qualitative findings through longitudinal research that follows graduates and employers over multiple years. Such investigations could examine whether curriculum reforms and new pedagogical practices, such as project-based or experiential learning, result in measurable improvements in employability outcomes.

**Comparative analyses across regions and institutions.** Since this study focused on one affiliated college in Andhra Pradesh, further research could compare autonomous and affiliated institutions across different Indian states and rural vs urban institutions. This would enable a broader understanding of how governance structures, resource availability, and regional industry ecosystems influence the alignment of curriculum and industry.

**Quantitative extension of qualitative results.** Building on the thematic analysis, future work could incorporate large-scale quantitative surveys among graduates, employers, and faculty to statistically validate the identified skill gaps. Such mixed-methods designs would strengthen the generalizability of the results beyond the present case.

**Curriculum co-design and pilot evaluation.** Further research should develop and test co-design frameworks that bring together faculty, students, and industry mentors to jointly revise syllabi and teaching practices. Pilot implementations at the now autonomous partner institution could be evaluated through pre- and post-assessments of student competence and employability indicators.

**Faculty development and capacity building.** In line with the identified *Faculty Capacity Building* theme and the findings by Spiesberger et al. [14], future work should explore sustainable models for continuous teacher training, including exposure to modern software development practices, agile mentoring, and active learning techniques. Long-term evaluation of such programs would help determine their impact on teaching quality and student outcomes.

**Inclusion of unemployed graduates.** To complement the present study's focus on employed graduates, future research should also incorporate the perspectives of graduates who were unable to secure employment after completing their studies. Engaging with this group would help uncover barriers related to foundational skill deficits, socio-economic constraints, confidence and agency, or structural hiring biases that remain invisible when sampling only successful graduates. Their experiences would significantly deepen the understanding of employability challenges in rural higher education.

**Cross-context ICT4D perspective.** Finally, future work could expand the scope of this study by comparing rural Indian higher education with other low-resource contexts, such as sub-Saharan Africa. Such comparative analyses would continue the research trajectory initiated by Buerstmayr and Spiesberger et al., contributing to a broader understanding of ICT4D-oriented education and curriculum reform in the Global South [14].

# List of Figures

# List of Tables

# Acronyms

**AICTE** All India Council for Technical Education. 49

**AP** Andhra Pradesh. 17, 42, 43, 45–47

**APITA** Andhra Pradesh Information Technology Academy. 31

**B.Tech.** Bachelor of Technology. 47

**CS** Computer Science. 8, 9, 23, 24, 42, 50

**CSE** Computer Science Education. xvi, 27, 37, 38, 40–42, 47, 83, 84, 87

**DRGA** Design-Reality Gap Analysis. 17

**GSDP** Gross State Domestic Product. 42

**HEI** Higher Education Institution. 5, 10, 11, 38, 50, 82, 83

**ICPC** International Collegiate Programming Contest. 20

**ICT** Information and Communications Technology. 3, 32, 42

**ICT4D** Information and Communications Technology for Development. 3, 10, 11, 32, 84

**ILR** Integrative Literature Review. 5, 6, 14, 20–26, 34, 45, 56, 62, 64, 66, 73, 76–78, 81–83

**INR** Indian Rupee. 42

**INSO** Industrial Software. 13, 17, 39

**IT** Information Technology. 42

**JNTUA** Jawaharlal Nehru Technological University Anantapuramu. 39

**MPI** Multidimensional Poverty Index. 1

**NASSCOM** National Association of Software and Services Companies. 19

**NEP 2020** National Education Policy 2020. 42, 69

**NGO** Non-Governmental Organization. 15

**OBE** Outcome-Based Education. 19, 42, 49, 50

**PBL** Project-Based Learning. 42

**PRISMA** Preferred Reporting Items for Systematic Reviews and Meta-Analyses. 23

**RQ** Research Question. 4, 22

**SDG** Sustainable Development Goal. 1, 3, 14

**TIAB** Title and Abstract Screening. 24, 25

**UGC** University Grant Commission. 39

**UN** United Nations. 1, 14

**UNESCO** United Nations Educational, Scientific and Cultural Organization. 10

**XP** Extreme Programming. 18

# References

[1] U. Nations, „2024 Global Multidimensional Poverty Index (MPI)“, United Nations, Tech. Rep., 2024. Accessed: Mar. 12, 2025. [Online]. Available: https://www.undp.org/arab-states/publications/2024-global-multidimensional-poverty-index-mpi.

[5] N. K. Bharti, L. Chancel, T. Piketty, and A. Somanchi, „Income and Wealth Inequality in India, 1922-2023: The Rise of the Billionaire Raj“, World Inequality Lab, Paris, Tech. Rep. 2024/09, 2024. [Online]. Available: https://wid.world/document/income-and-wealth-inequality-in-india-1922-2023-the-rise-of-the-billionaire-raj/.

[6] S. McGrath, „The role of education in development: An educationalist's response to some recent work in development economics“, *Comparative Education*, vol. 46, no. 2, pp. 237–253, 2010. DOI: 10.1080/03050061003775553.

[7] S. Marginson, „The worldwide trend to high participation higher education: Dynamics of social stratification in inclusive systems“, *Higher Education*, vol. 72, pp. 413–434, 2016. DOI: 10.1007/s10734-016-0029-2.

[8] M. G. Lodico, *Methods in educational research : from theory to practice*. Baltimore, MD : Laureate Education, Inc. ; Hoboken, NJ : John Wiley & Son, Inc., 2010, ISBN: 978-0-470-50833-6. [Online]. Available: http://archive.org/details/methodsineducati0000lodi.

[9] UNESCO, *Education for All 2000-2015: Achievements and Challenges*, 2015. [Online]. Available: https://unesdoc.unesco.org/ark:/48223/pf0000232205.

[10] UNICEF, *Education in India*, 2021. [Online]. Available: https://www.unicef.org/india/what-we-do/education.

[11] P. R. Nair, „Increasing employability of indian engineering graduates through experiential learning programs and competitive programming: Case study“, in *Procedia computer science*, vol. 172, Elsevier, 2020, pp. 831–837. DOI: 10.1016/j.procs.2020.05.119.

[13] V. Braun and V. Clarke, „Using thematic analysis in psychology“, *Qualitative Research in Psychology*, vol. 3, no. 2, pp. 77–101, 2006, ISSN: 1478-0887. DOI: 10.1191/1478088706qp063oa. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1191/1478088706qp063oa.

[14] P. Spiesberger, L. Bürstmayr, R. Vallon, and T. Grechenig, „Identifying Higher Software Engineering Education's Design-Reality Gaps in Rural India“, in *2023 46th MIPRO ICT and Electronics Convention (MIPRO)*, 2023, pp. 1572–1577. DOI: 10.23919/MIPRO57284.2023.10159803.

[15] M. Juvane, P. Spiesberger, K. Pinter, R. Vallon, and T. Grechenig, „Assessing Mozambique's Software Industry to Foster Local Universities-Industry Collaboration“, in *2020 IEEE Global Engineering Education Conference (EDUCON)*, 2020, pp. 1740–1747. DOI: 10.1109/EDUCON45650.2020.9125330.

[19] R. J. Torraco, „Writing Integrative Literature Reviews: Using the Past and Present to Explore the Future“, *Human Resource Development Review*, vol. 15, no. 4, pp. 404–428, 2016, ISSN: 1534-4843. DOI: 10.1177/1534484316671606. [Online]. Available: https://doi.org/10.1177/1534484316671606.

[20] R. J. Torraco, „Writing Integrative Literature Reviews: Guidelines and Examples“, *Human Resource Development Review*, vol. 4, no. 3, pp. 356–367, 2005, ISSN: 1534-4843, 1552-6712. DOI: 10.1177/1534484305278283. [Online]. Available: https://journals.sagepub.com/doi/10.1177/1534484305278283.

[22] B. Saunders, „Delivering an undergraduate course to a cross-cultural market using the World Wide Web“, *Proceedings of the International Conference on Information Technology Interfaces, ITI*, pp. 275–280, 2003, ISSN: 13301012. DOI: 10.1109/ITI.2003.1225357.

[23] M. Saunders, P. Lewis, and A. Thornhill, *Research methods for business students*. 2023, ISBN: 978-1-292-40272-7.

[24] T. Wilde and T. Hess, „Methodenspektrum der Wirtschaftsinformatik: Überblick und Portfoliobildung“, Institut für Wirtschaftsinformatik und Neue Medien der Ludwig-Maximilians-Universität München, München, Germany, Arbeitsbericht Nr. 2/2006, 2006. [Online]. Available: http://www.wim.bwl.uni-muenchen.de.

[25] B. B. Kawulich, „Participant Observation as a Data Collection Method“, *Forum: Qualitative Social Research*, vol. 6, no. 2, 2005. DOI: https://doi.org/10.17169/fqs-6.2.466.

[27] D. L. Morgan, *Focus groups as qualitative research*, 2nd ed. Thousand Oaks, CA: SAGE Publications, 1997.

[28] R. Whittemore and K. Knafl, „The integrative review: Updated methodology“, *Journal of Advanced Nursing*, vol. 52, no. 5, pp. 546–553, 2005, ISSN: 0309-2402, 1365-2648. DOI: 10.1111/j.1365-2648.2005.03621.x. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1111/j.1365-2648.2005.03621.x.

[29] R. Vallon, P. Spiesberger, M. Zoffi, C. Zrelski, C. Drager, and T. Grechenig, „Teaching Global Software Engineering in a Remote Customer Environment“, in *2018 IEEE 10th International Conference on Engineering Education (ICEED)*, Kuala Lumpur, Malaysia: IEEE, 2018, pp. 63–68, ISBN: 978-1-5386-6847-4. DOI: 10.1109/ICEED.2018.8626985. [Online]. Available: https://ieeexplore.ieee.org/document/8626985/.

[30] L. Buerstmayr, „A Proposal to Strengthen Higher Software Engineering Education in Rural India“, M.S. thesis, TU Wien, Vienna, 2022.

[31] J. M. Bass and R. Heeks, „Changing Computing Curricula in African Universities: Evaluating Progress and Challenges via Design-Reality Gap Analysis“, *The Electronic Journal of Information Systems in Developing Countries*, vol. 48, no. 1, pp. 1–39, 2011, ISSN: 1681-4835. DOI: 10.1002/J.1681-4835.2011.TB00341.X.

[32] G. B. Mustafa, K. G, V. Vijay C, M. S, S. S, and A. Paranthaman, „Bridging the Skill Gap Between Academia and Industry in Punjab: A Case Study“, in *Proceedings of the 12th International Conference on Education Technology and Computers*, New York, NY, USA: Association for Computing Machinery, 2021, pp. 239–245, ISBN: 978-1-4503-8827-6. DOI: 10.1145/3436756.3437052. [Online]. Available: https://doi.org/10.1145/3436756.3437052.

[33] H. Snyder, „Literature review as a research methodology: An overview and guidelines“, *Journal of Business Research*, vol. 104, pp. 333–339, 2019, ISSN: 01482963. DOI: 10.1016/j.jbusres.2019.07.039.

[34] D. Moher, A. Liberati, J. Tetzlaff, D. G. Altman, and T. P. Group, „Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement“, *PLoS Medicine*, vol. 6, no. 7, e1000097, 2009.

[35] V. Garousi, G. Giray, E. Tuzun, C. Catal, and M. Felderer, „Aligning software engineering education with industrial needs: A meta-analysis“, *Journal of Systems and Software*, vol. 156, pp. 65–83, 2019, ISSN: 0164-1212. DOI: 10.1016/j.jss.2019.06.044.

[36] V. Garousi, G. Giray, E. Tuzun, C. Catal, and M. Felderer, „Closing the gap between software engineering education and industrial needs“, *IEEE Software*, vol. 37, no. 2, pp. 68–77, 2020, ISSN: 0740-7459. DOI: 10.1109/MS.2018.2880823.

[37] V. Garousi, G. Giray, and E. Tuzun, „Understanding the knowledge gaps of software engineers: An empirical analysis based on SWEBOK“, *ACM Transactions on Computing Education*, vol. 20, no. 1, 2020, ISSN: 1946-6226. DOI: 10.1145/3360497.

[38] A. Radermacher, G. Walia, and D. Knudson, „Investigating the skill gap between graduating students and industry expectations“, in *36th international conference on software engineering (icse companion 2014)*, P. Jalote, L. Briand, and A. VanDer-Hoek, Eds., New York, NY, USA: ACM, 2014, pp. 291–300, ISBN: 978-1-4503-2768-8. DOI: 10.1145/2591062.2591159.

[39] F. Dobslaw, K. Angelin, L.-M. Oberg, and A. Ahmad, „The gap between higher education and the software industry - a case study on technology differences“, in *Proceedings of the 5th european conference on software engineering education, ecsee 2023*, New York, NY, USA: ACM, 2023, pp. 11–21, ISBN: 978-1-4503-9956-2. DOI: 10.1145/3593663.3593690.

[40] C.-P. R., C.-L. C., S.-A. P., G.-P. F.J., and T.-C. E., „Competence gaps in software personnel: A multi-organizational study“, *Computers in Human Behavior 29(2)*, 2013. DOI: https://doi.org/10.1016/j.chb.2012.04.021.

[41] R. Pham, S. Kiesling, L. Singer, and K. Schneider, „Onboarding inexperienced developers: Struggles and perceptions regarding automated testing“, *Software Quality Journal*, vol. 25, no. 4, SI, pp. 1239–1268, 2017, ISSN: 0963-9314. DOI: 10.1007/s11219-016-9333-7.

[42] W. Groeneveld, J. Vennekens, and K. Aerts, „Identifying non-technical skill gaps in software engineering education: What experts expect but students don't learn“, *ACM Transactions on Computing Education*, vol. 22, no. 1, 2022, ISSN: 1946-6226. DOI: 10.1145/3464431.

[43] A. Mishra and D. Mishra, „Industry linked graduate software engineering curriculum“, *SIGSOFT Softw. Eng. Notes*, vol. 36, no. 6, pp. 1–4, 2011, ISSN: 0163-5948. DOI: 10.1145/2047414.2047429.

[44] K. Falkner and N. J. G. Falkner, „Integrating communication skills into the computer science curriculum“, in *Sigcse 12: Proceedings of the 43rd acm technical symposium on computer science education*, New York, NY, USA: ACM, 2011, pp. 379–384, ISBN: 978-1-4503-1098-7.

[45] M. Ntinda, M. Apiola, and E. Sutinen, „Mind the gap: Aligning software engineering education and industry in namibia“, in *2021 ist-africa conference (ist-africa)*, New York, NY, USA: IEEE / IST Africa, 2021, ISBN: 978-1-905824-67-0.

[46] V. Bevanda, D. Brenko, and P. Frankovic, „University-industry collaboration: A case study of enhancing the quality of information technology education“, in *Inted2014: 8th international technology, education and development conference*, L. Chova, A. Martinez, and I. Torres, Eds., International Academy of Technology, Education and Development (IATED), 2014, pp. 252–261, ISBN: 978-84-616-8412-0.

[47] H. Hamrawi, „Design and implementation of computer science curriculum in extreme conditions: First results“, in *World sustainable development outlook 2014: West meet east: Sharing the past and current experience to benefit the future*, A. Ahmed, Ed., University of Sussex, England: World Association for Sustainable Development (WASD), 2014, pp. 469–481, ISBN: 978-1-907106-31-6.

[48] J. Plane and I. Venter, „Comparing capacity building frameworks for computer science education in underdeveloped countries: An african and asian perspective“, in *Iticse '08: Proceedings of the 13th annual conference on innovation and technology in computer science education*, New York, NY, USA: ACM, 2008, pp. 306–310, ISBN: 978-1-60558-078-4.

[49] A. Zeid, „Lessons learned from establishing a software engineering academic programme in developing countries“, in *20th conference on software engineering education & training (CSEET'07)*, 2007, pp. 11–18. DOI: 10.1109/CSEET.2007.34.

[50] G. M. Lundberg, B. R. Krogstie, and J. Krogstie, „Becoming fully operational: Employability and the need for training of computer science graduates“, in *Proceedings of the 2020 ieee global engineering education conference (educon 2020)*, A. Cardoso, G. Alves, and M. Restivo, Eds., New York, NY, USA: IEEE, 2020, pp. 644–651, ISBN: 978-1-7281-0930-5. DOI: 10.1109/educon45650.2020.9125188.

[51] W. A. Lawrence-Fowler, L. M. Grabowski, and C. F. Reilly, „Bridging the divide: Strategies for college to career readiness in computer science“, in *Frontiers in education conference (fie), 2015*, New York, NY, USA: IEEE, 2015, pp. 1792–1799, ISBN: 978-1-4799-8454-1.

[52] G. M. Lundberg, „To ease the transition to employment: Employability in higher IT education“, in *19th koli calling conference on computing education research (koli calling 2019)*, New York, NY, USA: ACM, 2019, ISBN: 978-1-4503-7715-7. DOI: 10.1145/3364510.3366167.

[53] S. J.-B.R.G., „Engineering employability skills: Students, academics, and industry perceptions“, *Int. J. of Mechanical Engineering Education*, 2025. DOI: 10.1177/03064190231214178.

[54] M. H. Nsenge, M. L. Waruguru, and M. S. Nyaga, „Exploratory factor analysis of congolese information technology graduates' employability: Towards sustainable employment“, *SAGE OPEN*, vol. 13, no. 4, 2023, ISSN: 2158-2440. DOI: 10.1177/21582440231210109.

[55] C. M. and Z. M., „A Bourdieusian approach to understanding employability“, *J. of Vocational Education & Training*, 2013. DOI: 10.1080/13636820.2013.782887.

[56] C. Maguire and L. Matthews-Dibbins, „Building graduate employability through professional certifications“, in *12th international technology, education and development conference (inted)*, L. Chova, A. Martinez, and I. Torres, Eds., Valencia, Spain: IATED-INT, 2018, pp. 6730–6735, ISBN: 978-84-697-9480-7.

[57] J. J. Siddoo V.; Sawattawee, „Exploring the competency gap of IT students in Thailand: The employers' view of an effective workforce“, *Journal of Technical Education and Training*, 2017. Accessed: Mar. 12, 2025. [Online]. Available: https://publisher.uthm.edu.my/ojs/index.php/JTET/article/view/1868.

[58] U. Sahin Y.G.; Celikkan, „Information technology asymmetry and gaps between HEIs and industry“, *Journal of Information Technology Education: Research*, 2020. DOI: 10.28945/4553.

[59] W. Groeneveld, B. A. Becker, and J. Vennekens, „Soft skills: What do computing program syllabi reveal about non-technical expectations of undergraduate students?“, in *Proceedings of the 2020 acm conference on innovation and technology in computer science education, iticse 2020*, New York, NY, USA: ACM, 2020, pp. 287–293, ISBN: 978-1-4503-6874-2. DOI: 10.1145/3341525.3387396.

[60] M. A. Mardis et al., „Assessing alignment between information technology educational opportunities, professional requirements, and industry demands“, *Education and Information Technologies*, vol. 23, no. 4, pp. 1547–1584, 2018, ISSN: 1360-2357. DOI: 10.1007/s10639-017-9678-y.

[61] S. Jayathissa et al., „Industry mentoring program: Empowering external degree students for career success: Case study from university of moratuwa, sri lanka“, in *Proceedings of the 23rd european conference on e-learning, ecel 2024*, F. Moriera, Ed., vol. 23/1, Curtis Farm, Kidmore End, RG4 9AY: ACAD Conferences Ltd, 2024, pp. 138–146, ISBN: 978-1-917204-21-7 978-1-917204-22-4.

[62] S. Balakrishna, „Is information technology (IT) education industry-relevant and gender-inclusive? Perspectives from andhra pradesh, india“, in *Proceedings of the 12th international conference on theory and practice of electronic governance (icegov2019)*, S. BenDhaou, L. Carter, and M. Gregory, Eds., ACM, 2019, pp. 497–499, ISBN: 978-1-4503-6644-1. DOI: 10.1145/3326365.3326439.

[63] P. Singh and L. K. Singh, „Reliability and safety engineering for safety-critical systems in computer science: A study into the mismatch between higher education and employment in brazil and india“, *IEEE Transactions on education*, vol. 64, no. 4, pp. 353–360, 2021, ISSN: 0018-9359. DOI: 10.1109/TE.2021.3057611.

[64] L. Dirckinck-Holmfeld, A. Bygholm, and G. O. Tabo, „Transforming education through ICT: Exploring students' study practices in a resource-constrained university setting“, *British Journal of Educational Technology*, vol. 54, no. 6, pp. 1463–1483, 2023, ISSN: 0007-1013. DOI: 10.1111/bjet.13367.

[65] E. Sutinen and M. Tedre, „ICT4D: A computer science perspective“, in *Algorithms and applications: Essays dedicated to esko ukkonen on the occasion of his 60th birthday*, T. Elomaa, H. Mannila, and P. Orponen, Eds., vol. 6060, Berlin, Germany: SPRINGER-VERLAG BERLIN, 2010, pp. 221–231, ISBN: 978-3-642-12475-4.

[66] I. M. of Education, *All indian survey on higher education instruction manual*, 2020. Accessed: Mar. 12, 2025. [Online]. Available: https://www.education.gov.in/sites/upload_files/mhrd/files/statistics-new/aishe_eng.pdf.

[67] U. G. Commission, „Annual report 2022-23“, University Grants Commission, Ministry of Education, Government of India, New Delhi, India, Tech. Rep., 2023. [Online]. Available: https://www.ugc.ac.in.

[68] Government of India, Ministry of Education, *National education policy 2020*. New Delhi, 2020. Accessed: Mar. 12, 2025. [Online]. Available: https://www.education.gov.in/sites/upload_files/mhrd/files/NEP_Final_English_0.pdf.

[69] P. J. Denning, „The profession of IT: The great principles of computing“, *Communications of the ACM*, vol. 52, no. 4, pp. 28–30, 2009. DOI: 10.1145/1498765.1498789.

[70] ACM and I. C. Society, *Computing curricula 2020: Paradigms for global computing education*, 2020. [Online]. Available: https://www.acm.org/binaries/content/assets/education/curricula-recommendations/cc2020.pdf.

[71] UNESCO, *ICT competency framework for teachers, version 3*. Paris: United Nations Educational, Scientific and Cultural Organization, 2021, ISBN: 978-92-3-100424-3. [Online]. Available: https://unesdoc.unesco.org/ark:/48223/pf0000371027.

[72] Aspiring Minds, „National employability report engineers 2019“, Aspiring Minds Assessment Pvt. Ltd., Tech. Rep., 2019. [Online]. Available: https://www.aspiringminds.com/research-reports/national-employability-report-engineers-2019/.

[73] World Bank, *India development update 2024: Toward inclusive growth*, 2024. Accessed: Oct. 21, 2024. [Online]. Available: https://www.worldbank.org/en/country/india/publication.

[74] NASSCOM, „Strategic review 2024: The technology sector in india“, New Delhi, Tech. Rep., 2024. [Online]. Available: https://nasscom.in.

[75] Government of Andhra Pradesh, Finance Department, *Socio-economic survey 2024–25*, 2025. Accessed: Mar. 12, 2025. [Online]. Available: https://apfinance.gov.in/socio.html.

[79] Software Technology Parks of India (STPI), *Annual performance report 2024*, 2024. Accessed: Mar. 12, 2025. [Online]. Available: https://stpi.in/en/node/5243.

[80] Government of Telangana, ITE&C Department, „IT/ITeS sector performance 2023–24 Annual Report“, Hyderabad, Tech. Rep., 2024. Accessed: Mar. 12, 2025. [Online]. Available: https://it.telangana.gov.in.

[81] Karnataka Digital Economy Mission, „Bengaluru innovation report 2024“, Bengaluru, Tech. Rep., 2024. Accessed: Mar. 12, 2025. [Online]. Available: https://karnatakadigital.in.

[82] R. Rajput, „Reforms in higher education in india: A study on NEP 2020“, *International Journal of Multidisciplinary Research*, vol. 9, no. 6, pp. 45–52, 2022.

[83] R. G. Burgess, *In the field: An introduction to field research*. London and New York: Routledge, 1984, ISBN: 978-0-415-07867-9.

[84] B. S. Bloom, M. D. Engelhart, E. J. Furst, W. H. Hill, and D. R. Krathwohl, „Handbook I: Cognitive domain“, *New York: David McKay*, pp. 483–498, 1956.

[86] J. Hufnagl, „Pragmatic Teaching Methods Raising the Quality of Higher Computer Science Education in Developing Countries by a Root Cause Analysis exemplified by Rural India“, Bachelor Thesis, TU Wien, Vienna, 2024.

[87] M. J. Healey and M. B. Rawlinson, „Interviewing business owners and managers: A review of methods and techniques“, *Geoforum; journal of physical, human, and regional geosciences*, vol. 24, no. 3, pp. 339–355, 1993, ISSN: 0016-7185. DOI: 10.1016/0016-7185(93)90026-E.

[88] J. Fereday and E. Muir-Cochrane, „Demonstrating rigor using thematic analysis: A hybrid approach of inductive and deductive coding and theme development“, *International Journal of Qualitative Methods*, vol. 5, no. 1, pp. 80–92, 2006.

# Web References


[2] *Worldbank - understanding poverty*. Accessed: Apr. 18, 2024. [Online]. Available: https://www.worldbank.org/en/understanding-poverty.

[3] *Home - United Nations Sustainable Development*. Accessed: Mar. 12, 2025. [Online]. Available: https://sdgs.un.org/.

[4] Statista, *Gross Domestic Product (GDP) in India*. Accessed: Mar. 12, 2025. [Online]. Available: https://www.statista.com/statistics/263771/gross-domestic-product-gdp-in-india/.

[16] ICT4D.at, *About ICT4D.at*. Accessed: Mar. 12, 2025. [Online]. Available: https://www.ict4d.at/about.

[17] *Goal 4 | Department of Economic and Social Affairs*. Accessed: Oct. 20, 2024. [Online]. Available: https://sdgs.un.org/goals/goal4#targets_and_indicators.

[18] *Goal 8 | Department of Economic and Social Affairs*. Accessed: Oct. 20, 2024. [Online]. Available: https://sdgs.un.org/goals/goal8#targets_and_indicators.

[76] Exchange-Rates.org, *Indian rupee (INR) to euro (EUR) exchange rate history*. Accessed: Mar. 12, 2025. [Online]. Available: https://www.exchange-rates.org/exchange-rate-history/inr-eur-2024.

[77] Statista Research Department, *Gross domestic product (GDP) in austria from 2013 to 2023*. Accessed: Mar. 12, 2025. [Online]. Available: https://www.statista.com/statistics/526181/gross-domestic-product-gdp-in-austria.

[78] Eurostat, *Gross domestic product (GDP) at market prices, EU-27*. Accessed: Mar. 12, 2025. [Online]. Available: https://ec.europa.eu/eurostat/databrowser/view/tec00001/default/table.

[85] C. Drew, *Bloom's taxonomy: Levels of understanding*, 2020. Accessed: Mar. 12, 2025. [Online]. Available: https://helpfulprofessor.com/levels-of-understanding/.

APPENDIX A

# Interview Guides

## Introduction (Warm-Up)

- Brief introduction of yourself and the purpose of the study.
- Obtain informed consent for recording and voluntary participation. Rights of the participating person, go through the document and underline the important parts:
  - anonym
  - right to step back
  - nobody will have access to recordings
  - free will
  - contact any time
- Explain the purpose of the interview: to explore the alignment of computer science education with industry needs. **Make clear that we discuss CSE graduates from rural engineering colleges who work**.

## Participant Background - 00:10

- Can you describe your educational background and your current job role?
- How did you get your job/role?
- How long have you been working in the software industry?

## Training and Development - 00:20

- How was your hiring process?
- How good were you prepared for the hiring process?
- What training did your company provide?
- Could you share specific examples?
- What would you have liked to learn during your university studies that could have reduced the need for additional training?

## Core Skills Discussion - 00:32

- What hard skills from your education have been **most** useful in your current job?
- What hard skills from your education have been **not so** useful in your current job?
- Which soft skills were taught during your studies?
- How relevant have these been at work?
- Were there areas where your education **fell short** in preparing you for real-world job requirements?
- Were there areas where your education preparing you **well** for real-world job requirements?

## Educational Preparedness - 00:42

- Do you think the curriculum provided enough hands-on experience?
- Could you provide examples?
- In what specific ways did your education help you in your career?
- Where do you feel improvements could have been made?

## Curriculum Improvement and Recommendations - 00:47

- If you could redesign the curriculum at your college, what changes would you make to better prepare students for the tech industry?
- What advice would you give to current students to better prepare themselves for industry expectations?

**Mental Break – Did I get all answer to my "What do I want to know"?**

## Closing and Open Discussion - 00:55

- Is there anything else you would like to add that we haven't covered?
- Any additional thoughts on how the education system can better align with the needs of the tech industry?

## Introduction (Warm-Up)

- Brief introduction of yourself and the purpose of the study.
- Obtain informed consent for recording and voluntary participation. Rights of the participating person, go through the document and underline the important parts:
  - anonym
  - right to step back
  - nobody will have access to recordings
  - free will
  - contact any time
- Explain the purpose of the interview: to explore the alignment of computer science education with industry needs. **Make clear that we discuss CSE graduates from rural engineering colleges**.

## Participant Background – 00:10

- Please introduce your company and its role in the tech industry?
- What is your role in the company?
- What interactions does your company have with computer science graduates from rural areas?

## Hiring and Training – 00:20

- How many rural and how many urban graduates do you hire?
- How do you select graduates to hire?
- Any difference between rural and urban graduates?

- How does the training of new graduates work in your company?
- How much time and resources does your company invest in training?
- What areas require the **most** training?
- What areas require **no** training?

## Core Skills Discussion – 00:32

- What skills do you expect from graduates?
- What technical skills do you expect from graduates?
- Are there any soft skills or behaviors **lacking** among graduates from rural colleges?
- Are there any soft skills or behaviors **good** among graduates from rural colleges?

## Educational Preparedness – 00:40

- To what extent do you feel that the education provided by rural colleges aligns with the **technical requirements** of your company?
- To what extent do you feel that the education provided by rural colleges aligns with the **soft skill requirements** of your company?
- Any recurring skills missing with graduates entering your workforce?
- Can you identify any recurring skills where graduates are good at while entering your workforce?
- Do you see a difference between graduates from rural colleges compared to urban colleges?

## Curriculum Improvement and Recommendations – 00:50

- How could colleges better prepare students for industry requirements?

**Mental Break – Did I get all answer to my "What do I want to know"?**

## Closing and Open Discussion – 00:55

- Is there anything else you would like to add that we haven't covered?
- Any additional thoughts on how the education system can better align with the needs of the tech industry?
- What will the future look like? (Asking because of Hackathons etc)

## Pre-Interview Material

- The expert will receive the complete set of Guidance Notes at least three days before the interview.
- The material is accompanied by a one-page summary describing:
  - the purpose of the evaluation,
  - the three key dimensions (clarity, feasibility, transferability),
  - and the expected duration of the interview.
- The participant is invited to make notes and prepare comments prior to the interview.

## Introduction (Warm-Up)

- Briefly introduce yourself and the purpose of this additional interview.
- Explain that this conversation focuses on evaluating the **Guidance Notes** developed in this thesis, which summarize practical recommendations for improving the alignment of computer science education and industry needs in rural India.
- Obtain informed consent for recording and voluntary participation. Review participant rights and underline key aspects:
  - anonymized data use
  - right to withdraw at any time
  - recordings are not accessible to others
  - voluntary participation
  - ability to contact the researcher anytime for clarifications
- Clarify that this is a reflective discussion about the practicality, clarity, and transferability of the developed Guidance Notes, not an evaluation of the participant's work or institution.

## Participant Background – 00:05

- Could you briefly describe your background and experience in higher education, curriculum design, or ICT4D projects?
- What is your current role, and how does it relate to curriculum management or educational reform?
- How familiar are you with challenges faced by rural higher education institutions, particularly in computer science or engineering education?

## Evaluation Dimension 1: Clarity and Comprehensibility – 00:10

- How clear and comprehensible do you find the structure and wording of the Guidance Notes?
- Are the recommendations formulated in a way that practitioners (e.g., faculty or administrators) could easily understand and apply them?
- Do you think additional examples, visuals, or contextual explanations would improve clarity?
- Were there any sections you found confusing, ambiguous, or too general?

## Evaluation Dimension 2: Feasibility and Applicability – 00:25

- From your experience, how realistic are the proposed recommendations for implementation in rural higher education contexts?
- Which of the recommendations seem most feasible to implement with existing institutional structures and resources?
- Which areas might face the strongest resistance or resource constraints?
- What would be necessary conditions for these recommendations to be successfully realized?

## Evaluation Dimension 3: Scalability and Transferability – 00:40

- In your opinion, could the Guidance Notes be adapted for other affiliated colleges or comparable institutions?

- Are there elements that should be modified or localized for different regional or institutional settings?
- What factors (e.g., governance, language, infrastructure) might affect their transferability?
- How could these recommendations be disseminated effectively to ensure uptake by other colleges or stakeholders?

## General Reflections and Additional Insights – 00:50

- Do you think the Guidance Notes adequately address the major issues identified in the study (e.g., skill mismatch, lack of practical training, institutional barriers)?
- Are there additional aspects or recommendations that you think should be included?
- How do you assess the potential of these Guidance Notes to support curriculum reform and employability enhancement in rural India?

**Mental Break – Did I get all answers to my "What do I want to know"?**

## Closing and Open Discussion – 00:55

- Is there anything else you would like to add or elaborate on?
- Do you have any advice on how the Guidance Notes could be improved or further evaluated (e.g., piloting, policy integration)?
- Thank the participant for their time and valuable feedback.

APPENDIX B

# Briefing Document for Expert Interview

## Evaluation of the Guidance Notes – Aligning Educational Outcomes with Industry Needs in Rural India

**Researcher:** Johannes Hufnagl, MSc Candidate, TU Wien
**Date:** November 2025

### Purpose of the Evaluation

The purpose of this evaluation is to gather expert feedback on the *Guidance Notes* developed as part of the diploma thesis *"Aligning Educational Outcomes with Industry Needs: A Study on Computer Science Education Exemplified by a College in Rural India."*

The Guidance Notes synthesize empirical findings from interviews, observations, and literature into actionable recommendations for improving the alignment between computer science education and industry requirements in rural Indian higher education institutions.

This expert interview aims to validate the **practical relevance**, **clarity**, and **feasibility** of these recommendations. The expert's insights will help assess whether the proposed measures are realistic, contextually appropriate, and transferable to other affiliated colleges.

### Evaluation Framework

During the interview, the discussion will focus on three key dimensions:

1. **Clarity and Comprehensibility**
   Are the recommendations clearly structured and easy to understand for their

intended audience (faculty, administrators, policymakers)? Would any sections benefit from simplification, examples, or visual support?

2. **Feasibility and Applicability**
   Are the recommendations realistically implementable under current institutional and infrastructural conditions of rural higher education? Which measures appear most and least achievable, and why?

3. **Scalability and Transferability**
   Could the Guidance Notes be adapted for other affiliated colleges or similar educational contexts? Which factors (e.g., language, governance, partnerships) influence their adaptability?

### Interview Format and Duration

- **Format:** Semi-structured interview
- **Estimated duration:** 45–60 minutes
- **Notes taking & Scribbling:** ...
- **Confidentiality:** All data will be anonymized and used solely for academic purposes
- **Preparation:** The expert receives the complete set of Guidance Notes prior to the interview to allow for review and note-taking

**Thank you for your valuable contribution.**
Your reflections will help ensure that the Guidance Notes are both practical and impactful for improving educational quality and employability in rural India.

APPENDIX C

# Integrative Literature Review

| Major Term | Database | Core Query (simplified) | Hits (raw) | Hits (after deduplication) | Export Format |
|---|---|---|---|---|---|
| Skill Gaps | Scopus | ("Skill gap" OR "Competency gap" OR "Knowledge gap" OR "Skills mismatch") AND ("Computer Science" OR "Software Engineering") | 275 | 198 | Excel |
| Skill Gaps | Web of Science | same core string | 231 | 162 | XLSX |
| Skill Gaps | IEEE Xplore | same core string | 190 | 144 | CSV |
| Skill Gaps | ACM DL | same core string | 120 | 97 | BibTeX |
| Curriculum Development | Scopus | ("Curriculum development" OR "Curriculum alignment" OR "Education reform") AND ("Higher Education" OR "Universities") | 342 | 216 | Excel |
| Curriculum Development | Web of Science | same core string | 298 | 203 | XLSX |
| Curriculum Development | IEEE Xplore | same core string | 220 | 174 | CSV |
| Curriculum Development | ACM DL | same core string | 165 | 112 | BibTeX |
| Employability | Scopus | ("Employability" OR "Job readiness" OR "Workplace skills") AND ("Computer Science graduates" OR "University graduates") | 411 | 307 | Excel |

*(continued on next page)*

*(continued from previous page)*

| **Major Term** | **Database** | **Core Query (simplified)** | **Hits (raw)** | **Hits (after deduplication)** | **Export Format** |
|---|---|---|---|---|---|
| Employability | Web of Science | same core string | 355 | 281 | XLSX |
| Employability | IEEE Xplore | same core string | 260 | 199 | CSV |
| Employability | ACM DL | same core string | 176 | 132 | BibTeX |
| Developing Countries | Scopus | ("Developing countries" OR "Rural India" OR "Emerging economies") AND ("Higher education" OR "Educational inequalities") | 192 | 133 | Excel |
| Developing Countries | Web of Science | same core string | 156 | 118 | XLSX |
| Developing Countries | IEEE Xplore | same core string | 97 | 82 | CSV |
| Developing Countries | ACM DL | same core string | 84 | 69 | BibTeX |
| Industry Expectations | Scopus | ("Industry expectations" OR "Employer expectations" OR "Industry requirements" OR "Employer needs" OR "Skills demand") AND ("IT sector" OR "Technology companies") | 320 | 231 | Excel |
| Industry Expectations | Web of Science | same core string | 393 | 287 | XLSX |

*(continued on next page)*

*(continued from previous page)*

| **Major Term** | **Database** | **Core Query (simplified)** | **Hits (raw)** | **Hits (after deduplication)** | **Export Format** |
|---|---|---|---|---|---|
| Industry Expectations | IEEE Xplore | same core string | 208 | 176 | CSV |
| Industry Expectations | ACM DL | same core string | 143 | 109 | BibTeX |

Table C.1: Search results before and after deduplication across databases for each major term.

APPENDIX D

# Thematic Coding

| Code | Description (interpretive summary) | Source(s) | Occ. |
|---|---|---|---|
| Hiring process | Hiring processes combine formal tests and informal evaluations, emphasis on technical and communication skills over academic grades. | Employer Interviews | 10 |
| Aptitude Test | Written or online aptitude tests used as initial filters to evaluate problem-solving and analytical ability. | Employer Interviews | 5 |
| Soft skills assessment in hiring | Teamwork, motivation, and communication frequently assessed via interviews or group discussions. | Employer Interviews | 8 |
| Hard skills assessment in hiring | Coding and technical assessments serve as primary selection criteria for software roles. | Employer Interviews | 12 |
| Hands-on hiring process with real coding tasks | Preference for task-based assessments reflecting real project work. | Employer Interviews | 6 |
| Worklife | Insights into workplace expectations, culture, and adaptation challenges faced by new graduates. | Graduate Interviews | 4 |
| Training on the job | Continuous learning and onboarding practices, learning-by-doing viewed as essential for bridging theory–practice gaps. | Employer + Graduate Interviews | 11 |
| Technology used in job | Programming languages, frameworks, and tools prevalent in industry (e.g., Python, cloud platforms). | Employer Interviews | 7 |
| Use of AI tools | Application of AI-assisted coding and productivity tools in daily work routines. | Employer Interviews | 4 |
| Product/Service based companies | Distinction between product- and service-oriented IT companies affecting job roles and learning scope. | Employer Interviews | 1 |
| Educational Preparedness | Graduates' perception of their readiness for real-world tasks, often limited due to outdated curricula. | Graduate + Employer Interviews | 8 |
| Learning | Reflections on learning strategies, combining formal instruction with self-motivated exploration. | Graduate Interviews | 2 |
| Self-learning through online platforms | Students rely on MOOCs, YouTube, and online tutorials to acquire up-to-date technical skills. | Graduate Interviews | 9 |

*Continued on next page*

| Code | Description (interpretive summary) | Source(s) | Occ. |
|---|---|---|---|
| Online certifications to supplement knowledge | Use of online certifications to enhance employability and signal skills to recruiters. | Graduate Interviews | 3 |
| RIT Experience | RIT modules offered valuable exposure to applied learning and practical projects. | Graduate + Observation | 4 |
| Teaching | Variation in lecturer quality and pedagogy, preference for interactive and project-based instruction. | Graduate + Observation | 7 |
| Exams | Exams perceived as theoretical and misaligned with practical skill needs. | Graduate Interviews | 2 |
| Hardskill Training | Institution-led training focused on programming or networking skills. | College Observation | 2 |
| Softskill Training | Occasional workshops for communication or teamwork, but limited institutional focus. | College Observation | 1 |
| College Experience | Broader student experiences shaping learning motivation and peer collaboration. | Graduate Interviews | 31 |
| Rural/Urban differences | Perceived disadvantages for rural students due to limited access to infrastructure, exposure, and networks. | Graduate Interviews | 4 |
| Gender differences | Female graduates face subtle barriers and lower participation in extracurricular learning. | Graduate Interviews | 1 |
| Placement in college | College placement cells offer limited outreach, reliance on self-initiated applications is common. | Graduate Interviews | 5 |
| Suggestions | Participants' reflective comments on how curricula and industry collaboration could be improved. | Graduate + Employer Interviews | 3 |
| Tips for students | Advice for future cohorts: focus on self-learning, internships, and communication skills. | Graduate Interviews | 4 |
| College Improvement suggestions | Recommendations for institutional reforms (modern labs, qualified teachers, project-based courses). | Graduate + Employer Interviews | 5 |
| Collaboration with Academia | Industry experts recommend structured collaborations such as workshops, mentorships, and joint projects. | Employer + Academic Experts | 14 |

Table D.1: Refined Codebook from MAXQDA (Interviews and Observations)

# APPENDIX E

# Interview Transcript Example

1 [0:00:00.0] **Graduate Interview G3**

2 **Transcript**

3 **S1:** Thank you for for taking your time and meeting with me.

4 [0:00:05.1] **S2:** I'm glad I can be of help for you.

5 [0:00:07.3] **S1:** You're very, very helpful. And, Yeah, I'm really, honored that you also show me around your office and and everything. Yes. So maybe can you describe your educational background a little bit and.

Hiring process

6 [0:00:24.1] **S2:** Yeah. So, yeah. I had my usual, schooling and high schooling, in a central board. And then I came to [partner college] for my bachelor's of technology in computer science. And after that, I mean, after I was also a part of [course] as well. So after that, I got an offer from, [Company name] India, for being a scholar, in [Company name]. So the scholar program is basically, I mean, a combination of both, work as well as an MTech, degree being given from Bits-pilani, which is one another prestigious institution in India. So, yeah. So, I was simultaneously doing my M.Tech as well as my work, for the past two years in [Company name], and now it's finally done. So I'm in my M.Tech is done. So yeah, now I'm a full time employee in the [Company name] .

..College Experience

7 [0:01:29.3] **S1:** Nice. Congratulations also to you. So this, college or university you did the M.Tech. Is, like remote, master's right because it's in Rajasthan or somewhere. [0:01:44.4]

..Training on the job

8 **S2:** Yes. Okay. So, it's a remote. It is, called work integrated learning program. So it basically, I mean, the some, some companies, sponsor that for the students and some have to, I mean, pay for the courses, but, [Company name] has sponsored the entire course for all the scholars, so. Yeah. So that is how I got this M.Tech. And that is basically I mean, it is remote. I mean, the people are I mean, the college is there in the Rajasthan, but during Saturdays, we usually have classes in the office itself. So there are some, I mean, in the meeting rooms or in, there are some other, classroom kind of, places here in the campus itself. So visiting lecturers or visiting professors, we can call them as, what do we call their, I'm not sure. I mean, they, I mean, part time lecturers or something, so. Yeah, they are again, from the I mean, they are part of the bits It's college and they usually come and teach us during the weekends. Yeah. So every Saturday we had classes. Nice. So yeah, exams were not remote. Exams were conducted in the office itself.

9 [0:03:09.7] **S1:** Okay, okay. So. Yeah. Yeah. That is how nice. Sounds nice. Also, that, your company is paying for that. Yeah. Also really nice. Yeah. So. And what's your current job role? Can you describe what you are? Yes.

Worklife

10 [0:03:25.8] **S2:** I currently work as an associate developer. Mainly doing the back end work in, using Java for a product called [Company name] product from [Company name]. Okay. Yes. So, my work is basically, I mean, it covers all the back end interactions. And as well as, I mean, there are different kinds of modules in that project. So I usually work on the task and visits module. So [Company product name] is like a project which is used by the sales group sales groups of different companies. So any sales persons or sales managers and all. So they are planning the visits to I mean for their sales and all if they are planning any. I mean if they have any tasks to accomplish for their sales team and all. So all those kind of management is being, provided by [Company product name] or [Company product name]. So the work I do is basically the back end work for that, project.

11 [0:04:38.5] **S1:** Nice sounds great. How did you get your your job? Your in.

12 [0:04:45.2] **S2:** Yeah. So usually in India, it's a custom kind of thing that, companies usually come to colleges.

13 [0:04:57.0] **S1:** I have to write it down a little bit because I actually have some notes from other colleagues say I interviewed. And they also I mean, it's to say, not so much different always. Yeah. So you thank you. There is the companies come to college, right. And they, do some placements or how does it work?

Hiring process

14 [0:05:25.8] **S2:** So, jobs in India is like, two different kind of phases. One is for freshers, one is for experienced people. So freshers one is basically freshers are known, the people who just graduated from the college and they are looking for jobs. So usually for freshers it is the entire hiring thing in the Companies is taken care by the placement. Is a thing called placements, where college has a tie up with the companies, and they invite the companies to their colleges and, test the skills of their students and then take a take people from them. So that is how placements work. So this is for freshers. And there is another way of getting into companies in India that is through your experience. So that is again using LinkedIn or any job online openings or postings, job postings. So based on that we can apply based on our skills we can apply and then even the freshers. Also, there is a chance where we can do the same using LinkedIn or something. But again, placements coming to

Hiring process

the college is the easiest way where students usually find the Jobs.

15 [0:06:51.0] **S1:** So also the most common way for a student to find a job placement.

16 [0:06:54.3] **S2:** Yes. I mean, the, I mean, common, most common way to get into company is through placements. Yeah.

17 [0:07:02.0] **S1:** It's always freshers, always BTech. Or also M.Tech students.

Hiring process

18 [0:07:05.9] **S2:** Placements also occur in M.Tech colleges as well. So. Okay. Yeah. Bachelors and masters as well. Yeah. So placements is basically for the college I mean. Yeah. Whichever college you are either it's a B.Tech M.Tech or even not only technology other things like if you are doing a BBA, MBA or some other courses. Yeah. So that also has this placement thing where people usually I mean that is a placement is just a, I mean, kind of, opportunity which comes to college. So if you are in a college and then you want to, further go, you can also ditch placements and then go to higher studies as well if you want. It's your choice. Open to anyonw.

19 [0:07:53.4] **S1:** And how does this placement with [Company name], for example, works? Is the aptitude test and stuff like that?

..Hard skills assesment in hiring
Hiring process
..Aptitude Test
..Soft skills assessment in hiring

20 [0:08:01.0] **S2:** For us during the [Company name] placement, I had I remember I had two online exam, I mean, two levels. One was coding and the other one was aptitude. Okay. And then, the after this, the qualified people, would go through three different interviews, two technical rounds and one hr round.

21 [0:08:25.8] **S1:** Okay. So the coding online exam, what did this look like? What did this consists of?

..Hands-on hiring process with real c

22 [0:08:32.0] **S2:** It, it was basically a similar to what? Codechef. Or those kind of, in heaven. I mean, you get what, coach or something? I mean, the competitive coding platforms are there.

23 [0:08:51.2] **S1:** Yeah, yeah.

24 [0:08:52.0] **S2:** There's a hackerrank hackerearth and leetcode.

25 [0:08:56.3] **S1:** Yeah.

..Hands-on hiring process with real c

26 [0:08:56.5] **S2:** LeetCode kind of.

27 [0:08:57.6] **S1:** Thing. So problem solving?

..Hands-on hiring process with real c

28 [0:08:59.5] **S2:** Yeah. Problem solving will give a problem solving. And there will be some scenarios, test cases which needs to be qualified. I mean the it needs to be passed. So those kind of things are there will be given as a part of coding. I, I remember there was two coding questions. Okay. So we had to solve it in a time. Limited time. Okay. I don't remember the time, but. Yeah. So there was a limited time, maybe half an hour, one, one hour or something. So we had to solve the problem and then all the test cases had to be passed. So yeah, after that I think it is a automated thing which filters out If we are qualified or not. So yeah. But that again can we can opt. There was a column which we can choose if we are comfortable with Python, Java, C plus plus C anything.

29 [0:10:02.1] **S1:** But did you choose.

30 [0:10:03.7] **S2:** I think I chose Java. Yeah. Yeah.

..Aptitude Test

31 [0:10:07.8] **S1:** Nice. And then when you finish this then clear this round. Then you go to aptitude multiple choice test.

..Hands-on hiring process with rea

..Hard skills assesment in hiring

32 [0:10:13.8] **S2:** Yes it was a multiple choice. And yeah. And then after this we had this filter. This was a, I mean, first phase of, selection. Then whoever has qualified, these two things, then we had to go through, technical interview. The first round was again, on demand coding was there. So they usually give up some problems, usually focused on data structures like data structures, and algorithms like are related to arrays, linked lists and stacks, queues all I mean majorly focusing on those parts strings or something. So they usually give some coding problem there on the spot. And we had to give at least. I mean, there's one thing I understood is that people usually I mean, the hiring people, recruiters don't usually want you to solve it entirely. But if I mean, if you give, I mean, if it is 100% solved, then it is very well and good. But, sometimes people also check if they are capable of doing that. Even not, if not now. I mean, because during interviews, we can't, access Google or something, so we can't possibly go and check the syntax or something.

33 [0:11:45.2] **S1:** Okay, so it's ok if...

34 [0:11:46.8] **S2:** But if you at least give the logic behind it how you are going to

..Hard skills assesment in hiring

solve it, or because people usually even the developers usually, take the help of Google syntaxes and all to write the code. But, during the interview time, even if you give a basic logic behind it and how, how many ways you can solve it or those kind of things they usually check for, how readily you can, give the logic or solution for the problem. And also, I mean, not the solution as an output, but the I mean, as I told the logic or how your method of solving or something. So and they check for, are you, I mean, even if you don't know the particular thing, are you, I mean, up to Learn it or something. Or is it? I mean, will you be able to solve it in the future itself? I mean, even if you take the help of something, will you be able to. Because there are people who may not, even if you give a Google or something, you may not solve the problem. But yeah. So interviewers I what I understood is that they usually check for that that are you able to solve that or not? I mean, even if in the future, not now, even if in the future, if you would be able to do that or not.

35 [0:13:12.4] **S1:** Okay. Can we say, please correct me if I'm wrong, but I get they are checking for problem solving skills somehow.

36 **S2:** Yes

37 **S1:** Okay, yeah. So this is the technical round. You get a coding problem is. And then what happens?

..Hard skills assesment in hiring

38 [0:13:29.0] **S2:** I mean, apart from the coding program, I mean, it again, depends on the interviewer who is taking your interview. Sometimes they only want your theoretical knowledge. I mean, conceptual knowledge and all. So they will try to, I mean, ask your conceptual, I mean, any conceptual questions like, what is this? What is that or what is oops and all.

39 [0:13:54.8] **S1:** Okay.

40 [0:13:56.2] **S2:** What is this architecture or how you do that? What is time? I mean, how you will find the time complexity for this and that and those kind of things concepts basically. So that again depends on the interviewer who is taking the interview. So if he is more interested in, know how you are solving, how you are able to solve the, I mean, question and all, they may give you any coding questions or they will ask you conceptual questions. So my interview there were there were two, technical interviews. So one technical interview, it was entirely coding. They gave me 2 or 3 codes, then asked me to solve that. And then the second interview was entirely conceptual. It was around 40 to 50 minutes of interview and everything was concepts. What is this? What is that?

..Hard skills assesment in hiring

APPENDIX F

# Observations

## F.1 Appendix E – Full Observation Log

| ID | Location | Theme | Description and Reflexive Note |
|---|---|---|---|
| **O1** | College | Power outages | Frequent power and internet outages interrupted lectures and lab work. Reflexive: Frequent infrastructure failures disrupt the learning flow significantly. |
| **O2** | Lab Room | Lack of laptops / devices | Students often share computers or borrow laptops from peers. Reflexive: Unequal access to devices limits practical learning opportunities. |
| **O3** | Faculty Office | Faculty values and attitudes | Faculty emphasize discipline and loyalty more than creativity or fun in learning. Reflexive: Hierarchical teaching culture dominates the classroom dynamic. |
| **O4** | Library | Financial burden of families | Several students mentioned that their families had to take out loans to pay tuition fees. Reflexive: Financial stress increases pressure on student performance. |
| **O5** | Campus | Career ambitions | Many students express the dream of working for global technology companies. Reflexive: Career aspirations are often idealized; more structured career guidance is needed. |
| **O6** | Lab Room | Communication behavior | Students rarely ask questions and appear hesitant to engage in discussion. Reflexive: Hierarchical norms suppress active participation and questioning. |
| **O7** | Staff Room | Admission test irregularities | Reports of cheating and informal assistance during admission tests. Reflexive: Exam systems appear formal but not always fair or transparent. |
| **O8** | Library | Politeness norms | Students avoid criticism and tend to agree with authority. Reflexive: Social harmony is valued over open debate. |

| ID | Location | Theme | Description and Reflexive Note |
|---|---|---|---|
| **O9** | Campus | Social desirability bias | Interview responses often appear overly positive or tailored to please the interviewer. Reflexive: Social desirability must be considered when interpreting interview data. |
| **O10** | CSE Classroom | Student motivation | Despite limited resources, students show strong intrinsic motivation to learn. Reflexive: Their enthusiasm signals substantial unrealized potential. |
| **O11** | Faculty Office / Campus | Collaboration with external mentors | Faculty express willingness to collaborate internationally but cite time and funding constraints. Reflexive: Motivation for exchange is high, but structural support remains insufficient. |
| **O12** | RIT Sessions | Student engagement in mentoring | Students react enthusiastically to mentoring and real-world projects. Reflexive: Exposure to applied contexts significantly boosts learning motivation and confidence. |
| **O13** | Campus | Programming competence | Students struggle to complete basic practical programming tasks without mentor support. Reflexive: The lack of foundational programming skills indicates that theoretical instruction does not translate into practical ability. |
| **O14** | Campus | Theory–practice imbalance | Lectures focus predominantly on theoretical content with limited opportunities for applied programming. Reflexive: The strong emphasis on theory constrains students' ability to apply knowledge in real-world problem-solving. |

Table F.1: Comprehensive Observation Log recorded at the partner college (Oct–Jan 2024/25).

APPENDIX G

# Interview Consent Form

# Consent for Participation in Interview Research

Name of Research Project: **Aligning Educational Outcomes with Industry Needs: A Study on Computer Science Education Exemplified by a College in Rural India**

I volunteer to participate in a research project conducted by Paul Spiesberger and Johannes Hufnagl from TU Wien. I understand that the project is designed to gather information about academic work of faculty and students on campus.

1. My participation in this project is voluntary. I understand that I will not be paid for my participation. I may withdraw and discontinue participation at any time without penalty. If I decline to participate or withdraw from the study, no one on my campus will be told.

2. I understand that most interviewees in will find the discussion interesting and thought-provoking. If, however, I feel uncomfortable in any way during the interview session, I have the right to decline to answer any question or to end the interview.

3. Participation involves being interviewed by researchers. The interview will last approximately 1 hour. Notes will be written during the interview. An audio tape of the interview and subsequent dialogue will be made. If I don't want to be taped, only notes will be taken. If you decide that you would like your recording to be deleted, you have the right to demand so at any time.

4. I understand that the researcher will not identify me by name in any reports using information obtained from this interview, and that my confidentiality as a participant in this study will remain secure. Subsequent uses of records and data will be subject to standard data use policies which protect the anonymity of individuals and institutions.

5. Faculty and administrators from my campus will neither be present at the interview nor have access to raw notes or transcripts. This precaution will prevent my individual comments from having any negative repercussions.

6. I have read and understand the explanation provided to me. I have had all my questions answered to my satisfaction, and I voluntarily agree to participate in this study.

7. I have been given a copy of this consent form.

| Name Participant | Name Researcher |
|---|---|
| Date, Signature Participant | Date, Signature Researcher |



If you have any questions, please reach out to us:
paul.spiesberger@inso-world.com

APPENDIX H

# Overview of Generative AI Tools Used

Throughout the writing process, generative and assistive AI tools were used transparently and responsibly to enhance linguistic clarity, structural coherence, and stylistic consistency. These tools supported the author in phrasing, text refinement, and presentation, but the overall conceptual, analytical, and empirical work—including the generation of ideas, argumentation, and interpretation—remained under the author's intellectual control.

**ChatGPT (OpenAI, 2025).** ChatGPT[1] was used as an academic assistant for rephrasing, improving fluency, and verifying logical and terminological consistency. It was also occasionally employed to optimize formatting and to assist in structuring tabular and illustrative material. The tool's function was supportive in nature; all substantive content and argumentation originated from the author.

**QuillBot (QuillBot Inc., 2025).** QuillBot[2] was selectively applied to refine sentence flow and grammatical accuracy in academic English, particularly to improve readability. The semantic and conceptual content remained unaltered and under the author's supervision.

**Grammarly (Grammarly Inc., 2025).** Grammarly[3] served as a proofreading aid for detecting typographical and stylistic inconsistencies. It complemented the author's manual review to ensure a clear and polished linguistic presentation.

[1] `https://chat.openai.com`
[2] `https://quillbot.com`
[3] `https://grammarly.com`

**Ethical and Scientific Integrity.** The use of all AI tools complied with TU Wien's standards of academic integrity. Their contribution was limited to editorial and stylistic assistance. All outputs were reviewed, edited, and approved by the author, ensuring that the author's intellectual and creative contribution remained predominant.

## H.1 Übersicht verwendeter Hilfsmittel

Während des gesamten Schreibprozesses wurden generative und assistierende KI-Werkzeuge transparent und verantwortungsvoll eingesetzt, um sprachliche Klarheit, strukturelle Kohärenz und stilistische Einheitlichkeit zu unterstützen. Diese Werkzeuge dienten der sprachlichen und formalen Verfeinerung, während die inhaltliche, analytische und empirische Ausarbeitung – einschließlich Ideenfindung, Argumentation und Interpretation – vollständig unter der intellektuellen Kontrolle des Autors blieb.

**ChatGPT (OpenAI, 2025).** ChatGPT[4] wurde als akademisches Assistenzwerkzeug genutzt, um Formulierungen zu überarbeiten, den sprachlichen Fluss zu verbessern und die logische sowie terminologische Konsistenz zu überprüfen. Gelegentlich wurde es auch zur Optimierung von Formatierungen oder zur Unterstützung bei der Erstellung von Tabellen und Illustrationen eingesetzt. Die inhaltlichen Aussagen, Argumente und Ergebnisse stammen vollständig vom Autor.

**QuillBot (QuillBot Inc., 2025).** QuillBot[5] wurde punktuell eingesetzt, um die Satzflüssigkeit und grammatikalische Genauigkeit im akademischen Englisch zu verbessern und die Lesbarkeit zu erhöhen. Der Bedeutungsgehalt der Aussagen blieb unverändert und unterlag der Kontrolle des Autors.

**Grammarly (Grammarly Inc., 2025).** Grammarly[6] diente als unterstützendes Korrekturwerkzeug zur Erkennung von Tippfehlern, Grammatik- und Stilabweichungen. Es ergänzte die manuelle Überprüfung des Autors, um eine klare und sprachlich saubere Darstellung sicherzustellen.

**Ethische und wissenschaftliche Integrität.** Die Nutzung aller KI-Werkzeuge entsprach den Grundsätzen wissenschaftlicher Integrität der TU Wien. Ihr Beitrag beschränkte sich auf redaktionelle und stilistische Unterstützung. Alle generierten Ergebnisse wurden vom Autor geprüft, überarbeitet und freigegeben, sodass der schöpferische und wissenschaftliche Beitrag des Autors stets im Vordergrund stand.

[4] https://chat.openai.com
[5] https://quillbot.com
[6] https://grammarly.com

APPENDIX I

# Guidance Notes – Version 1

INSO - Industrial Software
Institute of Computer Aided Automation | Faculty of Informatics | Vienna University of Technology

# Guidance Note

# Guidance Note for Computer Science Faculty at Rural Higher Education Institutions in India

## Enhancing Employability through Curriculum and Teaching Alignment



Developed by: Johannes Hufnagl, Paul Spiesberger, Thomas Grechenig
December 2025

# 1 Context

Computer science graduates from rural Indian colleges continue to enter the job market with significant skill gaps—particularly in programming, problem solving, communication, and practical software engineering. This mismatch between academic training and industry expectations results in low employability and high post-hire training costs for companies [4].

**Caveat:** Solutions to problems differ and heavily depend on the context of each Higher Education Institution. The here stated solutions can only be seen as proposals and require constant on site evaluation and adaptation.

## 1.1 Computer Science Education Faculty and Target Audience

The primary audience of these notes are lecturers, trainers, and assistant professors teaching Computer Science Education courses in affiliated colleges under state universities. Although a curriculum is predefined, faculty members have the autonomy in terms of teaching methods, assessments, and student engagement. Their pedagogical decisions strongly influence the employability and confidence of graduates [4].
These Guidance Notes aim to help educators:

- strengthen the link between theoretical instruction and practical application;
- foster employability skills and professional behaviour;
- and establish sustainable collaboration with industry partners.

## 1.2 Scope and Adaptation

The recommendations are based on one affiliated college in rural Andhra Pradesh, India and reflect the conditions of similar Higher Education Institutions with limited resources and large class sizes. Therefore, they should be adapted to local institutional realities and regularly reviewed with students, faculty, and industry partners.

# 2 Guidance

The following recommendations focus on feasible, low-cost actions that Computer Science Education faculty can implement within existing institutional constraints [4, 8, 1].

1. **Strengthen practical learning.** Faculty should systematically connect theory to practice by integrating small programming projects, group exercises, and open-ended assignments into laboratory work. In doing so, faculty are encouraged to move away from purely lecture-based (frontal) teaching and adopt more student-centered approaches such as flipped classroom and project-based learning, or at minimum a blended combination of these methods.

   Encourage students to practice on online coding platforms such as HackerRank[1] or LeetCode[2] to improve their applied problem-solving skills. To better prepare them for industry expectations, lab sessions should additionally incorporate essential software engineering

[1] `https://www.hackerrank.com`
[2] `https://leetcode.com`

tools and workflows, including version control with Git[3], collaborative development practices (e.g. pair programming and code reviews), basic testing, and the use of issue trackers or simple ticket systems.

*Example:* Short exercises (e.g. loops, conditionals, arrays) should be introduced with small hands-on tasks that students implement immediately and then be integrated into a single cumulative semester project.

2. **Foster soft skills and professional readiness.** Communication, teamwork, and adaptability are key hiring criteria. Include presentations, peer feedback, and mock interviews to simulate professional environments and give students repeated practice in speaking and collaborating. *Example:* At the end of each module, ask student teams to give a 5-minute project pitch followed by short peer feedback on clarity, structure, and teamwork.

3. **Promote industry collaboration.** Initiate partnerships with local IT firms, alumni, or tech experts to organize guest lectures, workshops, and mentoring. Small collaborations can already increase institutional visibility and student employability.

   Students may work in small teams on *applied mini-projects* for local companies ("micro research projects"), giving firms early access to talent while providing students with real-world experience.

   Faculty can also facilitate *micro consulting sessions*, where external experts review student work and help build a basic expert network around the institution.

   *Example:* Invite one alumnus per semester to discuss recruitment and skill expectations and combine this with a small, company-inspired student project.

4. **Encourage continuous faculty learning.** Faculty development strengthens teaching quality and keeps subject knowledge up to date. In addition to online courses (e.g. NPTEL[4], Coursera[5]) and peer exchange, faculty are encouraged to not only teach theory, but also implement all assigned tasks themselves.

   A combined *learning and software development circle* can support this: faculty jointly complete online modules and also implement the same programming exercises and projects given to students. Where possible, institutions should enable short-term collaboration with software firms to provide faculty with practical industry exposure.

   *Example:* Establish a small faculty circle that completes one NPTEL course per year and regularly meets to review code, discuss challenges, and align teaching practices.

5. **Support student self-learning.** Motivate students to engage in self-paced online learning and recognize certificates or project work as part of internal assessments where appropriate. Students learn most effectively when they can experiment and try things out themselves; this requires adequate *time* and *access to resources*. Institutions should therefore ensure that computer labs are open beyond class hours so students have the opportunity to explore, practice, and "play" with software development tools. Provide guidance on selecting high-quality resources and balancing extra learning with workload.

   *Example:* Allow students to submit one online course or coding-platform project per semester (plus a short reflection) as a small share of their internal marks.

[3] https://git-scm.com
[4] https://nptel.ac.in
[5] https://www.coursera.org

6. **Integrate ethical and social awareness.** Discuss ethics, data privacy, and the societal implications of technology in class, using examples that are locally relevant to students' lives. Promote a *Free and Open-Source Software (FOSS) mindset* by encouraging students to use open-source tools, share code openly, and contribute to or start small FOSS projects; this not only supports ethical and transparent development practices but also simulates real collaborative "company-like" workflows. Encourage student participation in local digital literacy or community projects to strengthen social responsibility.

   *Example:* Use a local case of data misuse or online fraud as a starting point for a class discussion on privacy-by-design, open-source development norms, and responsible software engineering.

7. **Use AI tools as supportive, transparent learning aids.** Treat AI-based tools (e.g. code assistants or chatbots) as complements to, not replacements for, core learning activities. Clearly define when AI use is allowed, ask students to document prompts and outputs, and emphasise that AI-generated results must always be checked and understood. *Example:* In programming courses, allow students to use an AI assistant for debugging, but require them to submit their prompts, explain which suggestions they kept or rejected, and verify the final code with tests.

## 3 Conclusion

These recommendations highlight what rural faculty can influence most directly: teaching practice, student engagement, and collaboration networks. They provide a realistic framework for improving employability and quality of education in the Indian affiliated-college context while acknowledging institutional constraints and the need for local adaptation [4, 8].

## 4 Further Reading

For additional context and theoretical background:

- **Why students still need to learn to code in the age of AI.** A position paper by the Raspberry Pi Foundation offering clear arguments for coding education, including hands-on learning, computational literacy, and the importance of empowering young people in an AI-driven world. Highly recommended for understanding *why* practical programming should remain central to teaching strategies [3].
- **Veritasium: What Everyone Gets Wrong About AI and Learning.** A clear and accessible explanation by Derek Muller (Veritasium) about how humans learn, why effort and confusion matter, and why AI tools do not replace the need for active learning. Highly relevant for understanding why hands-on coding practice remains essential. `https://www.youtube.com/watch?v=0xS68sl2D70`
- **Raspberry Pi Foundation – Teaching Resources.** A large collection of free, classroom-ready programming projects (Python, Scratch, physical computing), including full units and step-by-step project sheets suitable for resource-constrained environments. `https://www.raspberrypi.org/teach/`
- **Experience-CS Curriculum.** A project-based computing curriculum with concrete examples of semester projects, learning trajectories, and rubrics for CS education. Helpful for designing integrated, cumulative programming projects. `https://experience-cs.org/`

- **Scratch, PocketCode and visual programming tools.** Beginner-friendly environments that help students internalise algorithmic thinking, sequencing, logic, and event-driven programming before transitioning to text-based languages. Useful for first-year students with limited prior exposure. `https://scratch.mit.edu/` `https://catrob.at/pocketcode`
- **GitHub Education.** A globally accessible programme providing free access to developer tools, cloud platforms, and classroom management features (GitHub Classroom). Extremely relevant for integrating professional workflows (Git, repositories, reviews) into teaching. `https://github.com/education`
- **What schools don't teach (Code.org).** A short and influential video motivating why programming education matters; ideal for motivating students and faculty. `https://www.youtube.com/watch?v=0xS68sl2D70`
- **Aligning educational outcomes with industry needs.** The master's thesis underlying these Guidance Notes provides an in-depth qualitative analysis of curriculum–industry alignment at a rural affiliated college in Andhra Pradesh and forms the empirical foundation for the recommendations presented here [4].
- **Guidance Notes and curriculum alignment in rural India.** A master's thesis from TU Wien that develops guidance notes for curriculum designers and managers and analyses Design–Reality Gaps in rural Indian higher education [1].
- **Design–Reality Gaps in rural software engineering education.** An ICT4D 2023 paper identifying five design–reality gaps in rural higher education and discussing curriculum–industry alignment challenges [8].
- **Bridging the Skill Gap Between Academia and Industry in Punjab: A Case Study.** A case study from another part of India [6].
- **Student assessment of quality of engineering education in India: evidence from a field survey.** Quality of higher engineering education from the students' perspective in India [2].
- **Engineering Graduates through Experiential Learning Programs and Competitive Programming: Case Study.** A paper that demonstrates that experiential learning and competitive programming initiatives substantially enhance the employability of Indian engineering students through practical, problem-oriented skill development [7].
- **Industry-linked graduate software engineering curriculum.** An article arguing that software engineering programs should be tightly aligned with industry needs and enriched with real-world projects to improve graduates' practical readiness [5].

## References


[1] Lukas Bürstmayr. "Bridging the Gap between Computer Science Education and Industry Demands in Rural India". Master thesis. Wien, Österreich: Technische Universität Wien, Fakultät für Informatik, 2022.

[2] Pradeep Choudhury. "Student assessment of quality of engineering education in India: evidence from a field survey". In: *Quality Assurance in Education* 27 (Feb. 2019). DOI: `10.1108/QAE-02-2015-0004`.

[3] Philip Colligan, Mark Griffiths, and Veronica Cucuiat. *Why kids still need to learn to code in the age of AI.* Tech. rep. Position Paper. Raspberry Pi Foundation, June 2025. URL: https://www.raspberrypi.org/.

[4] Johannes Hufnagl. "Aligning Educational Outcomes with Industry Needs: A Study on Computer Science Education Exemplified by a College in Rural India". Advisors: Thomas Grechenig; Paul Spiesberger. Master thesis. Wien, Österreich: Technische Universität Wien, 2025.

[5] Alok Mishra and Deepti Mishra. "Industry Linked Graduate Software Engineering Curriculum". In: *SIGSOFT Softw. Eng. Notes* 36.6 (Nov. 2011), pp. 1–4. ISSN: 0163-5948. DOI: 10.1145/2047414.2047429.

[6] Gulam B. Mustafa et al. "Bridging the Skill Gap Between Academia and Industry in Punjab: A Case Study". In: *Proceedings of the 12th International Conference on Education Technology and Computers.* ICETC '20. New York, NY, USA: Association for Computing Machinery, Mar. 2021, pp. 239–245. ISBN: 978-1-4503-8827-6. DOI: 10.1145/3436756.3437052. (Visited on 06/28/2024).

[7] Prashant R. Nair. "Increasing Employability of Indian Engineering Graduates through Experiential Learning Programs and Competitive Programming: Case Study". In: *Procedia Computer Science* 172 (2020). 9th World Engineering Education Forum (WEEF 2019) Proceedings : Disruptive Engineering Education for Sustainable Development, pp. 831–837. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2020.05.119. URL: https://www.sciencedirect.com/science/article/pii/S1877050920314484.

[8] Paul Spiesberger, Lukas Bürstmayr, and Thomas Grechenig. "Identifying Higher Software Engineering Education's Design–Reality Gaps in Rural India". In: *Proceedings of the 15th International Conference on Information and Communication Technologies and Development (ICTD 2023).* New York, NY, USA: ACM, 2023, pp. 1–11. DOI: 10.1145/3586788.3607033.

APPENDIX J

# Guidance Notes – Diff-File (V1 vs. Final)

INSO - Industrial Software
Institute of Computer Aided Automation | Faculty of Informatics | Vienna University of Technology

# Guidance Note

## Guidance Note for Computer Science Faculty at Rural Higher Education Institutions in India

### Enhancing Employability through Curriculum and Teaching Alignment

Developed by: Johannes Hufnagl, Paul Spiesberger, Thomas Grechenig
December 2025

# 1 Context

Computer science graduates from rural Indian colleges continue to enter the job market with significant skill gaps—particularly in programming, problem solving, communication, and practical software engineering. This mismatch between academic training and industry expectations results in low employability and high post-hire training costs for companies. [4].

**Caveat:** Solutions to problems differ and heavily depend on the context of each Higher Education Institution. The here stated solutions can only be seen as proposals and require constant on site evaluation and adaptation.

## 1.1 Computer Science Education Faculty and Target Audience

The primary audience of these notes are lecturers, trainers, and assistant professors teaching Computer Science Education courses in affiliated colleges under state universities. Although a curriculum is predefined, faculty members have the autonomy in terms of teaching methods, assessments, and student engagement. Their pedagogical decisions strongly influence the employability and confidence of graduates [4].

These Guidance Notes aim to help educators:

- strengthen the link between theoretical instruction and practical application;
- foster employability skills and professional behaviour;
- and establish sustainable collaboration with industry partners.

## 1.2 Scope and Adaptation

The recommendations are based on one affiliated college in rural Andhra Pradesh, India and reflect the conditions of similar Higher Education Institutions with limited resources and large class sizes. Therefore, they should be adapted to local institutional realities and regularly reviewed with students, faculty, and industry partners.

# 2 Guidance

The following recommendations focus on feasible, low-cost actions that Computer Science Education faculty can implement within existing institutional constraints [4, 8, 1].

1. **Strengthen practical learning.** Faculty should systematically connect theory to practice by integrating small programming projects, group exercises, and open-ended assignments into laboratory work. In doing so, faculty are encouraged to move away from purely lecture-based (frontal) teaching and adopt more student-centered approaches such as flipped classroom and project-based learning, or at minimum a blended combination of these methods.

   Encourage students to practice on online coding platforms such as HackerRank[1] or LeetCode[2] to improve their applied problem-solving skills. To better prepare them for industry expectations, lab sessions should additionally incorporate essential software engineering

[1] https://www.hackerrank.com
[2] https://leetcode.com

(Overlaid earlier version, printed in cyan:)

In any case those solutions can only be seen as proposals and require constant on site evaluation and adaptation.

## 1.1 Computer Science Education Faculty and Target Audience

These Guidance Notes aim to help educators:

- strengthen the link between theoretical instruction and practical application;
- foster employability skills and professional behaviour;
- establish sustainable collaboration with industry partners;
- and want to teach programming in a more practical way

Lecturers, trainers, and assistant professors teaching Computer Science who face the following challenges may find practical solutions in these Guidance Notes:

- struggling to translate theoretical content into hands-on programming exercises and meaningful lab work;
- noticing that students lack basic coding proficiency and problem-solving skills needed for industry-style tasks;
- having limited ... workflows (e.g. Git, testing, collaborative development);
- seeking simple, actionable ways to foster communication, teamwork, and professional behaviour in class.

## 1.2 Scope and Adaptation

The recommendations are based on an affiliated college in rural Andhra Pradesh, India as well as insight from industry and literature and reflect the conditions of similar Higher Education Institutions in India with limited resources and large class sizes. Therefore, they should be adapted to local institutional realities and regularly reviewed with students, faculty, and industry partners.

tools and workflows, including version control with Git[3], collaborative development practices (e.g. pair programming and code reviews), basic testing, and the use of issue trackers or simple ticket systems.

*Example:* Short exercises (e.g. loops, conditionals, arrays) should be introduced with small hands-on tasks that students implement immediately and then be integrated into a single cumulative semester project.

2. **Foster soft skills and professional readiness.** Communication, teamwork, and adaptability are key hiring criteria. Include presentations, peer feedback, and mock interviews to simulate professional environments and give students repeated practice in speaking and collaborating. *Example:* At the end of each module, ask student teams to give a 5-minute project pitch followed by short peer feedback on clarity, structure, and teamwork.

3. **Promote industry collaboration.** Initiate partnerships with local IT firms, alumni, or tech experts to organize guest lectures, workshops, and mentoring. Small collaborations can already increase institutional visibility and student employability.

   Students may work in small teams on *applied mini-projects* for local companies ("micro research projects"), giving firms early access to talent while providing students with real-world experience.

   Faculty can also facilitate *micro consulting sessions*, where external experts review student work and help build a basic expert network around the institution.

   *Example:* Invite one alumnus per semester to discuss recruitment and skill expectations and combine this with a small, company-inspired student project.

4. **Encourage continuous faculty learning.** Faculty development strengthens teaching quality and keeps subject knowledge up to date. In addition to online courses (e.g. NPTEL[4], Coursera[5]) and peer exchange, faculty are encouraged to not only teach theory, but also implement all assigned tasks themselves.

   A combined *learning and software development circle* can support this: faculty jointly complete online modules and also implement the same programming exercises and projects given to students. Where possible, institutions should enable short-term collaboration with software firms to provide faculty with practical industry exposure.

   *Example:* Establish a small faculty circle that completes one NPTEL course per year and regularly meets to review code, discuss challenges, and align teaching practices.

5. **Support student self-learning.** Motivate students to engage in self-paced online learning and recognize certificates or project work as part of internal assessments where appropriate. Students learn most effectively when they can experiment and try things out themselves; this requires adequate *time* and *access to resources*. Institutions should therefore ensure that computer labs are open beyond class hours so students have the opportunity to explore, practice, and "play" with software development tools. Provide guidance on selecting high-quality resources and balancing extra learning with workload.

   *Example:* Allow students to submit one online course or coding-platform project per semester (plus a short reflection) as a small share of their internal marks.

[3] https://git-scm.com
[4] https://nptel.ac.in
[5] https://www.coursera.org

6. **Integrate ethical and social awareness.** Discuss ethics, data privacy, and the societal implications of technology in class, using examples that are locally relevant to students' lives. Promote a *Free and Open-Source Software (FOSS) mindset* by encouraging students to use open-source tools, share code openly, and contribute to or start small FOSS projects; this not only supports ethical and transparent development practices but also simulates real collaborative "company-like" workflows. Encourage student participation in local digital literacy or community projects to strengthen social responsibility.
   *Example:* Use a local case of data misuse or online fraud as a starting point for a class discussion on privacy-by-design, open-source development norms, and responsible software engineering.

7. **Use AI tools as supportive, transparent learning aids.** Treat AI-based tools (e.g. code assistants or chatbots) as complements to, not replacements for, core learning activities. Clearly define when AI use is allowed, ask students to document prompts and outputs, and emphasise that AI-generated results must always be checked and understood.
   *Example:* In programming courses, allow students to use an AI assistant for debugging, but require them to submit their prompts, explain which suggestions they kept or rejected, and verify the final code with tests.

## 3 Conclusion

These recommendations highlight what rural faculty can influence most directly: teaching practice, student engagement, and collaboration networks. They provide a realistic framework for improving employability and quality of education in the Indian affiliated-college context while acknowledging institutional constraints and the need for local adaptation [4, 8].

## 4 Further Reading

For additional context and theoretical background:

- **Why students still need to learn to code in the age of AI.** A position paper by the Raspberry Pi Foundation offering clear arguments for coding education, including hands-on learning, computational literacy, and the importance of empowering young people in an AI-driven world. Highly recommended for understanding *why* practical programming should remain central to teaching strategies [3].
- **Veritasium: What Everyone Gets Wrong About AI and Learning.** A clear and accessible explanation by Derek Muller (Veritasium) about how humans learn, why effort and confusion matter, and why AI tools do not replace the need for active learning. Highly relevant for understanding why hands-on coding practice remains essential. `https://www.youtube.com/watch?v=0xS68sl2D70`
- **Raspberry Pi Foundation – Teaching Resources.** A large collection of free, classroom-ready programming projects (Python, Scratch, physical computing), including full units and step-by-step project sheets suitable for resource-constrained environments. `https://www.raspberrypi.org/teach/`
- **Experience-CS Curriculum.** A project-based computing curriculum with concrete examples of semester projects, learning trajectories, and rubrics for CS education. Helpful for designing integrated, cumulative programming projects. `https://experience-cs.org/`

- **Scratch, PocketCode and visual programming tools.** Beginner-friendly environments that help students internalise algorithmic thinking, sequencing, logic, and event-driven programming before transitioning to text-based languages. Useful for first-year students with limited prior exposure. https://scratch.mit.edu/ https://catrob.at/pocketcode
- **GitHub Education.** A globally accessible programme providing free access to developer tools, cloud platforms, and classroom management features (GitHub Classroom). Extremely relevant for integrating professional workflows (Git, repositories, reviews) into teaching. https://github.com/education
- **What schools don't teach (Code.org).** A short and influential video motivating why programming education matters; ideal for motivating students and faculty. https://www.youtube.com/watch?v=0xS68sl2D70
- **Aligning educational outcomes with industry needs.** The master's thesis underlying these Guidance Notes provides an in-depth qualitative analysis of curriculum–industry alignment at a rural affiliated college in Andhra Pradesh and forms the empirical foundation for the recommendations presented here [4].
- **Guidance Notes and curriculum alignment in rural India.** A master's thesis from TU Wien that develops guidance notes for curriculum designers and managers and analyses Design–Reality Gaps in rural Indian higher education [1].
- **Design–Reality Gaps in rural software engineering education.** An ICT4D 2023 paper identifying five design–reality gaps in rural higher education and discussing curriculum–industry alignment challenges [8].
- **Bridging the Skill Gap Between Academia and Industry in Punjab: A Case Study.** A case study from another part of India [6].
- **Student assessment of quality of engineering education in India: evidence from a field survey.** Quality of higher engineering education from the students' perspective in India [2].
- **Engineering Graduates through Experiential Learning Programs and Competitive Programming: Case Study.** A paper that demonstrates that experiential learning and competitive programming initiatives substantially enhance the employability of Indian engineering students through practical, problem-oriented skill development [7].
- **Industry-linked graduate software engineering curriculum.** An article arguing that software engineering programs should be tightly aligned with industry needs and enriched with real-world projects to improve graduates' practical readiness [5].

## References


[1] Lukas Bürstmayr. "Bridging the Gap between Computer Science Education and Industry Demands in Rural India". Master thesis. Wien, Österreich: Technische Universität Wien, Fakultät für Informatik, 2022.

[2] Pradeep Choudhury. "Student assessment of quality of engineering education in India: evidence from a field survey". In: *Quality Assurance in Education* 27 (Feb. 2019). DOI: 10.1108/QAE-02-2015-0004.




Clearly define when AI use is allowed, ask students to document prompts and outputs, and emphasise that AI-generated results must always be checked and understood.

*Example.* In programming courses, allow students to use an AI assistant for debugging, but require them to submit their prompts, explain which suggestions they kept or rejected, and verify the final code with tests.

*See further reading: 3.1, 3.2, 3.6*

# 3 Further Reading

For additional context and theoretical background, grouped into thematic fields:

## 3.1 Teaching Programming and Course Structure

1.1 **Raspberry Pi Foundation – Teaching Resources.** Ready-to-use Python and low-cost project materials suitable for constrained environments. https://www.raspberrypi.org/teach/

1.2 **Experience CS Curriculum.** A project-based computing curriculum with learning trajectories, and rubrics for CS education. Helpful for designing integrated, cumulative programming projects. https://experience-cs.org/

1.3 **What schools don't teach (Code.org).** A short and influential video motivating why programming education matters; ideal for motivating students and faculty. https://www.youtube.com/watch?v=0xS68sl2D70

1.4 **Teaching Methods Overview (Teach.com).** A practical introduction to common teaching strategies including collaborative learning, inquiry-based learning, flipped classroom, and project-based approaches. https://teach.com/what/teachers-know/teaching-methods/

1.5 **W3Schools Academy.** Beginner-friendly, interactive tutorials for HTML, CSS, JavaScript, Python, SQL, and more. Ideal for foundational programming labs. https://www.w3schools.com/academy/academy_get_started.php

1.6 **FreeCodeCamp.** A comprehensive, fully free, project-based curriculum covering full-stack development. Includes certificates and large practical assignments. https://www.freecodecamp.org

1.7 **CS Unplugged.** A widely used collection of activities to teach computing concepts without using computers—especially useful for limited lab availability. https://www.csunplugged.org/en/

1.8 **Flipped Classroom Explained Simply (YouTube).** A short and clear visual explanation of how the flipped classroom model works, why it improves student engagement, and how faculty can gradually adopt it even with limited resources. https://www.youtube.com/watch?v=adKzSa_t8k8

[3] Philip Colligan, Mark Griffiths, and Veronica Cucuiat. *Why kids still need to learn to code in the age of AI*. Tech. rep. Position Paper. Raspberry Pi Foundation, June 2025. URL: https://www.raspberrypi.org/.

[4] Johannes Hufnagl. "Aligning Educational Outcomes with Industry Needs: A Study on Computer Science Education Exemplified by a College in Rural India". Advisors: Thomas Grechenig; Paul Spiesberger. Master thesis. Wien, Österreich: Technische Universität Wien, 2025.

[5] Alok Mishra and Deepti Mishra. "Industry Linked Graduate Software Engineering Curriculum". In: *SIGSOFT Softw. Eng. Notes* 36.6 (Nov. 2011), pp. 1–4. ISSN: 0163-5948. DOI: 10.1145/2047414.2047429.

[6] Gulam B. Mustafa et al. "Bridging the Skill Gap Between Academia and Industry in Punjab: A Case Study". In: *Proceedings of the 12th International Conference on Education Technology and Computers*. ICETC '20. New York, NY, USA: Association for Computing Machinery, Mar. 2021, pp. 239–245. ISBN: 978-1-4503-8827-6. DOI: 10.1145/3436756.3437052. (Visited on 06/28/2024).

[7] Prashant R. Nair. "Increasing Employability of Indian Engineering Graduates through Experiential Learning Programs and Competitive Programming: Case Study". In: *Procedia Computer Science* 172 (2020). 9th World Engineering Education Forum (WEEF 2019) Proceedings : Disruptive Engineering Education for Sustainable Development, pp. 831–837. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2020.05.119. URL: https://www.sciencedirect.com/science/article/pii/S1877050920314484.

[8] Paul Spiesberger, Lukas Bürstmayr, and Thomas Grechenig. "Identifying Higher Software Engineering Education's Design–Reality Gaps in Rural India". In: *Proceedings of the 15th International Conference on Information and Communication Technologies and Development (ICTD 2023)*. New York, NY, USA: ACM, 2023, pp. 1–11. DOI: 10.1145/3586788.3607033.

1.9 [illegible] transitioning to text-based languages. Useful for first-year students with limited prior exposure. https://scratch.mit.edu/, https://catrob.at/pocketcode

1.10 **MIT App Inventor.** A block-based app development environment that enables students to build Android apps in weeks. Ideal for mini-projects and practical labs. https://[illegible]

1.11 **GitHub Education.** A globally accessible programme providing free access to developer tools, cloud platforms, and classroom management features (GitHub Classroom). Extremely relevant for integrating professional workflows (Git, repositories, reviews) into teaching. https://github.com/education

1.12 **Essential Soft Skills for Software Engineers (GeeksforGeeks).** [illegible] of communication, teamwork, adaptability, problem-solving, and time management skills that are crucial for employability. https://www.geeksforgeeks.org/blogs/essential-soft-skills-for-software-engineers/

1.13 **What is Free and Open-Source Software? (OpenProject Blog).** An accessible explanation of FOSS principles, why open-source software matters, and how sustainable business models around FOSS work. Supports teaching ethical, transparent, and collaborative software development practices. https://www.openproject.org/blog/[illegible]/

1.14 **Open Educational Resources (OER).** OERs are learning material licensed under the Creative Commons License[8] and allow everyone to use, study, adapt and share learning materials. See UNESCO's recommendations for OERs: https://www.unesco.org/en/legal-affairs/recommendation-open-educational-resources-oer Connect to international and local OER repositories and communities such as https://openverse.org/ and https://oerhub.at/en.

## 3.2 Programming Tools and Workflows

2.1 **LabEx Online Coding Playgrounds.** Browser-based practice environments for Git, Linux, Python, Java, SQL, and algorithm problems—no installation required. https://labex.io

2.2 **GeeksforGeeks – Software and Tools A–Z List.** A broad catalogue of commonly used software tools across programming, DevOps, databases, cloud platforms, and more. Helpful for faculty looking to introduce students to industry-relevant tools. https://www.geeksforgeeks.org/websites-apps/software-and-tools-a-to-z-list/

2.3 **What is Agile Software Development? (YouTube).** A short animated introduction to Agile principles, Scrum roles, sprints, and iterative development. Useful for faculty who want to integrate simple Agile workflows into classroom projects. https://www.youtube.com/watch?v=vYI7-UD9tEQ

[8] https://creativecommons.org/

## 3.3 Artificial Intelligence

3.1 **Why students still need to learn to code in the age of AI.** A blog entry and position paper by the Raspberry Pi Foundation offering clear arguments for coding education, including hands-on learning, computational literacy, and the importance of empowering young people in an AI-driven world. Highly recommended for understanding *why* hands-on programming should remain central to teaching strategies. `https://www.raspberrypi.org/blog/why-kids-still-need-to-learn-to-code-in-the-age-of-ai/`

3.2 **Veritasium: What Everyone Gets Wrong About AI and Learning.** A clear and accessible explanation by Derek Muller (Veritasium) about how humans learn, why effort and confusion matter, and why AI tools do not replace the need for active learning. Highly relevant for understanding why hands-on coding practice remains essential. `https://www.youtube.com/watch?v=0xS68sl2D70`

3.3 **OpenAI Example Prompts for Coding Support.** Demonstrates how students can use AI tools responsibly for debugging, learning concepts, reviewing code, generating test cases, and documenting solutions. `https://platform.openai.com/docs/examples`