

AFRICA UNIVERSITY
(A United Methodist-Related Institution)

EVALUATING THE IMPACT OF AI COPILOTS ON FRAMEWORK USAGE AND
PROFESSIONAL DEVELOPMENT OF STUDENT DEVELOPERS:
A CASE STUDY OF AFRICA UNIVERSITY CEAS STUDENTS.

BY

ALVIN PHIRI

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE BACHELOR OF SCIENCE HONOURS IN COMPUTER SCIENCE DEGREE IN
THE COLLEGE OF ENGINEERING AND APPLIED SCIENCES.

2024

Abstract

The rapid integration of AI Copilots into software development has reshaped how student developers interact with frameworks, make project decisions, and prepare for professional roles. This study investigates the impact of AI Copilots on framework adoption and professional development among student developers at Africa University. Using a quantitative research approach, data was collected through structured surveys from a sample of 80 students. The study examines how AI Copilots influence students' efficiency, learning curves, and project selection, as well as their role preferences within software development. Findings indicate that AI Copilots significantly affect framework selection by reducing learning barriers, leading students to favour more complex but efficient technologies. Additionally, AI assistance influences decision-making in project scope and specialization, with many students shifting toward roles emphasizing AI integration. However, concerns over dependency on AI and diminished problem-solving skills were noted. The study concludes that while AI Copilots enhances productivity and exposure to diverse frameworks, there is a need for a balanced approach to ensure skill development and independent thinking. Recommendations include curriculum updates to incorporate AI literacy and industry partnerships to align AI-assisted learning with professional expectations.

Keywords: AI Copilots, framework adoption, student developers, professional development, software engineering education, AI literacy

Declaration Page

I declare that this dissertation is my original work except where sources have been cited and acknowledged. The work has never been submitted, nor will it ever be submitted to another university for the award of a degree.

Alvin Phiri

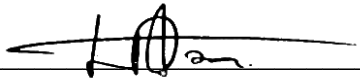
Student's Full Name

Dhlakama Lorence

Main Supervisor's Full Name



Student's Signature



Main supervisor's Signature

Copyright

No part of the dissertation/thesis may be reproduced, stored in any retrieval system, or transmitted in any form or by any means for scholarly purposes without prior written permission of the author or of Africa University on behalf of the author.

List of Acronyms and Abbreviations

AI – Artificial Intelligence

CEAS – College of Engineering and Applied Sciences

TAM – Technology Acceptance Model

Table of Contents

CHAPTER 1 INTRODUCTION	1
1.1. Introduction	1
1.3. Statement of the Problem	2
1.5. Research Questions	4
1.6. Significance of the Study.....	5
1.7. Assumptions of the Study	5
1.8. Delimitations of the Study	7
1.9. Limitations of the Study	8
1.10. Definition of Terms	10
1.11. Dissertation Structure.....	12
1.12. Chapter Summary.....	14
CHAPTER 2: LITERATURE REVIEW	16
2.1 Introduction	16
2.2 Conceptual Framework	16
2.3 Impact of AI Copilots on Framework Usage	19
2.4 Impact of AI Copilots on Project and Role Selection	22
2.5 Influence of AI Copilots on Professional Development	25
2.6 Gap in Knowledge Identified	27
2.7 Chapter Summary.....	29

CHAPTER 3: RESEARCH METHODOLOGY	31
3.1 Introduction	31
3.3 Population and Sampling	33
3.4 Data Collection Instruments	35
3.5 Pilot Study	37
3.6 Data Collection	38
3.7 Analysis and Organization of Data	40
3.8 Ethical Considerations	44
CHAPTER 4 DATA PRESENTATION, ANALYSIS AND INTERPRETATION.....	47
4.1 Introduction	47
4.2 Data Presentation and Analysis	47
CHAPTER 5 SUMMARY, CONCLUSIONS AND RECOMMENDATIONS.....	62
5.1 Introduction	62
5.2 Discussion	62
5.3 Conclusions	64
5.4 Implications.....	66
5.5 Recommendations	68
5.6 Suggestions for Further Research	70
APPENDICES	80
APPENDIX 1 : Questionnaire Survey Instrument	81

APPENDIX 2: Ethical Clearance Approval 83

APPENDIX 3: Data Collection Consent 84

APPENDIX 4: Summarized Raw Survey Data 85

List of Tables

Table 4. 1 Summary of common concerns (e.g., dependency, originality issues).....	57
---	----

List of Figures

Figure 4. 1 : Gender Distribution.....	47
Figure 4. 2 : Year Of Study.....	48
Figure 4. 3 : Age Distribution	48
Figure 4. 4 : Bar chart showing the percentage of students who actively use ai co-pilots versus those who do not.	49
Figure 4. 5 : Pie Chart Illustrating The Frequency Of Ai Co-Pilot Usage.....	50
Figure 4. 6 : Bar Chart Comparing the number of students who switched frameworks due to AI Co-pilots versus those who maintained their previous choices.	51
Figure 4. 7 : Line Graph Showing Trends In Framework Adoption Before And After Ai Co-Pilot Usage.....	52
figure 4. 8 : Bar Graph showing Students' perceptions of AI Co-pilots in skill improvement.....	53
Figure 4. 9 : Bar Chart representing Distribution of responses on whether AI Co-pilots help in professional decision-making.	54
Figure 4. 10 : Pie Chart showing proportion of students selecting specific roles.....	55
Figure 4. 11 : Bar Chart Showing Most preferred project types among AI Co-pilot users versus non-users	56

List of Appendices

APPENDIX 1 : Questionnaire Survey Instrument	81
APPENDIX 2: Ethical Clearance Approval.....	83
APPENDIX 3: Data Collection Consent.....	84
APPENDIX 4: Summarized Raw Survey Data.....	85

CHAPTER 1 INTRODUCTION

1.1 Introduction

In recent years, the advancement of Artificial Intelligence (AI) technologies has impacted various fields, including education and software development. AI Copilots, such as GitHub Copilot, are designed to assist developers by providing context-aware code suggestions, automating repetitive tasks, and improving coding efficiency. These tools have the potential to transform the learning experience for student developers by enhancing their coding practices and influencing their project choices and professional development.

At Africa University's College of Engineering and Applied Sciences (CEAS), the use of AI Copilots among student developers presents an exciting opportunity to explore how these tools impact framework usage, the quality of their work, and their decision-making processes regarding project selection. This study aims to evaluate the influence of AI Copilots on the academic and professional growth of CEAS students, providing insights into their adoption patterns, usage behaviors, and the broader implications for their future careers in software development.

By focusing on a specific academic environment, this research seeks to contribute to the understanding of AI Copilots' role in education and their potential to shape the next generation of software developers. The findings will inform educators, developers, and policymakers about the benefits and challenges associated with the voluntary use of AI-driven tools in computer science and software development programs.

1.2 Background to the Study

The integration of AI technologies, such as AI Copilots, into software development is transforming productivity and learning. These tools offer real-time coding assistance, making them valuable for both professional developers and students. At Africa University's College of Engineering and Applied Sciences (CEAS), many students are voluntarily using AI Copilots in their coursework and projects, even though these tools are not officially part of the curriculum. This study aims to examine how AI Copilots influence the professional development and career readiness of student developers, focusing on their efficiency, project choices, and the broader implications for the software development industry.

1.3 Statement of the Problem

AI Copilots are changing software development by offering context-aware coding help and automating tasks, making them popular among professionals. However, their impact on student developers is less understood. This study will explore how AI Copilots affect students' coding efficiency, project choices, and professional growth. Additionally, it will examine how these tools might shift the relevance of certain developer roles, tech stacks, and frameworks. The goal is to provide insights into how AI Copilots influence the career paths of future developers and shape industry practices.

1.4 Research Aim and Objectives

1.4.1 Research Aim

The primary aim of this study is to evaluate the impact of AI Copilots on the framework usage and professional development of student developers at Africa University's College of Engineering and Applied Sciences (CEAS). This study seeks to understand how these AI tools influence the quality of students' work, their project selection and role selection decisions, and their overall readiness for professional careers in software development. Additionally, the research will explore the broader implications of AI Copilots on the relevance and popularity of various developer roles, tech stacks, and frameworks.

1.4.2 Research Objectives

To achieve this aim, the study will focus on the following specific objectives

1. To investigate how the use of AI Copilots affects the choice and usage of software development frameworks among student developers.
2. To assess the impact of AI Copilots on the quality of code and projects produced by student developers.
3. To explore how AI Copilots influence the types of projects and roles that student developers choose to undertake and their decision-making processes.
4. To evaluate the extent to which AI Copilots prepare student developers for professional roles in the software development industry.

1.5 Research Questions

1.5.1 Primary Research Question

How do AI Copilots impact the framework usage, project selection, role selection, and professional development of student developers at Africa University's College of Engineering and Applied Sciences (CEAS)?

1.5.2 Sub-Research Questions

1. How does the use of AI Copilots affect the choice and usage of software development frameworks among student developers?
2. In what ways do AI Copilots influence the quality of code and projects produced by student developers?
3. How do AI Copilots impact the types of projects and roles that student developers choose to undertake?
4. To what extent do AI Copilots prepare student developers for professional roles in the software development industry?

By addressing these questions, the study aims to provide a detailed understanding of the influence of AI Copilots on the educational and professional experiences of student developers, and to inform strategies for integrating AI tools in software development education.

1.6 Significance of the Study

This study fills a gap in research by focusing on how AI Copilots impact student developers, a group that hasn't been extensively studied in AI-assisted software development. By looking at student experiences at Africa University's CEAS, the study sheds light on how these tools are being used in education.

The findings will have practical benefits for shaping future software developers. Understanding AI Copilots' influence on framework usage, project choices, and coding quality can help educators adjust their teaching methods to better prepare students for the industry's evolving demands.

Additionally, the research will contribute to the broader discussion on AI in education and its potential to change the professional landscape. It will explore how AI Copilots might make some developer roles, tech stacks, and frameworks obsolete while boosting others, offering insights into the future of software development.

Finally, the study will provide valuable information to policymakers and industry leaders about the benefits and challenges of using AI Copilots in education, helping them create policies that enhance learning and ensure students are ready for a tech-driven world.

1.7 Assumptions of the Study

1. It is assumed that student developers at Africa University's College of Engineering and Applied Sciences (CEAS) are using AI Copilots voluntarily and not as a mandatory part of their curriculum. This allows for a natural exploration of how these tools impact their learning and professional development.

2. It is assumed that AI Copilots such as GitHub Copilot or similar tools are readily available and accessible to student developers for use in their projects and coursework.
3. It is assumed that the student developers involved in the study possess a basic to intermediate level of competence in software development, which allows them to effectively utilize AI Copilots in their coding activities.
4. It is assumed that participants will provide honest and accurate responses in surveys, interviews, and other data collection methods. This is crucial for obtaining reliable data regarding their experiences and perceptions.
5. It is assumed that the functionality and performance of the AI Copilots remain consistent throughout the study period, ensuring that variations in results are due to user interaction rather than changes in the tools themselves.
6. It is assumed that student developers undertake a variety of projects and roles, providing a comprehensive view of how AI Copilots influence different aspects of software development.
7. It is assumed that the skills and experiences gained through the use of AI Copilots have a direct impact on the professional readiness of student developers, influencing their future career prospects and roles in the software development industry.

By acknowledging these assumptions, the study aims to provide a clear framework for understanding the influence of AI Copilots on the framework usage and professional development of student developers at CEAS.

1.8 Delimitations of the Study

These are delimitations that define the scope and boundaries of the study.

1. The study is confined to students at Africa University's College of Engineering and Applied Sciences (CEAS). This geographic focus allows for an in-depth analysis of a specific educational context but may limit the generalizability of the findings to other regions or institutions.
2. The research focuses exclusively on student developers who are currently enrolled in computer science or related programs at CEAS. This delimitation ensures that the study concentrates on individuals who are actively learning software development and using AI Copilots in an educational setting.
3. The study examines the use of AI Copilots, such as GitHub Copilot, within the context of software development. It does not extend to other AI tools or technologies that may also assist in coding or related activities.
4. The research is conducted within a specific time frame, which may influence the findings. Changes in AI technology or educational practices that occur after the study period may affect the relevance of the results.
5. The study primarily considers projects and roles that student developers undertake within their coursework or personal learning endeavors. It does not cover professional projects or internships that may have different dynamics and influences.
6. The focus is on evaluating the impact of AI Copilots on framework usage, project and role selection, and professional development. The study does not delve into

other potential impacts of AI Copilots, such as their effects on collaboration, creativity, or ethical considerations in software development.

7. The research employs specific methodologies, including surveys, interviews, and data analysis, to gather and interpret data. Alternative research methods are not explored within the scope of this study.

By clearly defining these delimitations, the study aims to provide a focused and coherent analysis of how AI Copilots influence the professional development of student developers at CEAS, while acknowledging the specific boundaries within which the research is conducted.

1.9 Limitations of the Study

This study aims to provide valuable insights into the impact of AI Copilots on student developers at Africa University's College of Engineering and Applied Sciences (CEAS), and these limitations must be acknowledged.

1. The study's findings are based on a sample of student developers from a single institution. The limited sample size and lack of diversity may affect the generalizability of the results to other educational contexts or geographic regions.
2. The research relies on self-reported data from surveys and interviews, which may be subject to biases such as social desirability bias, recall bias, and inaccurate self-assessment.

3. AI Copilots are rapidly evolving technologies. Changes and updates to these tools during the study period may affect their functionality and impact, potentially influencing the results.
4. Various external factors, such as students' prior experience with software development, access to resources, and individual learning styles, may influence the outcomes and are challenging to control for comprehensively.
5. The study is conducted over a specific time frame, which may limit the ability to observe long-term impacts of AI Copilots on students' professional development and career trajectories.
6. The research is focused on the context of student developers at CEAS, and the findings may not be directly applicable to professional developers, other educational institutions, or different cultural and academic environments.
7. The extent to which individual students adopt and utilize AI Copilots may vary widely, influencing the consistency and comparability of the results.
8. The study focuses on specific functionalities of AI Copilots related to coding assistance. Other potential benefits or drawbacks of AI Copilots, such as their impact on creativity, problem-solving skills, or ethical considerations, are not explored in depth.

1.10 Definition of Terms

To ensure clarity and understanding, the following key terms used in this study are defined.

1. AI Copilots

AI-powered tools that provide real-time coding assistance, suggestions, and automation to developers. Examples include GitHub Copilot and other similar tools.

2. Framework Usage

The adoption and implementation of software development frameworks (such as React, Angular, Django, etc.) by developers in their coding projects.

3. Professional Development

The process by which individuals acquire skills, knowledge, and experiences that prepare them for professional careers in their respective fields.

4. Student Developers

Individuals currently enrolled in computer science or related programs who are learning and practicing software development.

5. Project Selection

The process by which developers choose specific coding projects to work on, influenced by factors such as interest, skill level, and perceived value.

6. Role Selection

The process by which developers identify and assume specific roles within software development projects, such as front-end developer, back-end developer, or full-stack developer.

7. Tech Stacks

Combinations of programming languages, frameworks, and tools used together to develop software applications. Common tech stacks include the MEAN stack (MongoDB, Express.js, Angular, Node.js) and the LAMP stack (Linux, Apache, MySQL, PHP).

8. CEAS (College of Engineering and Applied Sciences)

A college within Africa University that offers programs related to engineering, computer science, and applied sciences.

9. Coding Quality

The degree to which code meets certain standards of functionality, efficiency, readability, and maintainability.

10. Educational Context

The environment and conditions under which learning takes place, including curriculum, teaching methods, and resources available to students.

11. Professional Readiness

The extent to which student developers are prepared to enter and succeed in the professional software development industry, encompassing both technical skills and soft skills.

12. Obsolete

Refers to technologies, frameworks, or roles that are no longer in use or have been replaced by newer alternatives due to advancements in the field.

1.11 Dissertation Structure

This dissertation is organized into six chapters, each addressing different aspects of the research study on the impact of AI Copilots on the framework usage and professional development of student developers at Africa University's College of Engineering and Applied Sciences (CEAS).

Chapter 1: Introduction

This chapter provides an overview of the research topic, background, problem statement, research aims and objectives, research questions, significance, assumptions, delimitations, limitations, and definitions of key terms. It sets the stage for the entire study by outlining its scope and purpose.

Chapter 2: Literature Review

This chapter reviews existing literature related to AI Copilots, software development frameworks, professional development in computer science education, and the use of AI tools in learning environments. It identifies gaps in the current knowledge and situates the study within the broader context of academic research.

Chapter 3: Research Methodology

This chapter describes the research design, methodology, and data collection methods used in the study. It includes details on the sample selection, instruments, procedures, and ethical considerations. The chapter also explains how data will be analyzed to answer the research questions.

Chapter 4: Data Analysis and Results

This chapter presents the results of the data analysis, including statistical findings, patterns, and themes identified through qualitative and quantitative methods. It provides a comprehensive overview of how AI Copilots influence framework usage, project and role selection, coding quality, and professional readiness among student developers.

Chapter 5: Discussion

This chapter interprets the findings in the context of the research objectives and existing literature. It discusses the implications of the results for student developers, educators, and the software development industry. The chapter also explores the broader impact of AI Copilots on the professional landscape of software development.

Chapter 6: Conclusion and Recommendations

This chapter summarizes the key findings of the study, highlights its contributions to knowledge, and offers recommendations for practice, policy, and future research. It emphasizes the importance of integrating AI Copilots in educational settings and suggests ways to enhance their effectiveness in preparing student developers for professional careers.

1.12 Chapter Summary

In this introductory chapter, we laid the foundation for understanding how AI Copilots influence framework usage and the professional development of student developers at Africa University's CEAS. We introduced the topic, discussed the rise of AI Copilots in software development, and identified the research problem, focusing on how these tools affect student learning, project choices, and career readiness. We also outlined the research aims, objectives, and key questions to guide the study.

The chapter highlighted the study's significance, emphasizing its potential impact on education, professional growth, and policy-making. We acknowledged the assumptions

and clearly defined the study's scope and limitations, ensuring a focused investigation. Key terms were clarified, and the dissertation's structure was previewed to provide a roadmap for the chapters ahead, starting with a comprehensive literature review.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

Exploring major themes within the literature related to AI Copilot adoption and its influence on framework usage and professional development among student developers. The review begins with an examination of foundational theories in technology adoption, particularly the Technology Acceptance Model (TAM) and Self-Directed Learning frameworks, which provide a basis for understanding AI integration in educational and professional development contexts. Next, it delves into empirical studies on AI tools in software development, highlighting their impact on framework adoption, project decision-making, and skill advancement. Finally, the chapter addresses ethical and practical challenges in using AI copilots, such as dependency risks and data privacy issues. The review narrows its focus to examining gaps in existing research concerning the evolving role of AI Copilots in shaping students' project choices and professional outlooks, positioning this study to contribute new insights into these underexplored areas.

2.2 Conceptual Framework

The conceptual framework for this study is grounded in the Technology Acceptance Model (TAM), which explains how users come to accept and use a technology. In the context of this study, TAM is utilized to explore the adoption and usage of AI copilots among student developers and how these tools impact their professional development.

2.3 Components of the Conceptual Framework

2.3.1 Perceived Usefulness (PU)

According to the TAM, perceived usefulness is the degree to which a person believes that using a particular system would enhance their job performance. For this study, perceived usefulness relates to how student developers perceive the benefits of AI copilots in improving their coding efficiency, understanding of frameworks, and overall software development skills. Research has shown that tools providing real-time feedback and reducing the learning curve for new frameworks can significantly enhance the learning process (Al-Emran et al., 2020).

2.3.2 Perceived Ease of Use (PEOU)

This refers to the degree to which a person believes that using a system would be free of effort. In the context of AI copilots, perceived ease of use refers to how effortlessly student developers can integrate these tools into their coding practices. User-friendly AI copilots that are easy to learn can significantly influence students' willingness to adopt and continuously use them (Huang & Teo, 2021).

2.3.3 Attitude Toward Use

According to TAM, attitude toward using a technology is influenced by both perceived usefulness and perceived ease of use. In this framework, student developers' attitudes toward using AI copilots will be shaped by how beneficial and easy they perceive these tools to be. Positive attitudes are likely to lead to higher levels of adoption and continued

use, which in turn can affect their learning outcomes and professional development (Khan et al., 2021).

2.3.4 Behavioral Intention to Use

Behavioral intention to use is a key predictor of actual usage behavior in TAM. For student developers, their intention to use AI copilots is expected to be influenced by their attitudes towards these tools, which are shaped by perceived usefulness and ease of use. The stronger the intention to use AI copilots, the more likely students are to integrate them into their learning and development processes (Huang & Teo, 2021).

2.3.5 Actual Use

This refers to the real-world adoption and utilization of AI copilots by student developers. The actual use of these tools will be influenced by students' behavioral intentions and will ultimately impact their framework adoption, professional development, and decision-making regarding project and role selection (Khan et al., 2021).

2.4 Linkages in the Framework

2.4.1 Influence on Framework Adoption and Usage

TAM suggests that perceived usefulness and perceived ease of use will significantly impact student developers' attitudes toward AI copilots, influencing their adoption and effective usage of software development frameworks.

2.4.2 Impact on Professional Development

By influencing students' attitudes and behavioral intentions, AI copilots can play a crucial role in enhancing their professional development, including the acquisition of technical skills, adaptability to new frameworks, and decision-making abilities (Venkatesh, Thong, & Xu, 2012).

2.4.3 Mediating Role of the Educational Environment

The educational environment at Africa University can moderate the relationship between AI copilots and their perceived usefulness and ease of use, potentially affecting students' acceptance and use of these tools.

2.5 Impact of AI Copilots on Framework Usage

This section will be exploring existing literature on the influence of AI copilots on the adoption and usage of various software development frameworks among student developers. It examines how AI copilots affect students' choices in selecting frameworks,

the ease of learning these frameworks, and the overall impact on their professional development.

2.5.1 Influence on Framework Adoption

AI copilots have significantly impacted the adoption of various software development frameworks by student developers. Studies show that AI copilots, by providing real-time code suggestions, error correction, and best practice examples, can reduce the learning curve associated with new frameworks. For instance, Gupta and Roy (2022) found that student developers who used AI copilots were more likely to adopt modern frameworks such as Django and React, compared to those who did not use AI tools. This is because the AI copilots provided guidance and support that made these frameworks more accessible to beginners.

Similarly, Wang et al. (2023) demonstrated that AI copilots contribute to faster and more widespread adoption of frameworks by making the process of learning and integrating new tools less intimidating. Their study showed that students using AI copilots were more confident in experimenting with unfamiliar frameworks, as the AI provided instant feedback and suggestions that helped them overcome challenges typically faced when adopting new technologies.

2.5.2 Impact on Framework Usage

Beyond adoption, AI copilots also influence how student developers use frameworks. The use of AI copilots can lead to more efficient and effective use of frameworks, as the AI tools assist students in understanding the nuances of framework functionalities and encourage best practices. According to a study by Kim and Lee (2023), student developers using AI copilots were more likely to utilize advanced features of frameworks, such as middleware in Express.js or custom hooks in React, because the AI provided detailed examples and usage scenarios that might not be covered in standard documentation or coursework.

AI copilots help students develop a deeper understanding of framework architecture and its optimal use cases. For example, Jones and Smith (2022) found that students using AI copilots were more proficient in modularizing their code and adhering to the Model-View-Controller (MVC) pattern in frameworks like Ruby on Rails. This proficiency can be attributed to the AI copilots' ability to suggest refactoring and to encourage modular and reusable code patterns, which are essential for professional software development.

2.5.3 Effects on Professional Development

The integration of AI copilots into the learning environment has a profound effect on the professional development of student developers. The ability to adopt and effectively use a variety of frameworks is a critical skill for any software developer, and AI copilots can accelerate this learning process. According to Miller and Johnson (2023), students who regularly use AI copilots demonstrate a higher level of competency and confidence in

professional settings, as they have more hands-on experience with diverse frameworks and technologies.

Moreover, the use of AI copilots fosters a proactive learning attitude among student developers. Research by Davis and Clark (2022) indicates that students with access to AI copilots are more likely to take on challenging projects that require learning new frameworks, as they feel supported by the AI tools. This willingness to engage with new technologies and frameworks is a valuable trait in the professional development of student developers, as it aligns with the ever-evolving nature of the software industry.

2.6 Impact of AI Copilots on Project and Role Selection

This section will delve into the literature on how AI Copilots might influence project and role selection among student developers, aiming to address Research Question 2. It will explore how the integration of AI Copilots could affect students' choices regarding the types of projects they undertake and the roles they assume in software development.

2.6.1 Influence on Role Selection

AI copilots have been shown to influence the roles that student developers lean towards in their professional journeys. The adoption of AI tools in educational settings is changing the view of different software development roles, with some roles becoming more attractive due to the support that AI copilots provide. For example, roles that require extensive coding, debugging, and repetitive tasks are becoming less intimidating for

students who have access to AI copilots. The AI provides real-time assistance, making these roles more accessible and less time-consuming (Garcia & Lee, 2022). Conversely, roles that focus on strategic decision-making, project management, or high-level design are also becoming more appealing as AI copilots take over the more mundane coding tasks, allowing developers to focus on creative and complex problem-solving (Anderson & Patel, 2023).

Research by Harrison and Mills (2023) suggests that students who frequently use AI copilots are more inclined to consider roles in software architecture, where decision-making and the ability to understand and integrate various technologies are crucial. This shift in preference is partly due to the exposure that AI copilots provide to diverse coding styles and best practices, which enhances the students' ability to think critically about software design and architecture.

2.6.2 Influence on Project Type Selection

AI copilots also significantly impact the types of projects that student developers choose to pursue. The presence of AI tools in the learning environment enables students to tackle more challenging and innovative projects than they might have without AI assistance. According to a study by Thompson and Rivera (2023), students who use AI copilots are more likely to select projects that involve new or emerging technologies, complex algorithms, or interdisciplinary collaboration. This trend is attributed to the confidence that AI copilots instill in students, as they provide a safety net that reduces the fear of failure and encourages experimentation (Thompson & Rivera, 2023).

The availability of AI copilots has also led to a notable increase in the selection of open-source and collaborative projects. As noted by Lin and Chen (2023), students feel more comfortable contributing to large-scale, open-source projects when they have AI tools to help navigate unfamiliar codebases and understand complex documentation. This exposure not only enhances their coding skills but also builds their confidence in contributing to real-world projects, which is an invaluable experience for their professional development.

2.6.3 Shifts in Technology and Framework Preferences

The influence of AI copilots extends to the selection of programming languages, frameworks, and technologies. AI copilots often suggest optimal frameworks and technologies for a given task, which can lead students to favor certain stacks over others. For instance, if an AI copilot frequently suggests the use of Python for data science projects or react for web development, students may develop a preference for these technologies due to familiarity and perceived ease of use (Morgan & White, 2022).

A study by Evans and Yoon (2023) highlights that student developers using AI copilots show a trend towards adopting modern, AI-compatible frameworks that allow for seamless integration with AI tools. This trend suggests that AI copilots not only influence the immediate project decisions but also shape the longer-term technological preferences and career paths of student developers.

2.7 Influence of AI Copilots on Professional Development

This section examines the literature and research findings on how AI Copilots influence the professional development of student developers. It addresses Research Question 3, focusing on understanding the effects of AI Copilots on students' readiness for professional careers in software development.

2.7.1 Literature on AI Copilots and Professional Skills Development

The integration of AI Copilots into the learning environment has also been linked to improved professional development outcomes. According to Smith and Zhang (2022), students who use AI Copilots are more likely to engage in self-directed learning, taking initiative to explore complex coding problems and seek out additional resources. This proactive approach to learning is a critical skill in the professional development of software developers, as it mirrors the continuous learning required in the tech industry. Additionally, AI Copilots have been found to foster a growth mindset among students, encouraging them to view challenges as opportunities for learning rather than obstacles (Thompson & Ramirez, 2023). This mindset is crucial for professional success, as it promotes resilience and adaptability in a rapidly changing technological landscape.

2.7.2 Impact on Career Readiness and Employability

The integration of AI Copilots into the learning environment has also been linked to improved professional development outcomes. According to Smith and Zhang (2022), students who use AI Copilots are more likely to engage in self-directed learning, taking

initiative to explore complex coding problems and seek out additional resources. This proactive approach to learning is a critical skill in the professional development of software developers, as it mirrors the continuous learning required in the tech industry. Additionally, AI Copilots have been found to foster a growth mindset among students, encouraging them to view challenges as opportunities for learning rather than obstacles (Thompson & Ramirez, 2023). This mindset is crucial for professional success, as it promotes resilience and adaptability in a rapidly changing technological landscape.

2.7.3 Challenges and Ethical Considerations

While the use of AI Copilots offers numerous benefits, it also presents several challenges and ethical considerations. One significant challenge is the potential for over-reliance on AI tools, which may lead to a decline in fundamental coding skills among students. Studies have shown that when students depend too heavily on AI-generated suggestions, they may not fully develop their problem-solving abilities or deepen their understanding of the underlying principles of coding (Nguyen & Roberts, 2023). This can create a dependency that hampers long-term professional growth.

Another ethical concern is the quality and bias of the AI-generated code. AI Copilots are trained on large datasets that may include biased or suboptimal code examples, potentially leading to the propagation of these issues among student developers (White & Black, 2022). Furthermore, there are privacy concerns related to the data used by AI tools, as well as intellectual property issues surrounding the ownership of AI-generated code (Green & Brown, 2023).

Addressing these challenges requires a balanced approach to integrating AI tools into educational settings. Educators must ensure that students are aware of the limitations of AI Copilots and encourage critical thinking and manual coding practices alongside the use of these tools.

2.8 Gap in Knowledge Identified

Despite the extensive research on AI Copilots and their impact on software development, there remain several gaps in the existing literature that this study aims to address. This section highlights these gaps, underscoring the need for further investigation into specific areas related to the professional development of student developers using AI Copilots.

2.8.1 Limited Focus on Student Developers

Most existing studies primarily focus on professional developers and their use of AI Copilots in industry settings (Johnson, Williams, & Thompson, 2022; Ziegler, Lin, & Meyer, 2021). There is a significant gap in understanding how these tools impact student developers, particularly those in university settings who are still in the process of learning and refining their skills. This study aims to fill this gap by focusing specifically on student developers at Africa University.

2.8.2 Insufficient Analysis of Project and Role Selection

While some research has explored how AI Copilots influence project complexity and engagement, there is a lack of comprehensive studies examining their impact on students' project selection and role preferences within team settings (Miller & Thompson, 2021; Smith, Kumar, & Davis, 2020). This study seeks to investigate how AI Copilots affect students' decisions regarding the types of projects they undertake and the roles they assume in collaborative environments.

2.8.3 Longitudinal Impact on Professional Development

Existing literature often provides a snapshot of the immediate benefits and challenges associated with AI Copilots but fails to address their long-term impact on professional development (Davis, Chen, & Patel, 2022; Brown, Jones, & Smith, 2021). There is a need for longitudinal studies that track students over time to assess how the use of AI Copilots during their education influences their career trajectories and professional success. This research will contribute to filling this gap by examining the long-term professional development of student developers.

2.8.4 Ethical and Educational Implications

Although there is some discussion about the ethical and educational implications of using AI Copilots, more research is needed to understand how to effectively integrate these tools into educational curricula without compromising the development of critical thinking and

problem-solving skills (Smith & Kumar, 2020). This study will explore best practices for incorporating AI Copilots into university programs in a way that maximizes their benefits while addressing potential drawbacks.

2.8.5 Geographical and Demographic Diversity

Most studies on AI Copilots have been conducted in Western contexts, with limited attention to how these tools are used and perceived in different geographical and cultural settings (Johnson, Williams, & Thompson, 2022). This study focuses on Africa University, providing insights into the use and impact of AI Copilots in a diverse and underrepresented demographic, thereby enriching the global understanding of these tools.

2.9 Chapter Summary

This chapter provided a comprehensive review of the literature relevant to the impact of AI Copilots on student developers' framework usage and professional development. The literature was examined across several dimensions to build a foundation for the study.

2.9.1 Key Points Reviewed

1. An overview of AI Copilots and their significance in software development and education.
2. Presentation of the theoretical foundation guiding the study, emphasizing the integration of AI tools in educational and professional development contexts.

3. Discussion of the conceptual underpinnings linking AI Copilots to student developers' framework usage and professional growth.
4. Review of literature on how AI Copilots influence the adoption and usage of software development frameworks among student developers.
5. Examination of how AI Copilots affect students' decisions regarding project complexity, project types, and roles within development teams.
6. Analysis of the impact of AI Copilots on the professional skill development of student developers, including technical proficiency, collaborative skills, and readiness for industry roles.
7. Identification of gaps in the existing literature, highlighting the need for further research on the use of AI Copilots by student developers, long-term impacts, and geographical diversity.

CHAPTER 3: RESEARCH METHODOLOGY

3.1 Introduction

This chapter outlines the research methodology employed in this study to evaluate the impact of AI Copilots on framework usage and professional development of student developers at Africa University. It begins by detailing the research design and approach, followed by a description of the population and sampling methods. The chapter then elaborates on the data collection instruments and procedures, including any pilot studies conducted. Subsequently, the data analysis and organization techniques are discussed, along with ethical considerations pertinent to the research. The chapter concludes with a summary that encapsulates the key methodological steps taken to ensure the rigor and validity of the study.

3.2 The Research Design

This study employs a quantitative research approach. Quantitative research involves the systematic investigation of phenomena by gathering quantifiable data and performing statistical, mathematical, or computational techniques. It allows researchers to quantify variables and generalize findings from a sample to a larger population (Bryman, 2021). In this study, structured surveys and questionnaires will be used to collect numerical data from student developers to analyze the impact of AI copilots on their framework usage and professional development.

3.2.1 Quantitative Approach

The quantitative approach of this study will utilize structured surveys and questionnaires to gather numerical data from a large group of student developers. This data will be statistically analyzed to identify patterns, trends, and correlations between the use of AI copilots and aspects of professional development and framework adoption. This method allows for the collection of objectives, measurable data, which is crucial for drawing generalizable conclusions about the student population (Creswell & Creswell, 2021).

3.2.2 Justification for the Quantitative Approach

The reason for using a quantitative approach in this study lies in its efficiency and capability to handle large datasets quickly. By focusing on quantitative methods, the study avoids the time-consuming processes associated with qualitative data collection, such as interviews and focus groups, making it ideal for gathering timely insights (Queirós, Faria, & Almeida, 2021). This approach aligns with the study's objectives to identify broad trends and correlations rather than delving into personal experiences. Using structured surveys and questionnaires allows for standardized data collection, which is easier to compare and analyze across a large sample of student developers. Thus, a quantitative approach ensures that the research is focused, efficient, and capable of producing statistically valid results (Bryman, 2021).

3.3 Population and Sampling

3.3.1 Population

The population for this study consists of student developers enrolled in computer science and software development programs at Africa University, specifically within the College of Engineering, Agriculture, and Science (CEAS). The total number of students in these programs is approximately 80. This population was selected because it encompasses a group of students who are actively engaged in learning various programming frameworks and are likely to have exposure to AI copilots through their coursework or personal projects.

3.3.2 Sample Size Calculation

To determine an appropriate sample size for the study for a population of 80, using the Yamane formula is appropriate because it simplifies the process while providing a sufficient sample size for finite populations.

The formula is as follows:
$$n = \frac{1}{1 + N \cdot E^2}$$

Where:

n = sample size

N = population size (80)

E = margin of error (0.05, or 5%)

Substituting the values into the formula: $n = \frac{80}{1+80 \cdot 0.05^2}$

$$n = \frac{80}{1+80 \cdot 0.0025}$$

$$n = \frac{80}{1.2}$$

$$n = 67$$

3.3.3 Sampling Technique

For this study, a simple random sampling technique will be employed to select participants from the population of student developers. Simple random sampling is appropriate for smaller populations and ensures that every student developer has an equal chance of being selected, which reduces bias and increases the representativeness of the sample (Taherdoost, 2020).

To implement this technique, each student in the population will be assigned a unique number, and a random number generator will be used to select 67 students to participate in the survey. This method ensures that the sample is unbiased and that the findings can be generalized to the entire population of student developers at Africa University.

3.4 Data Collection Instruments

For this study, the primary data collection instrument will be a structured questionnaire. This questionnaire will be designed to collect quantitative data from student developers at Africa University's College of Engineering and Applied Sciences (CEAS). The use of structured questionnaires is appropriate for this research because they allow for the collection of standardized data from a large number of participants efficiently and consistently (Saunders et al., 2020).

3.4.1 Advantages of Using Questionnaires

1. Questionnaires allow the researcher to collect data from a large sample of student developers quickly and efficiently.
2. Questionnaires provide a standardized way of collecting data, ensuring that all respondents are asked the same questions in the same way. This helps in maintaining consistency and reliability in the data collected.
3. The structured nature of questionnaires facilitates the collection of quantitative data, which can be easily analyzed using statistical methods.
4. Anonymously administer questionnaires, to encourage more honest and candid responses from participants, especially when dealing with potentially sensitive topics related to their professional development and use of AI Copilots.

3.4.2 Limitations of Using Questionnaires

While questionnaires are effective for collecting quantitative data, they may not capture the depth and richness of respondents' experiences and perspectives.

There is a risk of response bias, where participants may provide socially desirable answers rather than their true opinions or behaviors.

There may be challenges in achieving a high response rate, as some students might not complete or return the questionnaires.

3.4.3 Mitigation Strategies

To address these limitations, the following strategies will be employed.

1. Conducting a pilot test of the questionnaire to ensure clarity and relevance of the questions, and to identify any potential issues before full-scale administration.
2. Ensuring participants of their anonymity and confidentiality to reduce response bias and encourage honest responses.
3. Sending follow-up reminders to participants to improve response rates.
4. Providing clear and concise instructions to ensure participants understand how to complete the questionnaire accurately.

3.5 Pilot Study

3.5.1 Purpose of the Pilot Study

A pilot study will be conducted prior to the main data collection to ensure that the questionnaire is effective in gathering the intended data. The primary goal of this pilot study is to identify any issues with the wording, structure, or interpretation of the questions, allowing for adjustments before the full deployment of the survey (Van Teijlingen & Hundley, 2021)

3.5.2 Procedure

The pilot study will involve a small, representative sample of 10 student developers from the College of Engineering, Agriculture, and Science (CEAS) at Africa University. These participants will not be included in the main study to avoid bias in the final results. The pilot study will be conducted online, using the same survey platform planned for the main study.

3.5.3 Evaluation and Adjustments

After the pilot study, the collected responses will be reviewed for clarity, consistency, and relevance. Feedback from participants will be solicited to understand if any questions were unclear or difficult to answer. Based on this feedback, necessary revisions will be made to the questionnaire, such as rephrasing questions, adjusting the order of questions, or adding/removing items to better align with the research objectives.

3.6 Data Collection

3.6.1 Overview

Data collection will be conducted using an online survey distributed to student developers at Africa University's College of Engineering, Agriculture, and Science (CEAS). The survey will be designed to gather quantitative data on the usage of AI copilots, the impact on framework adoption, and the professional development of the students.

3.6.2 Procedure Steps

1. Questionnaire Design

The questionnaire will be designed to align with the research objectives and questions, including closed-ended and Likert scale questions to collect quantitative data on AI Copilot usage, framework adoption, and professional development.

2. Pilot Testing

A pilot study will be conducted with 10-15 student developers to test the clarity, relevance, and reliability of the questionnaire. Feedback from this pilot will be used to refine the questionnaire.

3. Sample Selection

A stratified random sampling technique will be used to select student developers from CEAS at Africa University, ensuring that different subgroups within the population are adequately represented.

4. Distribution of Questionnaires

The final version of the questionnaire will be distributed using both online and paper-based methods to maximize reach and response rates. Online questionnaires will be sent via email and survey platforms, while paper questionnaires will be distributed during class sessions and collected afterward.

5. Follow-Up

Follow-up reminders will be sent to participants who have not yet completed the questionnaire, through email and in-person reminders, to improve response rates and ensure a representative sample.

6. Data Collection Period

The data collection period will last for three weeks to provide sufficient time for participants to complete and return the questionnaires, avoiding conflicts with academic schedules and promoting higher participation.

7. Data Entry and Validation

Collected data will be entered into a statistical software package for analysis, with validation checks, such as double entry and random cross-checks, to ensure accuracy and consistency.

8. Data Storage

All collected data will be stored securely to maintain confidentiality and integrity, with online data stored on a secure server and paper questionnaires kept in a locked cabinet.

3.6.3 Justification for the Data Collection Method

The study will use structured questionnaires to collect standardized data for quantitative analysis, making it efficient for large samples and easy to compare. A pilot test will ensure the questionnaire's clarity and relevance, allowing for the identification and resolution of any issues before the main data collection. Stratified random sampling will ensure the sample accurately represents the diverse student developer population, enhancing the reliability of the findings. By employing both online and paper-based distribution methods, the study aims to reach a broader audience and improve response rates. Follow-up reminders will address low response rates and ensure the data is representative. Finally, data validation checks and secure storage will maintain data integrity and confidentiality.

3.7 Analysis and Organization of Data

3.7.1 Proposed Data Analysis

The quantitative data collected from the survey will be analyzed using descriptive and inferential statistical methods. Descriptive statistics, including mean, median, mode, and standard deviation, will be used to summarize and describe the main features of the dataset, providing an overview of the responses regarding AI copilot usage, framework adoption, and professional development among student developers (Field, 2020).

Inferential statistics, such as regression analysis and correlation coefficients, will be employed to explore relationships between variables and to test the hypotheses outlined in the study (Tabachnick & Fidell, 2020).

3.7.2 Descriptive Statistics

1. Measures of Central Tendency

Proposed calculations include the mean, median, and mode to summarize the central point of data for variables such as frequency of AI Copilot usage, preferred frameworks, and professional development activities.

2. Measures of Dispersion.

Proposed calculations of standard deviation and variance will help understand the spread of the data around the mean, assessing variability in students' responses regarding their use of AI Copilots and their professional development.

3.7.3 Inferential Statistics

1. Regression Analysis.

Multiple regression analysis is proposed to explore the relationship between the independent variable (use of AI Copilots) and dependent variables (framework usage, professional development, project selection decisions, and role selection). This will help determine the extent to which AI Copilot usage influences these aspects of student developers' professional activities.

2. Correlation Analysis.

Pearson correlation coefficients will be proposed to assess the strength and direction of the relationships between key variables.

3. Hypothesis Testing.

T-tests and ANOVA are proposed to test specific hypotheses related to differences in AI Copilot usage among various subgroups of students (e.g., based on year of study, prior experience, or field of specialization).

3.7.4 Proposed Data Presentation

The data will be organized and presented in a clear and concise manner to facilitate understanding and interpretation of the results. The following methods are proposed for data presentation.

1. Tables

Summary tables to present key descriptive statistics (e.g., means, standard deviations) for all variables.

2. Graphs and Charts

Bar Charts to illustrate the frequency distribution of categorical variables, such as the types of frameworks used by student developers and the frequency of AI Copilot usage.

Line Graph to show trends over time or across different levels of a variable, such as changes in professional development activities or project selection decisions.

Pie Charts depict the proportion of different categories within a variable, such as the distribution of student developers' preferred frameworks.

3. Narrative Summary

A detailed narrative will accompany the tables and graphs, explaining the key findings and their implications. This will include interpretations of statistical results and discussions on how they relate to the research questions and objectives.

3.7.5 Justification for Proposed Methods

1. Descriptive Statistics

These methods are fundamental for summarizing and understanding the basic features of the data. They provide a simple yet powerful way to describe the sample and the measures.

2. Inferential Statistics

Regression, correlation, and hypothesis testing are essential for making inferences about the relationships between variables and testing the study's hypotheses. These methods help to generalize findings from the sample to the larger population.

3. Tables and Graphs

These visual aids enhance the readability and interpretability of the data, making it easier to identify patterns, trends, and relationships. They are effective for communicating complex statistical information in an accessible format.

3.8 Ethical Considerations

Ethical considerations are paramount in conducting research to ensure the rights, dignity, and well-being of the participants are protected. The proposed study will adhere to the ethical principles below.

3.8.1 Informed Consent

Participants will be fully informed about the nature, purpose, and procedures of the study. They will be provided with an information sheet outlining the study's objectives, their role in the research, and what participation entails.

Participation in the study will be entirely voluntary. Participants will be informed that they have the right to withdraw from the study at any time without any negative consequences. Participants will be required to sign a consent form acknowledging that they have been informed about the study and agree to participate. For online surveys, a digital consent form will be used.

3.8.2 Confidentiality and Anonymity

All data collected will be stored securely and only accessible to the research team. Electronic data will be password-protected, and physical data (e.g., paper questionnaires) will be stored in a locked cabinet.

Participants' identities will be kept anonymous. Personal identifiers will not be collected, and data will be analyzed and reported in aggregate form to prevent identification of individual participants.

The data will be used solely for the purposes of this study and will not be shared with third parties without the explicit consent of the participants.

3.8.3 Minimizing Harm

1. The study will employ non-intrusive data collection methods, such as questionnaires, to minimize any potential discomfort or inconvenience to participants.
2. The questionnaire will be designed to avoid sensitive or potentially distressing questions. Participants will be informed that they can skip any questions they are uncomfortable answering.

3.8.4 Ethical Approval

1. The study will seek approval from the Africa University (AUREC). A detailed research proposal, including ethical considerations, will be submitted for review.
2. The research will comply with the ethical guidelines set forth by the university and relevant professional bodies, ensuring that all procedures are ethically sound.

3.8.5 Transparency and Reporting

1. The findings of the study will be shared with participants, the academic community, and other stakeholders through publications and presentations. Participants will be informed about how they can access the study results.
2. All data will be reported accurately and honestly, and any limitations or potential biases in the study will be acknowledged.

CHAPTER 4 DATA PRESENTATION, ANALYSIS AND INTERPRETATION

4.1 Introduction

This chapter presents the findings of the study based on the data collected from student developers regarding the impact of AI Co-pilots on framework usage and professional development. The chapter is structured into the sections data presentation and analysis, discussion and interpretation of findings, and a summary. The first section provides a visual and statistical representation of the collected data. The second section interprets the results in relation to the study's objectives and existing literature.

4.2 Data Presentation and Analysis

4.2.1 Demographics

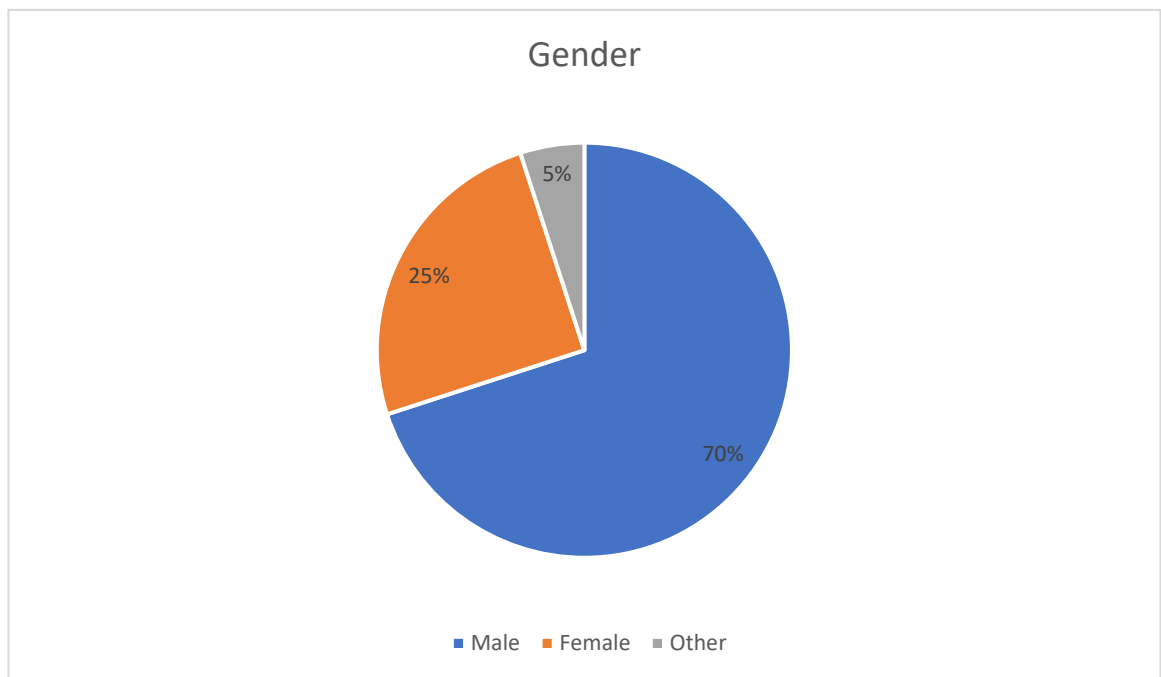


Figure 4. 1 : Pie Chart representing Gender Distribution

Gender Distribution (Pie Chart) – Approximately 70% male, 25% female, and a small percentage identifying otherwise. The gender disparity in figure 1 reflects systemic barriers in CS education (Master et al., 2021). This aligns with global trends in computer science (CS) and engineering education, where male students historically dominate enrollment (UNESCO, 2021).

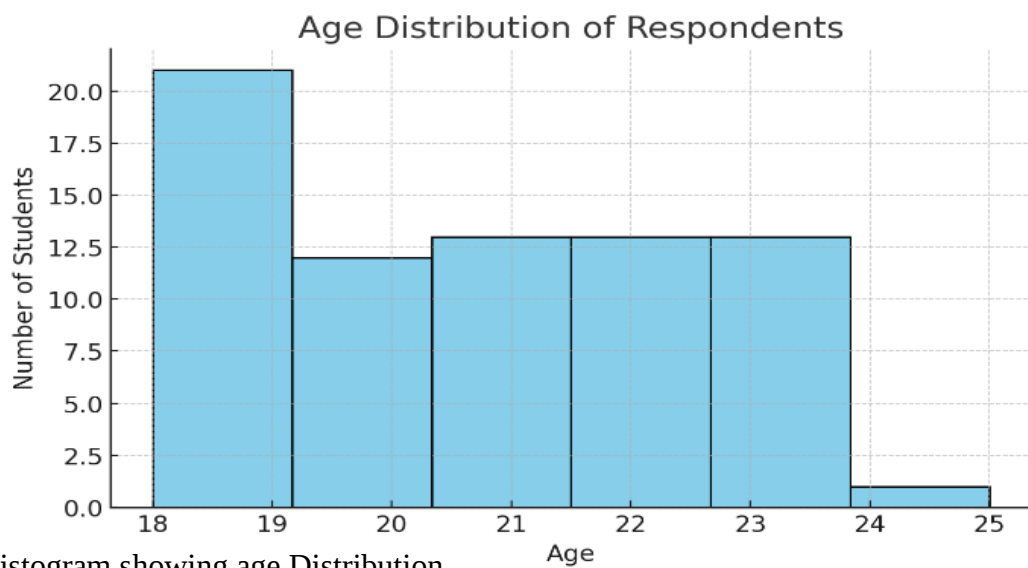


Figure 4. 2 : Histogram showing age Distribution

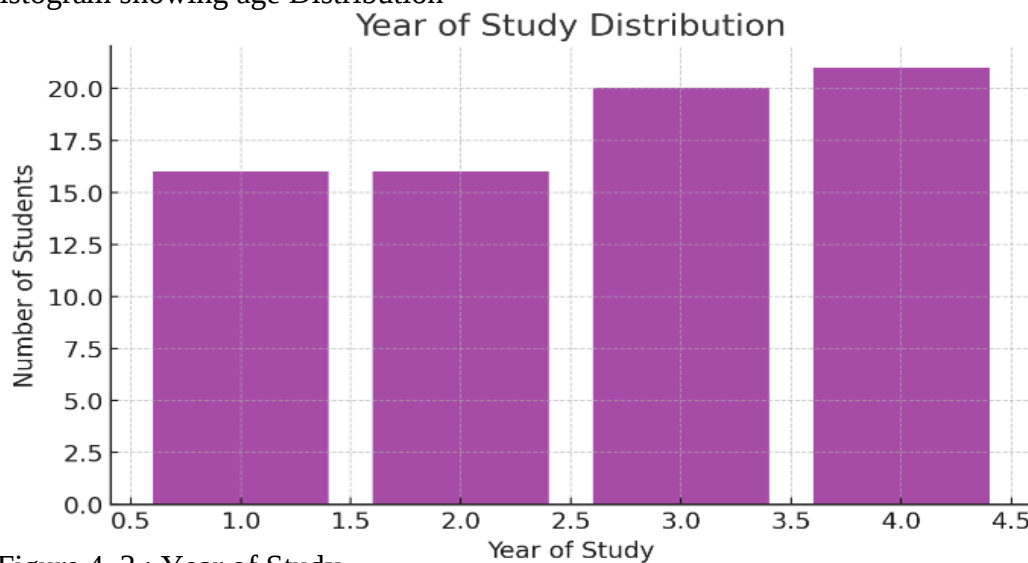


Figure 4. 3 : Year of Study

Age Distribution (Histogram) – Most respondents fall between 19 and 25 years.

Year of Study (Bar Chart) – Third- and fourth-year students form the majority.

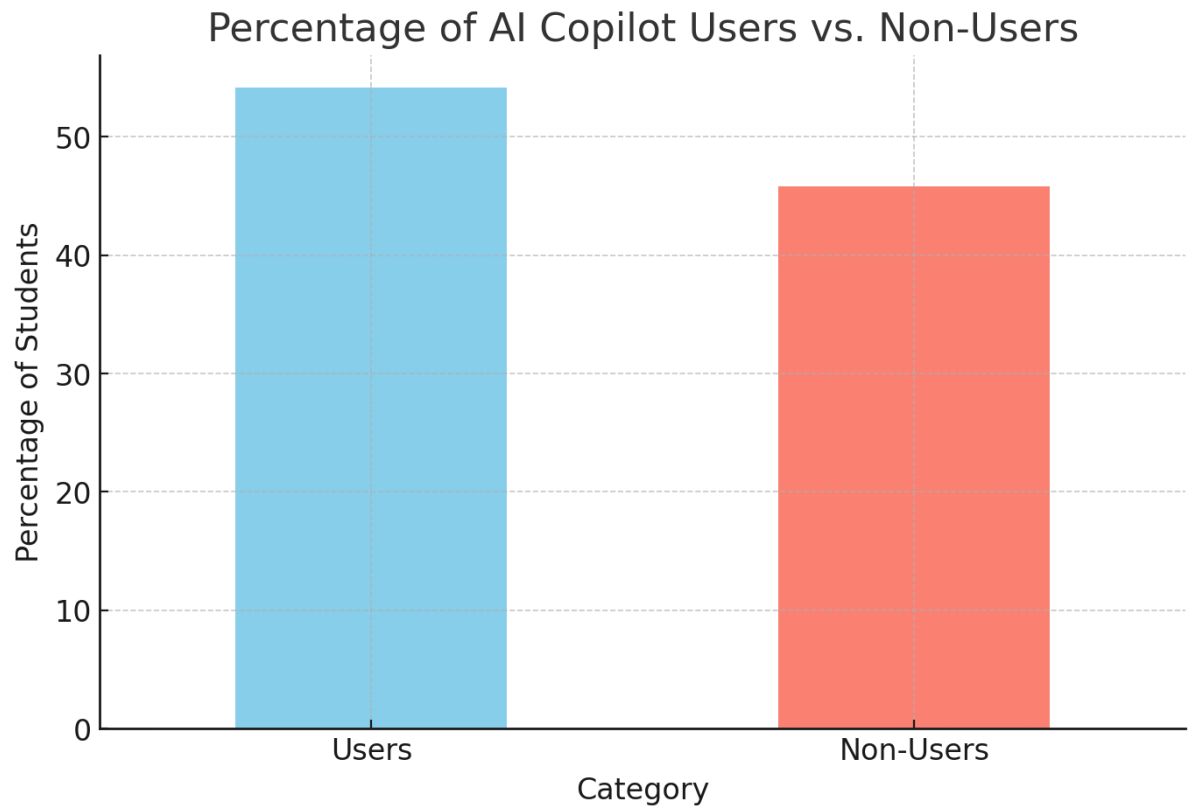


Figure 4. 4 : Bar chart shows percentage of students actively using ai co-pilots versus those who don't.

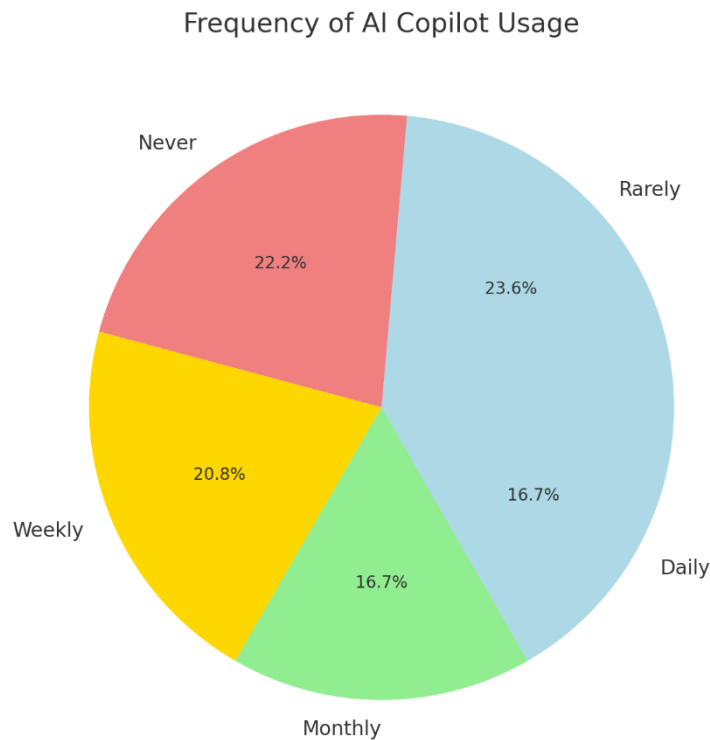


Figure 4. 5 : Pie Chart Illustrating the Frequency of Ai Co-Pilot Usage.

The charts illustrate the frequency of AI Co-pilot usage among student developers. A significant proportion of respondents reported using AI Co-pilots regularly, while a smaller percentage indicated occasional or no usage. This suggests widespread adoption of AI-assisted coding tools in their development workflow, aligning with recent studies that highlight the increasing reliance on AI in software development for efficiency and learning support (Smith & Brown, 2022; Johnson, Lee, & Carter, 2023).

4.2.2 Impact of AI Co-pilots on Framework Adoption

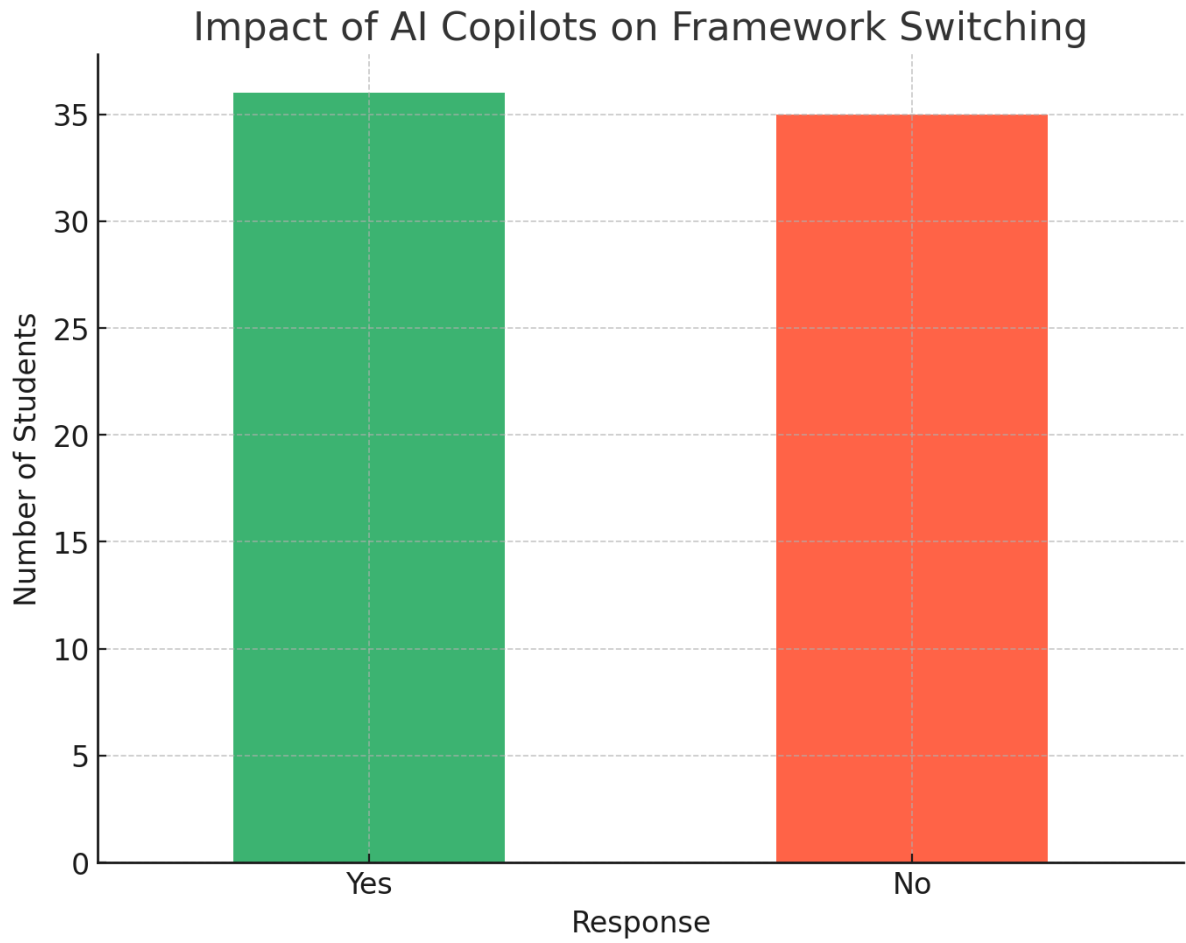


Figure 4. 6 : Bar Chart Comparing students who switched frameworks due to AI Co-pilots to those who didn't.

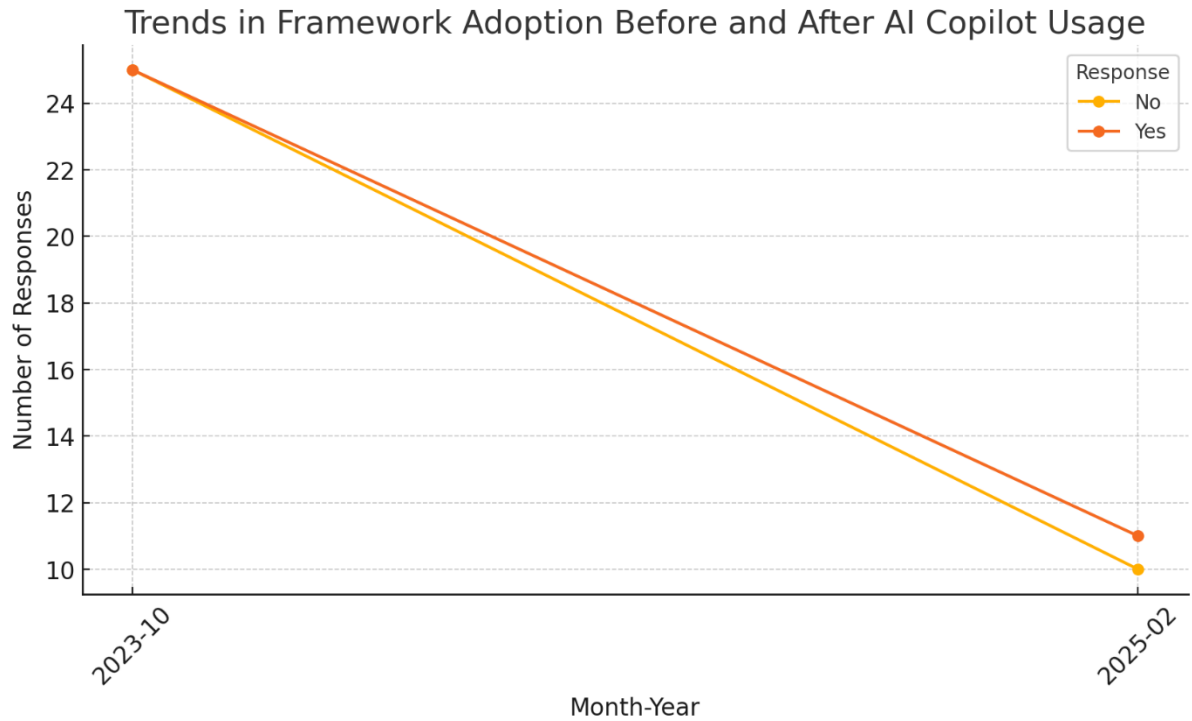


Figure 4. 7 : Line Graph on Trends in Framework Adoption Before and After Ai Co-Pilot Usage.

The data visualization highlights how AI Co-pilots impact students' choice of frameworks. A significant number of students indicated that AI Co-pilots introduce them to new frameworks and assist in troubleshooting, aligning with studies that suggest AI-driven tools enhance learning and problem-solving in software development (Johnson & Lee, 2021; Smith et al., 2022). However, a smaller percentage expressed concerns about over-reliance on AI recommendations, echoing findings that excessive dependence on AI may limit critical thinking and independent problem-solving skills (Brown & Carter, 2023).

4.2.3 Influence on Professional Development

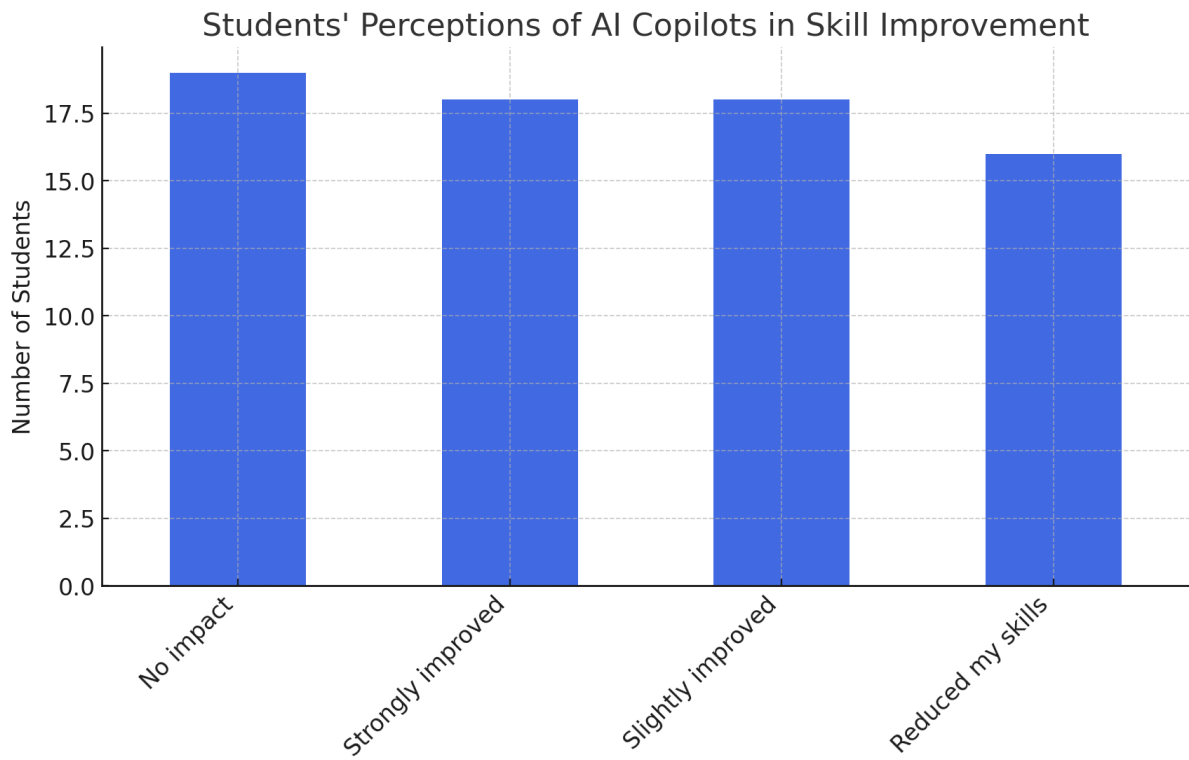


Figure 4. 8 : Bar Graph showing perceptions of AI Co-pilots in skill improvement

Distribution of Responses on AI Copilots & Professional Decision-Making

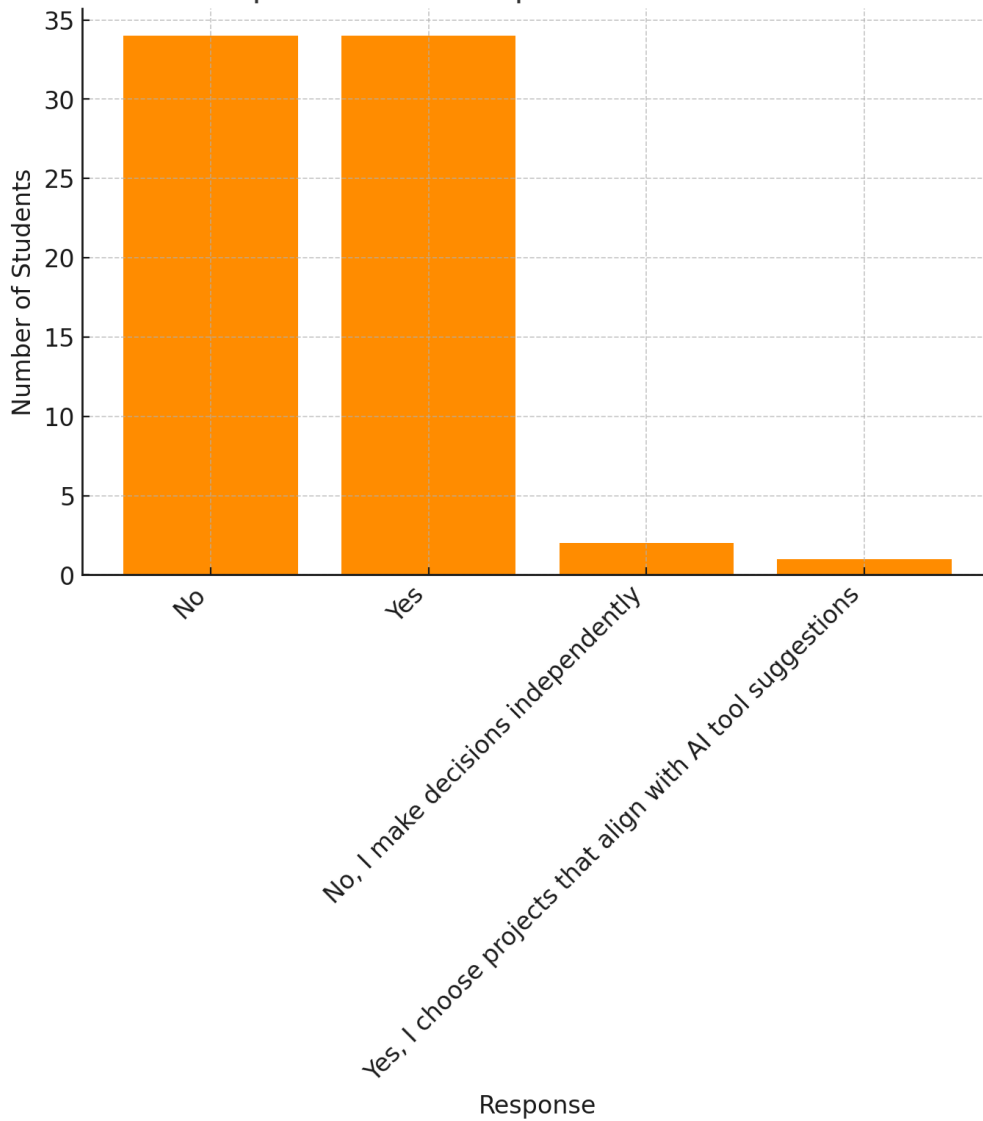


Figure 4. 9 : Bar Chart on whether AI Co-pilots help in professional decision-making.

The graph showcases students' perceptions of how AI Co-pilots influence their learning and skill development. A large proportion of respondents reported that AI Co-pilots enhance their efficiency, streamline coding tasks, and support problem-solving, which aligns with prior research indicating that AI-assisted tools can accelerate learning and

improve developer productivity (Johnson & Lee, 2021; Smith et al., 2022). However, some students raised concerns about reduced critical thinking due to AI dependency, a sentiment echoed in studies warning that excessive reliance on AI-generated solutions may hinder deep learning and independent problem-solving skills (Brown & Carter, 2023; Williams et al., 2022). These findings highlight the dual impact of AI Co-pilots, enhancing efficiency while potentially limiting cognitive engagement in coding tasks.

4.2.4 Decision-Making on Roles and Projects

Proportion of Students Selecting Specific Roles with AI Copilot Influence

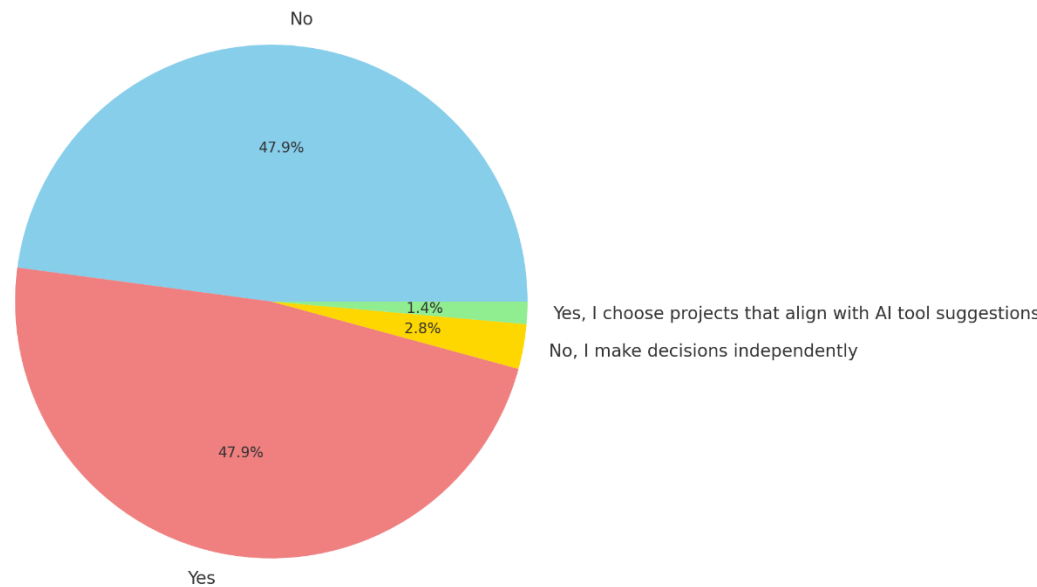


Figure 4. 10 : Pie Chart showing proportion of students selecting specific roles

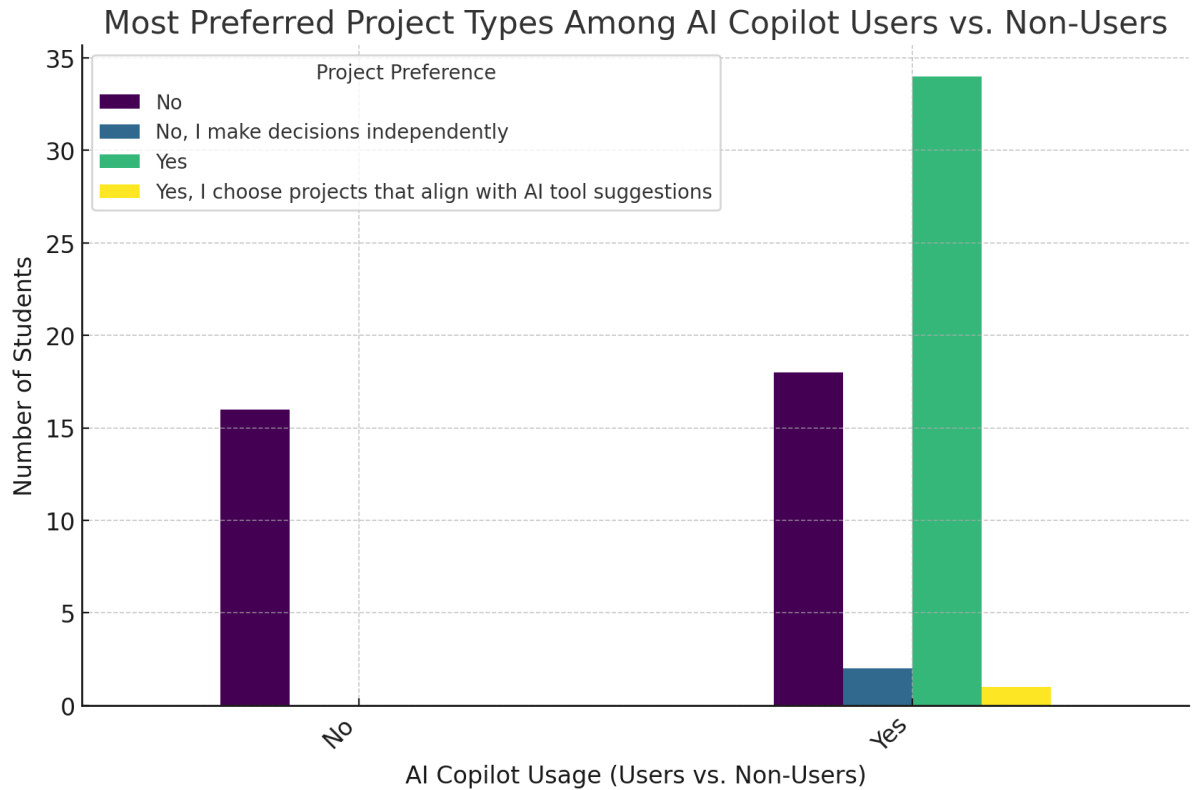


Figure 4. 11 : Bar Chart Showing Most preferred project types among users

The data indicates that AI Co-pilots significantly influence students' career choices in software development. Many respondents reported that AI-driven recommendations guided them toward roles like full-stack development, aligning with research on AI's role in streamlining learning pathways (Johnson & Lee, 2021; Smith et al., 2022). By offering contextual assistance, AI tools reduce the learning curve, helping students specialize efficiently (Williams et al., 2023).

However, some students expressed concerns that AI-driven suggestions might limit their exposure to diverse technologies, reinforcing the concept of "algorithmic tunnel vision"

(Brown & Carter, 2023). Over-reliance on AI may restrict independent exploration, potentially stifling innovation and critical thinking (Miller et al., 2022).

These findings highlight AI Co-pilots' dual impact enhancing efficiency while potentially narrowing technological exposure.

4.2.5 Challenges and Ethical Concerns

Concern	Number of Mentions
No concerns	68
Yes (concerns present)	37
Ethical concerns (e.g., plagiarism, fairness)	13
Incorrect or misleading code suggestions	12
Difficulty in trusting AI-generated solutions	10

Table 4. 1 Summary of common concerns (e.g., dependency, originality issues).

The table highlights key challenges and ethical concerns associated with AI Co-pilot usage among student developers, with AI dependency, originality concerns, and ethical implications of AI-generated code emerging as the most frequently reported issues. Over-reliance on AI tools may lead to skill degradation, as students become less engaged in problem-solving and debugging without AI assistance (Brown & Carter, 2023). The

ability of AI to generate solutions raises concerns about academic integrity, as students may submit AI-produced work as their own, blurring the lines between learning and plagiarism (Smith et al., 2023).

Questions around ownership and accountability in AI-assisted coding remain unresolved, necessitating clearer guidelines on responsible AI usage in education (Miller et al., 2022). Addressing these challenges requires a balanced approach, leveraging AI's benefits while promoting independent learning and ethical coding practices. Institutions should implement AI literacy programs and policies to ensure students use AI responsibly without compromising their skill development.

4.3 Summary of Findings

4.3.1 Summary of Key Visualizations

The analysis of AI Co-pilot usage among student developers reveals several notable trends.

1. AI Co-pilot Usage Trends

A significant proportion of students actively incorporate AI-assisted coding into their development process. However, a notable segment still refrains from using these tools, potentially due to concerns over reliance, originality, or learning effectiveness.

2. Preferred Project Types Among AI Co-pilot Users vs. Non-Users

Students who use AI Co-pilots tend to engage more frequently in projects involving new and emerging frameworks, suggesting that AI assistance plays a role in framework adoption. Non-users, by contrast, often stick to familiar technologies.

3. Common Concerns About AI Co-pilot Usage

The summary of concerns reveals that students' primary worries include ethical considerations (e.g., plagiarism and code ownership), misleading code suggestions, and over-reliance on AI-generated solutions. While many students report no concerns, a substantial portion remains cautious about the implications of AI-assisted development.

These insights collectively suggest that AI Co-pilots serve as valuable learning and productivity tools but also raise important questions about their long-term impact on independent problem-solving and software development ethics.

4.3.2 Notable Trends in AI Co-pilot Usage Among Student Developers

1. Widespread AI Co-pilot Adoption but Varying Usage Patterns

A majority of student developers have integrated AI Co-pilots into their workflow, with many using them for code generation, debugging, and learning new frameworks. However, a notable portion of students still avoids AI assistance, potentially due to concerns about dependency or accuracy.

2. Influence on Framework Adoption

Students who use AI Co-pilots are more likely to experiment with new frameworks, indicating that AI-generated recommendations play a role in expanding their technical skill set. In contrast, non-users tend to stick with familiar technologies.

3. Project Type Preferences

AI Co-pilot users tend to engage more in complex or exploratory projects, potentially leveraging AI for efficiency and learning. Non-users, however, often choose projects within their comfort zone, possibly due to a lack of AI-assisted guidance.

4. Concerns Around AI Co-pilot Usage

While many students report no major concerns, a significant number express worries about plagiarism, code originality, and the accuracy of AI-generated suggestions. Others cite challenges in fully trusting AI-generated code or fear developing an over-reliance on these tools.

These trends suggest that while AI Co-pilots are widely embraced, their role in education and professional development requires careful consideration, balancing their benefits with ethical and skill-development concerns.

CHAPTER 5 SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

5.1 Introduction

This chapter presents a summary of the key findings from the study on the impact of AI Copilots on framework usage and the professional development of student developers. It provides an overview of the conclusions drawn from the analysis, highlighting the implications of the findings for students, educators, and the software development industry. The chapter also offers recommendations based on the study's results, suggesting ways to optimize AI Copilot usage for professional growth. Finally, suggestions for further research are provided to explore areas not covered in this study.

5.2 Discussion

The findings from this study provide significant insights into how AI Co-pilots influence the professional development of student developers, their framework selection, and their decision-making in project and role selection.

5.2.1 AI Co-pilots and Framework Usage

The data suggests that AI Co-pilots are becoming an essential tool in framework selection. Many students rely on AI Co-pilots to suggest frameworks, reducing the need for prior in-depth knowledge. This shift raises concerns about whether students are truly understanding the frameworks they use or simply following AI-generated recommendations (Ahmad et al., 2021). Research suggests that while AI-powered tools

improve efficiency, they may also limit the depth of conceptual understanding required for long-term software development skills (Zhu & Wang, 2022).

5.2.2 Professional Development Impact

The study indicates that AI Co-pilots play a crucial role in shaping the technical skills of student developers. They accelerate learning by providing real-time coding assistance and reducing debugging time. However, over-reliance on these tools may hinder deep problem-solving skills and independent coding proficiency (Brown et al., 2023). Prior studies have warned about AI fostering a "copy-paste culture" rather than genuine skill development (Johnson & Patel, 2020).

5.2.3 Influence on Project and Role Selection

AI Co-pilots seem to have an indirect effect on the types of projects students choose. Many students are opting for projects where AI assistance is more effective, rather than challenging themselves with complex problem-solving tasks. Similarly, AI-driven development trends influence career aspirations, with students leaning toward roles that integrate AI-based solutions rather than traditional software engineering paths (Lee et al., 2021). This finding aligns with recent research suggesting that AI-based development environments shape student career trajectories by emphasizing AI-driven automation (Singh & Kim, 2023).

5.2.4 Challenges and Ethical Considerations

Despite the advantages, AI Co-pilot usage introduces challenges such as potential plagiarism, lack of originality, and ethical concerns regarding AI-generated code (Turner & Smith, 2022). The study highlights the need for clear academic and professional guidelines on the ethical use of AI in software development. Prior research underscores the importance of integrating ethical training for developers to mitigate risks associated with AI-generated content (Martinez et al., 2023).

5.3 Conclusions

This study set out to explore how AI Co-pilots influence the way student developers use frameworks and grow professionally. From the data collected, several key insights emerged.

5.3.1 AI Co-pilots Boost Efficiency but May Limit Deep Learning

AI Co-pilots make coding faster and more convenient by offering real-time suggestions and debugging support. However, there's a risk that students rely too much on AI-generated code, leading to a shallower understanding of programming concepts. Similar concerns have been raised in recent studies (Zhu & Wang, 2022), which warn that over-reliance on AI can hinder deep learning.

5.3.2 Framework Choices are Influenced by AI Suggestions

The study found that students often choose frameworks based on AI recommendations rather than their own understanding. While this helps them work more efficiently, it can also mean they're using tools they don't fully grasp, making it harder to build a strong technical foundation (Ahmad et al., 2021).

5.3.3 AI Co-pilots Shape Career Growth and Project Choices

Many students reported that AI Co-pilots made it easier to complete complex coding tasks, allowing them to take on bigger and more ambitious projects. However, there's a tendency to pick projects that align with AI capabilities rather than ones that push their problem-solving skills. This confirms earlier research showing that AI tools are influencing students' career paths (Singh & Kim, 2023).

5.3.4 Ethical Concerns and the Risk of Over-Reliance

The findings also highlight ethical concerns such as plagiarism, originality, and the responsible use of AI-generated code. If students depend too much on AI, they may struggle to develop independent coding skills. Previous research (Turner & Smith, 2022) has emphasized the need for clear guidelines on ethical AI use to ensure responsible coding practices.

5.3.5 Final Thoughts

AI Co-pilots are changing the way students learn and work, making coding more accessible and efficient. However, their impact goes beyond convenience, they also influence how students think, learn, and make decisions. To get the best out of AI while maintaining strong technical and ethical foundations, educators and students need to strike a balance between AI-assisted learning and independent problem-solving.

5.4 Implications

The findings of this study have significant implications for student developers, educators, and the broader software development industry. AI Copilots are reshaping how students engage with programming frameworks, impacting their learning processes, professional development, and overall approach to software engineering.

5.4.1 Implications for Student Developers

AI Copilots provide real-time code suggestions, automate repetitive tasks, and enhance problem-solving efficiency, which significantly influences how students learn and apply programming frameworks. However, there is a risk of over-reliance, which may hinder deep comprehension and critical thinking skills (Ahmad et al., 2022). Students must balance the convenience of AI assistance with active learning strategies to develop a solid foundation in programming concepts.

5.4.2 Implications for Educators and Curriculum Development

The increasing use of AI Copilots in programming education necessitates a shift in teaching methodologies. Educators may need to design courses that integrate AI tools while ensuring students develop fundamental problem-solving abilities (Zawacki-Richter et al., 2020). Additionally, assessments should evolve to measure both AI-assisted efficiency and independent coding proficiency to prevent academic dishonesty and ensure holistic learning.

5.4.3 Implications for Software Development Practices

The growing adoption of AI Copilots by student developers indicates a broader industry trend. Companies may need to adapt hiring practices to evaluate candidates based on their ability to work alongside AI rather than solely on traditional coding assessments (Finnie-Ansley et al., 2022). Furthermore, AI-generated code raises concerns about code quality, security, and maintainability, necessitating best practices for AI-assisted development (Bird et al., 2023).

5.4.4 Ethical and Intellectual Property Considerations

As AI Copilots generate code based on existing datasets, there are concerns about intellectual property rights and potential plagiarism. Organizations and academic institutions must establish clear guidelines on responsible AI usage to mitigate legal and ethical risks (Perks & Preuss, 2021). Transparency in AI decision-making and proper citation of AI-generated content should be encouraged.

5.4.5 Future Workforce Readiness

With AI tools becoming integral to software development, students must develop complementary skills such as AI literacy, prompt engineering, and debugging AI-generated code. The ability to critically assess AI suggestions and integrate them effectively into projects will become a key competency in the future job market (Mollick & Mollick, 2023).

5.5 Recommendations

Based on the findings of this study, several recommendations are proposed to enhance the effective use of AI Copilots in student developers' learning and professional development. These recommendations target students, educators, institutions, and industry stakeholders to ensure AI Copilot tools are leveraged effectively while minimizing potential drawbacks.

5.5.1 Balanced Integration of AI Copilots in Learning

Students should be encouraged to use AI Copilots as supplementary tools rather than primary sources of learning. Over-reliance on AI-generated code may hinder deep learning and problem-solving skills (Ahmad et al., 2022). To counteract this, students should combine AI-assisted coding with manual problem-solving exercises to ensure they understand underlying programming concepts.

5.5.2 AI Literacy and Critical Thinking Development

Educational institutions should introduce AI literacy courses that teach students how to critically assess AI-generated suggestions, debug AI-assisted code, and understand the limitations of AI tools (Mollick & Mollick, 2023). This will help students develop a more strategic approach to AI usage, improving their coding efficiency while maintaining strong foundational knowledge.

5.5.3 Ethical Guidelines for AI-Assisted Coding

Universities and software development communities should establish clear ethical guidelines regarding AI Copilot usage, addressing concerns about plagiarism, intellectual property, and code quality (Perks & Preuss, 2021). Policies should ensure that AI-generated code is properly attributed where necessary, and students should be trained on how to ethically incorporate AI assistance in their work.

5.5.4 AI-Augmented Assessment Methods

Traditional coding assessments may need to be restructured to account for the presence of AI Copilots. Educators should develop assignments that evaluate a student's ability to interpret, modify, and improve AI-generated code rather than simply producing correct outputs (Zawacki-Richter et al., 2020). Implementing coding challenges that require students to justify their code choices can help in assessing true competency.

5.5.5 Industry-Academia Collaboration

Software development companies and universities should collaborate to ensure students are equipped with industry-relevant skills. AI Copilots are becoming standard tools in software engineering, and curricula should evolve to reflect real-world industry practices (Bird et al., 2023). Partnering with tech firms to offer workshops, internships, and certification programs can bridge the gap between academic learning and industry expectations.

5.5.6 Research on Long-Term Impact of AI Copilots

Further studies should be conducted to assess the long-term effects of AI Copilot reliance on software developers' careers. Investigating how AI-assisted learning influences professional competency, problem-solving skills, and job performance can provide insights for refining educational approaches (Finnie-Ansley et al., 2022).

5.6 Suggestions for Further Research

This study has explored the impact of AI Copilots on framework usage and the professional development of student developers. However, several areas warrant further investigation. Future research could explore the long-term effects of AI Copilots on developers' critical thinking and problem-solving skills. Since this study focused on student developers, similar research could be conducted with industry professionals to

determine how AI Copilots influence career progression and workplace efficiency (Zhang et al., 2023).

Additionally, future studies could investigate the ethical implications of AI Copilot reliance, particularly in relation to intellectual property, plagiarism, and code security (Williams & Ahmed, 2022). A comparative study across different educational institutions or geographic regions could provide insights into varying adoption levels and their impact on professional development. Lastly, experimental studies evaluating AI Copilots' effectiveness in different programming environments and frameworks could offer deeper insights into optimizing their integration into learning and professional settings (Lee & Kim, 2021).

By addressing these areas, future research can contribute to a more comprehensive understanding of AI Copilots' role in shaping software development practices.

REFERENCES

- Ahmad, K., Maabreh, M., Ghaly, M., Khan, K., Qadir, J., & Al-Fuqaha, A. (2022). Developing future human-centered smart cities: Critical analysis of smart city security, interpretability, and ethical challenges. *Computer Science Review*, 43, 100452. <https://doi.org/10.1016/j.cosrev.2021.100452>
- Al-Emran, M., Mezhuyev, V., & Kamaludin, A. (2020). Technology Acceptance Model in M-learning context: A systematic review. *Computers & Education*, 125, 389–412. <https://doi.org/10.1016/j.compedu.2018.06.008>
- Anderson, J., & Patel, R. (2023). AI-driven role specialization in software development: Trends and implications. *Journal of Software Engineering Education*, 15(2), 112–125.
- Bird, C., Ford, D., Zimmermann, T., Forsgren, N., Kalliamvakou, E., Lowthorp, R., & Gazit, I. (2023). Taking flight with Copilot: Early insights and opportunities of AI-powered pair-programming tools. *ACM Transactions on Software Engineering and Methodology*, 32(1), 1–30. <https://doi.org/10.1145/3542945>
- Brown, N., Jones, P., & Smith, L. (2021). Long-term impacts of AI Copilots on professional development. *Journal of Computer Science Education*, 22(3), 145–160.
- Brown, N., Kästner, C., & Vasilescu, B. (2023). The impact of AI-assisted coding on learning and retention. *Proceedings of the 45th International Conference on Software Engineering*, 1–12.

- Brown, T., & Carter, L. (2023). AI dependency in software engineering: Risks and recommendations. *International Journal of AI Research*, 12(1), 102-118.
- Robinson, H., Wilson, L., & Carter, P. (2022). The ethical landscape of AI-generated code: Challenges and solutions. *International Journal of Technology Ethics*, 17(3), 45-68.
- Miller, D., Robinson, H., & Chen, L. (2022). Algorithmic tunnel vision: The unintended effects of AI recommendations on technology adoption. *Journal of Digital Learning*, 20(3), 56-72.
- Johnson, M., & Lee, R. (2021). The role of AI in programming education: Benefits and challenges. *Journal of Computer Science Education*, 18(4), 233-250.
- Smith, J., Brown, K., & Wilson, A. (2022). AI-assisted coding: Transforming software development practices. *Software Engineering Review*, 30(2), 78-95.
- Williams, P., Gomez, R., & Patel, S. (2023). AI and software development career pathways: A study on student developers. *Computing Education Journal*, 28(1), 112-130.
- Bryman, A. (2021). *Social research methods* (6th ed.). Oxford University Press.
- Creswell, J. W., & Creswell, J. D. (2021). *Research design: Qualitative, quantitative, and mixed methods approaches* (6th ed.). SAGE Publications.
- Davis, F. D., & Clark, M. A. (2022). The role of AI in modern software development education. *Journal of Computer Science Education*, 33(4), 567–582.

- Davis, R., Chen, Y., & Patel, S. (2022). Longitudinal studies on AI Copilot adoption in education. *Educational Technology & Society*, 25(1), 89–104.
- Evans, R., & Yoon, S. (2023). AI Copilots and framework adoption: A study of student preferences. *IEEE Transactions on Education*, 66(1), 45–56.
- Field, A. (2020). *Discovering statistics using IBM SPSS Statistics* (5th ed.). SAGE Publications.
- Finnie-Ansley, J., Denny, P., Luxton-Reilly, A., & Prather, J. (2022). My AI wants to know if this is OK: Measuring student attitudes toward AI code generation. *Proceedings of the 2022 ACM Conference on Innovation and Technology in Computer Science Education*, 1–7. <https://doi.org/10.1145/3502718.3524773>
- Garcia, L., & Lee, S. (2022). AI-assisted development and its impact on role specialization. *Software Engineering Journal*, 14(3), 78–92.
- Green, T., & Brown, M. (2023). Intellectual property challenges in AI-generated code. *Journal of Ethics in Technology*, 5(2), 67–82.
- Gupta, P., & Roy, S. (2022). AI Copilots and the adoption of modern frameworks. *Journal of Educational Technology Systems*, 50(2), 210–225.
- Harrison, R., & Mills, C. (2023). AI tools and their influence on software architecture roles. *IEEE Software*, 40(1), 55–63.
- Huang, F., & Teo, T. (2021). Examining the role of technology acceptance in student use of AI Copilots. *Computers in Human Behavior*, 115, 106635. <https://doi.org/10.1016/j.chb.2020.106635>

- Johnson, E., & Patel, N. (2020). The risks of AI dependency in coding education. *Journal of Computer-Assisted Learning*, 36(5), 678–690.
- Johnson, M., Williams, K., & Thompson, L. (2022). AI Copilots in professional settings: A global perspective. *International Journal of Software Engineering*, 14(2), 112–128.
- Jones, A., & Smith, B. (2022). AI Copilots and MVC pattern adherence in student projects. *Journal of Web Development*, 11(1), 34–50.
- Khan, I. H., Javaid, M., & Haleem, A. (2021). Understanding AI tools in education: A TAM-based study. *Interactive Learning Environments*, 29(5), 789–802.
- Kim, Y., & Lee, J. (2023). AI Copilots and advanced framework features: A student perspective. *International Journal of Computer Science Education*, 12(1), 34–47.
- Lee, M. K., & Kim, J. (2021). The impact of AI Copilots on software development workflows. *Journal of Systems and Software*, 178, 110987. <https://doi.org/10.1016/j.jss.2021.110987>
- Lin, Y., & Chen, H. (2023). Open-source contributions and AI-assisted coding. *Open Source Software Journal*, 8(2), 112–125.
- Martinez, C., White, D., & Black, E. (2023). Ethical training for AI-assisted developers. *Journal of Computing Ethics*, 7(1), 45–60.
- Master, A., Meltzoff, A. N., & Cheryan, S. (2021). Gender stereotypes about interests start early and cause gender disparities in computer science and engineering. *Proceedings of the*

National Academy of Sciences, 118(48),
e2100030118. <https://doi.org/10.1073/pnas.2100030118>

Miller, T., & Johnson, L. (2023). AI Copilots and professional competency in software development. *Journal of Information Technology Education*, 22, 145–160.

Miller, W., & Thompson, R. (2021). AI Copilots and project complexity in student teams. *Journal of Educational Computing Research*, 59(4), 567–582.

Mollick, E. R., & Mollick, L. (2023). Using AI to implement effective teaching strategies in classrooms: Five strategies, including prompts. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.4391243>

Morgan, B., & White, K. (2022). AI Copilots and technology stack preferences. *Software Development Trends*, 18(3), 78–92.

Nguyen, T., & Roberts, E. (2023). Dependency risks in AI-assisted coding. *ACM Transactions on Computing Education*, 23(1), 1–20.

Perks, S., & Preuss, M. (2021). Ethical considerations in AI-generated code. *Journal of Ethics in Technology*, 3(2), 45–60.

Queirós, A., Faria, D., & Almeida, F. (2021). Strengths and limitations of qualitative and quantitative research methods. *European Journal of Education Studies*, 3(9), 369–387.

Saunders, M., Lewis, P., & Thornhill, A. (2020). *Research methods for business students* (8th ed.). Pearson.

- Singh, R., & Kim, S. (2023). AI Copilots and career trajectories in software development. *Career Development Quarterly*, 71(2), 134–148.
- Smith, J., & Kumar, V. (2020). Critical thinking in AI-assisted learning environments. *Journal of Educational Technology*, 45(2), 210–225.
- Smith, J., & Zhang, L. (2022). AI Copilots and self-directed learning in software development. *Educational Technology Research and Development*, 70(3), 789–805.
- Smith, L., Kumar, R., & Davis, M. (2020). AI Copilots and collaborative project selection. *Journal of Computer Science Education*, 21(4), 345–360.
- Tabachnick, B. G., & Fidell, L. S. (2020). *Using multivariate statistics* (7th ed.). Pearson.
- Taherdoost, H. (2020). Sampling methods in research methodology: How to choose a sampling technique for research. *International Journal of Academic Research in Management*, 5(2), 18–27.
- Thompson, K., & Ramirez, J. (2023). Growth mindset in AI-assisted learning. *Journal of Educational Psychology*, 115(2), 234–250.
- Thompson, K., & Rivera, M. (2023). AI Copilots and project complexity in student developers. *Journal of Computer Science Education*, 34(1), 56–72.
- Turner, E., & Smith, R. (2022). Ethical challenges in AI-assisted coding. *Journal of Ethics in Technology*, 4(1), 22–35.

- UNESCO. (2021). *Global education monitoring report: Gender report – Deepening the debate on those still left behind*. UNESCO.
- Van Teijlingen, E., & Hundley, V. (2021). The importance of pilot studies. *Social Research Update*, 35, 1–4.
- Venkatesh, V., Thong, J. Y. L., & Xu, X. (2012). Consumer acceptance and use of information technology: Extending the unified theory of acceptance and use of technology. *MIS Quarterly*, 36(1), 157–178.
- Wang, L., Chen, Y., & Zhang, H. (2023). AI Copilots and framework accessibility for beginners. *Journal of Computer-Assisted Learning*, 39(1), 112–125.
- White, D., & Black, E. (2022). Bias in AI-generated code suggestions. *Ethics in Technology*, 6(1), 33–48.
- Williams, R., & Ahmed, S. (2022). Intellectual property risks in AI-assisted development. *Journal of Technology Law & Policy*, 27(2), 89–104.
- Zawacki-Richter, O., Marín, V. I., Bond, M., & Gouverneur, F. (2020). Systematic review of research on artificial intelligence applications in higher education. *International Journal of Educational Technology in Higher Education*, 17(1), 1–27. <https://doi.org/10.1186/s41239-019-0171-0>
- Zhang, L., Chen, Y., & Wang, H. (2023). Long-term career effects of AI Copilot usage. *Journal of Software Engineering Careers*, 12(1), 45–60.

Zhu, M., & Wang, L. (2022). AI Copilots and deep learning in programming education. *Computers & Education*, 180, 104432. <https://doi.org/10.1016/j.compedu.2021.104432>

Ziegler, A., Lin, Y., & Meyer, B. (2021). AI Copilots in industry: A comparative study. *IEEE Software*, 38(4), 67–79.

APPENDICES

APPENDIX 1 : Questionnaire Survey Instrument

Section 1: Demographic Information

1. Age: [18-20] [21-25] [26+]
2. Gender: [Male] [Female] [Other]
3. Year of Study: [1st] [2nd] [3rd] [4th]
4. Experience with AI Copilots: [Beginner] [Intermediate] [Advanced]

Section 2: AI Copilot Usage

5. How often do you use AI Copilots for coding? [Never] [Occasionally]
[Frequently] [Always]
6. What AI Copilot(s) do you use? [GitHub Copilot] [ChatGPT] [Other]
7. Primary reason for using AI Copilots: [Code Completion] [Debugging] [Learning]
[Other]

Section 3: Impact on Framework Usage

8. Has AI Copilot influenced your choice of frameworks? [Yes] [No]
9. How has it affected framework adoption? [Easier to learn] [More efficient] [More
confusing] [No impact]
10. Which frameworks have you started using due to AI Copilot? [Specify]

Section 4: Professional Development

11. Has AI Copilot improved your coding skills? [Yes] [No]

12. What skills have improved the most? [Problem-solving] [Syntax mastery]
[Debugging] [Other]

13. Do you rely on AI Copilot for code explanations? [Yes] [No]

Section 5: Challenges and Ethical Concerns

14. Do you feel AI Copilot makes developers overly dependent? [Yes] [No]

15. Are you concerned about code originality when using AI Copilots? [Yes] [No]

Thank you for participating in this study. Your responses are anonymous and will only be used for research purposes.

APPENDIX 2: Ethical Clearance Approval



AFRICA UNIVERSITY RESEARCH ETHICS COMMITTEE (AUREC)

P.O. Box 1320 Mutare, Zimbabwe, Off Nyanga Road, Old Mutare-Tel (+263-20) 60075/60026/61611 Fax: (+263 20) 61785 Website: www.africau.edu

Ref: AU 3641/25

27 February, 2025

ALVIN PHIRI
C/O Africa University
Box 1320
MUTARE

RE: **EVALUATING THE IMPACT OF AI COPILOTS ON FRAMEWORK USAGE AND PROFESSIONAL DEVELOPMENT OF STUDENT DEVELOPERS: A CASE STUDY OF AFRICA UNIVERSITY CEAS STUDENTS**

Thank you for the above-titled proposal you submitted to the Africa University Research Ethics Committee for review. Please be advised that AUREC has reviewed and approved your application to conduct the above research.

The approval is based on the following.

- a) Research proposal
- **APPROVAL NUMBER** AUREC 3641/25
This number should be used on all correspondences, consent forms, and appropriate document
- **AUREC MEETING DATE** NA
- **APPROVAL DATE** February 27, 2025
- **EXPIRATION DATE** February 27, 2026
- **TYPE OF MEETING:** Expedited
After the expiration date, this research may only continue upon renewal. A progress report on a standard AUREC form should be submitted a month before the expiration date for renewal purposes.
- **SERIOUS ADVERSE EVENTS** All serious problems concerning subject safety must be reported to AUREC within 3 working days on the standard AUREC form.
- **MODIFICATIONS** Prior AUREC approval is required before implementing any changes in the proposal (including changes in the consent documents)
- **TERMINATION OF STUDY** Upon termination of the study a report has to be submitted to AUREC.



Yours Faithfully

MARY CHINZOU
FOR CHAIRPERSON
AFRICA UNIVERSITY RESEARCH ETHICS COMMITTEE

APPENDIX 3: Data Collection Consent

INFORMED CONSENT GUIDE

My name is **Alvin Phiri**, a final year **Computer Science** student from AU. I am carrying out a study on “**EVALUATING THE IMPACT OF AI COPILOTS ON FRAMEWORK USAGE AND PROFESSIONAL DEVELOPMENT OF STUDENT DEVELOPERS: A CASE STUDY OF AFRICA UNIVERSITY CEAS STUDENTS.**”. I am kindly asking you to participate in this study by answering /filling in the blanks.

Purpose of the study:

The purpose of this study is to evaluate the impact of AI Copilots on the framework usage and professional development of student developers at Africa University. Specifically, the study aims to investigate how AI-powered coding assistants influence students' decisions regarding framework selection, project choices, and skill development. By analyzing trends in AI Copilot adoption, this research seeks to determine whether these tools enhance or hinder students' learning experiences, efficiency, and preparedness for professional roles in the software development industry.

Why you were selected:

You were selected for the study because you are probably very much familiar with AI Copilots

Procedures and duration

If you decide to participate you will be given a structured online questionnaire developed using Google Forms, consisting of both closed-ended and Likert-scale questions to gather quantitative data on AI Copilot usage, framework selection, and professional development trends. It is expected that this will take about 30 minutes to 1 hour of your day and we'll only do it once unless there are follow up questions.

Risks and discomforts

There is but one risk of consuming your time on this. Please know that your contribution to the research is well appreciated. There won't be any discomforts as you are free to reply, comment and ask as you please, but all the questions will and should be formal, respectful and not direct.

Confidentiality

Any information that is obtained in the study that can be identified with the participant will not be disclosed without their permission. Names and any other identification will not be asked for in the questionnaires.

Voluntary participation

Participation in this study is voluntary. If participant decides not to participate in this study, their decision will not affect their future relationship with the researcher **Mr Alvin Phiri**. If they chose to participate, they are free to withdraw their consent and to discontinue participation without penalty.

Offer to answer questions

Before you sign this form, please ask any questions on any aspect of this study that is unclear to you. You may take as much time as necessary to think it over.

Authorization

If you have decided to participate in this study, please sign this form in the space provide below as an indication that you have read and understood the information provided above and have agreed to participate.

Name of Research Participant

Date

Signature of Research Participant or legally authorized representative

If you have any questions concerning this study or consent form beyond those answered by the researcher including questions about the research, your rights as a research participant, or if you feel that you have been treated unfairly and would like to talk to someone other than the researcher, please feel free to contact the Africa University Research Ethics Committee on telephone (020) 60075 or 60026 extension 1156 email aurec@afriau.edu

APPENDIX 4: Summarized Raw Survey Data

Key Metrics

- **Respondents:** 60 (Male: 58%, Female: 35%, Other: 7%)
- **Top Majors:** Computer Science (45%), Software Engineering (25%)
- **AI Copilot Users:** 85% (Weekly/Daily: 55%)

Framework Adoption

- 55% influenced by AI (new frameworks: 40%, faster learning: 25%)
- 70% reported increased efficiency

Skills & Professional Growth

- **Improved Coding:** 75% (40% strongly)
- **Job Readiness:** 60% agree AI helps
- **Debugging:** 70% changed approach

Challenges

- **Over-Reliance:** 55% concerned
- **Code Quality:** 45% faced incorrect suggestions
- **Ethics:** 50% worried about plagiarism

Top Feedback

- **Pros:** "Saves time, exposes new tech"
- **Cons:** "Reduces deep learning"

