Gayatri Chavan, 2026, 14:6
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

A Comparative Survey-Based Study of AI Adoption, Code Verification Practices, and Future Career Readiness Among Computing Students and Software Developers in India

Gayatri Chavan[1] Pramod Devtarase[2]
Department Of Mca, Siddhant
Institute Of Computer Application, Sudumbre, Pune, Maharashtra, India.

Abstract- Artificial Intelligence (AI) is becoming a useful part of software development, particularly in activities such as writing programs, debugging errors, testing code, and learning programming concepts. This research focuses on how computing students and software professionals in India use AI tools, check AI-generated code, and prepare themselves for changes in the software industry. For this study, information was gathered through a Google Forms questionnaire completed by 113 participants. The participants included undergraduate and postgraduate computing students, working software developers, and IT professionals. The responses were examined using frequency counts, percentages, and comparisons between the respondent groups. The survey results indicate that 96.5% of participants use AI tools for programming-related tasks. At the same time, 65.5% reported that they frequently or very frequently face incorrect or incomplete code generated by AI tools. The responses also show that 57.5% of participants always check AI-generated code before using it. In addition, 89.4% believe that AI will bring significant or moderate changes to the skills needed in future software development. The study indicates that AI tools are becoming common in programming activities, but their outputs still require careful examination. It also highlights the importance of programming fundamentals, problem-solving, code verification, and continuous learning. Overall, AI can assist software professionals, but human understanding and responsibility remain important for producing reliable software.

Keywords: Artificial Intelligence, Software Development, AI Adoption, Code Verification, Career Readiness, Computing Students, IT Professionals.

I. INTRODUCTION

Artificial Intelligence has become an important part of the modern software development process. AI-based tools are now used by students, programmers, and IT professionals to perform different programming-related tasks. These tools can help users understand programming concepts, generate code, identify errors, prepare test cases, and improve existing programs.

Some commonly used AI tools in software development include ChatGPT, Google Gemini, Claude, GitHub Copilot, and Cursor. These tools provide suggestions and solutions that may reduce development time and support learning. As a result, AI is changing the way people learn programming and develop software applications.

However, AI-generated code is not always correct or reliable. Sometimes, it may contain logical errors, incomplete solutions, outdated information, or security weaknesses. Therefore, developers should not depend completely on AI-generated outputs. They must understand the generated code, test it with different inputs, check its logic, and verify whether it meets the required conditions.

The increasing use of AI also raises questions about the future skills required in the software industry. Programming knowledge remains important, but skills such as problem-solving, critical thinking, debugging, testing, code review, prompt writing, and cybersecurity awareness are also becoming necessary.

© 2026 Authors Gayatri Chavan, This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly credited.

Computing students need to understand these changes because they will enter a job market where AI tools are commonly used. Similarly, working software developers and IT professionals need to update their knowledge and adapt to new development practices.

This research presents a comparative survey-based study of AI adoption, code verification practices, and future career readiness among computing students and software professionals in India. The study uses responses collected through Google Forms to understand how different respondent groups use AI tools, verify generated code, and prepare for changes in the software development field.

II. LITERATURE REVIEW

Artificial Intelligence has become an important research area in software engineering. Sofian et al. [1] presented a systematic mapping study of AI techniques used in different software engineering activities. Their study discussed how AI techniques can support software development processes and identified areas where further research is required.

Martínez-Fernández et al. [2] conducted a survey on software engineering practices for AI-based systems. Their research highlighted the importance of software quality, testing, dependability, and maintenance when developing AI-based applications. This indicates that proper software engineering practices remain important in AI-related development.

Peng et al. [3] studied the impact of GitHub Copilot on developer productivity through a controlled experiment. Their findings showed that developers using Copilot completed a programming task faster than the control group. The study indicates that AI coding assistants can support developers and improve productivity in certain programming tasks.

Tabassi [4], in the NIST Artificial Intelligence Risk Management Framework, discussed the importance of managing risks associated with AI systems. The framework highlights the need for trustworthy, reliable, and responsible AI practices. These principles are also relevant when developers use AI-generated code.

Fu et al. [5] examined security weaknesses in AI-generated code from GitHub projects. Their study identified security-related problems in code generated by AI coding assistants. This highlights the importance of checking and testing AI-generated code before using it in real software projects.

A recent systematic mapping study on AI for source code understanding [6] reviewed research related to AI techniques, code understanding, and software engineering tasks. It identified the growing use of deep learning and transformer-based approaches and highlighted areas that require further investigation.

Overall, previous studies have examined AI adoption in software engineering, developer productivity, software quality, and security. These studies provide a foundation for understanding the increasing role of AI in programming and software development.

III. RESEARCH GAP

Previous studies have mainly focused on AI techniques in software engineering, developer productivity, AI-based systems, and security issues in generated code. However, there is a need to understand how computing students and working software professionals use AI tools in their regular programming activities.

There is also a need to examine how frequently users verify AI-generated code, what types of errors they experience, and how prepared they feel for future changes in software development.

The present study addresses this gap through a comparative survey of undergraduate computing students, postgraduate computing students, working software developers, and IT professionals in India. It analyses 113 responses to understand AI tool adoption, code verification practices, and future career readiness.

Unlike studies that focus mainly on the technical performance of AI tools, this research considers the opinions and experiences of different user groups. It provides survey-based findings that can help identify important skills required for AI-assisted software development.

1. Objecti
es Of The Study

The main objectives of this research are:

- To study the adoption and usage of AI tools among computing students and software professionals.
- To identify the commonly used AI tools for programming-related activities.
- To examine how frequently users verify AI-generated code.
- To understand the common errors and limitations found in AI-generated code.
- To compare the usefulness of AI tools among different respondent groups.
- To study the preparedness of respondents for AI-assisted software development.
- To identify the important technical and professional skills required for future software development careers.
- To understand respondents' opinions about the impact of AI on future software development skills.

IV. RESEARCH METHODOLOGY

1. Research Design
This study follows a quantitative and survey-based research design. The research focuses on collecting information about AI tool adoption, code verification practices, and career readiness among computing students and software professionals.

2. Data Collection Method
Primary data was collected using a structured questionnaire created through Google Forms. The questionnaire included questions related to AI tool usage, programming activities, code verification, errors in AI-generated code, career preparedness, and future skills.

3. Sample Size
A total of **113 responses** were collected and considered for the study.

4. Sampling Technique Convenience sampling was used to collect responses from participants who were accessible and willing to take part in the survey.

5. Respondent Categories
The respondents were divided into the following four categories:

Table 1: Distribution of Respondents

Respondent Category	Number of Respondents	Percentage
Undergraduate Computing Students	25	22.1%
Postgraduate Computing Students	27	23.9%
Working Software Developers	28	24.8%
IT Professionals	33	29.2%
Total	113	100%

6. Data Analysis Method
The collected responses were analysed using descriptive and comparative analysis. Frequency counts and percentages were calculated to identify overall response patterns. Tables, pie charts, bar charts, and grouped bar charts were used to present the findings in a clear and systematic manner.

The comparative analysis examined the responses of four categories: undergraduate computing students, postgraduate computing students, working software developers, and IT professionals. This analysis helped identify similarities and differences in their AI tool usage, code verification practices, and career preparedness.

7. Scope Of The Study
The study focuses on the use of AI tools in programming and software development. It examines users' experiences with AI-generated code, their verification practices, and their opinions about future career requirements. The study is limited to the 113 responses collected through the survey and does not claim to represent every computing student or software professional in India.

V. RESULTS AND DATA ANALYSIS

1. Respondent Distribution Chart Type: Pie Chart

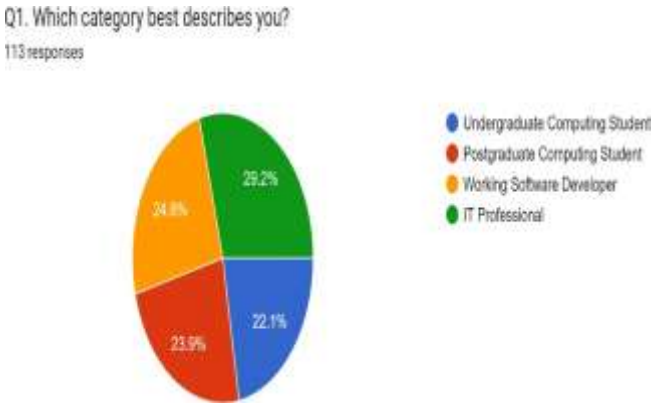

Figure 1: Respondent Distribution (N = 113)

1. Analysis:
The above pie chart shows the distribution of 113 respondents who participated in the survey. Out of the total respondents, 33 (29.2%) were IT Professionals, while 28 (24.8%) were Working Software Developers. Postgraduate Computing Students represented 27 (23.9%) respondents, and Undergraduate Computing Students represented 25 (22.1%). The data shows that all four categories are represented in the survey, with IT Professionals forming the largest group.

2. Result:
The survey includes responses from both computing students and working professionals. This distribution provides an opportunity to compare their experiences and opinions regarding AI adoption, code verification practices, and future career readiness.

2. Ai Tool Adoption

1. **Use of AI Tools for Programming Chart Type:** Pie Chart

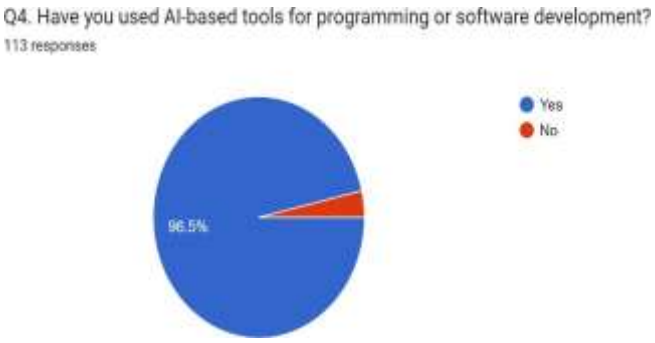

Figure 2: Use of AI Tools for Programming (N = 113)

1. Analysis:
The survey results show that 52 respondents (46.0%) often use AI tools for programming-related tasks, while 44 respondents (38.9%) use them sometimes. Only 5 respondents (4.4%) use AI tools very often. A total of 10 respondents (8.8%) rarely use AI tools, and only 2 respondents (1.8%) never use them.

2. Analysis:
The survey results show that 109 respondents (96.5%) use AI tools for programming-related activities, while only 4 respondents (3.5%) reported that they do not use AI tools. This indicates that AI tools have become widely used among the surveyed computing students and working professionals.

3. Result:
The majority of respondents use AI tools for programming. This shows a high level of AI adoption among the participants.

3. Most Commonly Used Ai Tools Chart Type: Bar Chart

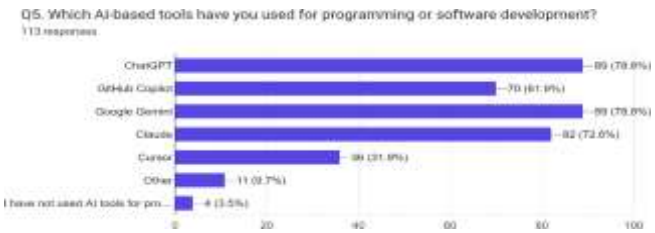

Figure 3: Most Commonly Used AI Tools (Multiple Responses Allowed)

1. Analysis:
The results show that ChatGPT and Google Gemini are the most commonly selected AI tools, with 89 respondents (78.8%) each. Claude was selected by 82 respondents (72.6%), followed by GitHub Copilot with 70 respondents (61.9%). Cursor was selected by 36 respondents (31.9%), while 11 respondents (9.7%) selected other tools.

Since respondents could select more than one AI tool, the total percentage is greater than 100%.

2. Result:
ChatGPT and Google Gemini are the most popular AI tools among the surveyed participants. The results indicate that respondents use different AI tools to support their programming-related activities.

VI. FREQUENCY OF AI TOOL USAGE

1. **Frequency of AI Tool Usage Chart Type:** Pie chart

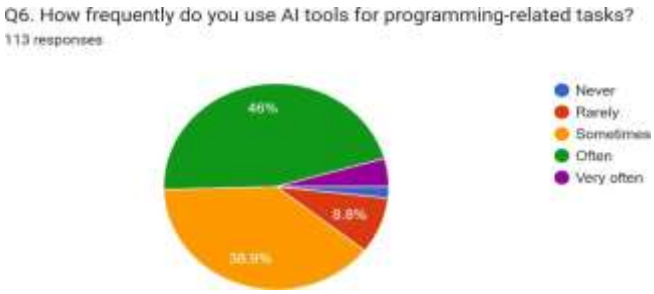

Figure 4: Frequency of AI Tool Usage (N = 113)

The results indicate that AI tools are regularly used by most participants. The highest percentage of respondents selected the "Often" option.

1. Result:
The findings show that AI tools have become a regular part of programming-related activities for the surveyed participants. Most respondents use AI tools either often or sometimes.

2. Activities For Which Ai Tools Are Used
1. **Common Activities Performed Using AI Tools Chart Type:** Bar Chart

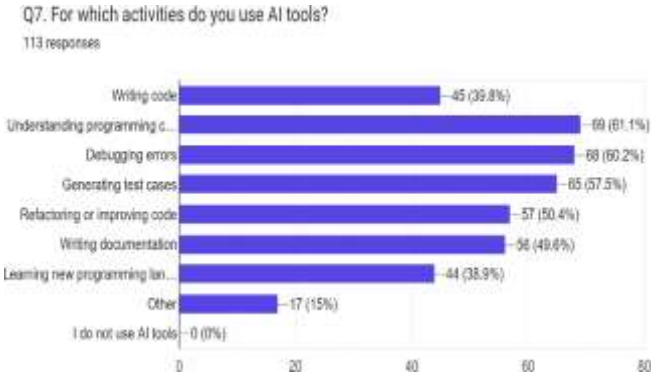

Figure 5: Activities for Which AI Tools Are Used

1. Analysis:
The survey results show that the most common use of AI tools is for understanding programming concepts, selected by 69 respondents. Debugging code was selected by 68 respondents, followed by generating test cases with 65 responses. Refactoring or improving code received 57 responses, while writing documentation received 56 responses. Writing code was selected by 45 respondents.

These findings indicate that respondents use AI tools not only for code generation but also for learning, debugging, testing, and improving existing code.

3. Result:
The results show that AI tools support different stages of software development. Understanding programming concepts, debugging, and generating test cases are the most commonly reported activities.

3. Code Verification Practices

1. **Frequency of AI-Generated Code Verification Chart Type:** Pie Chart

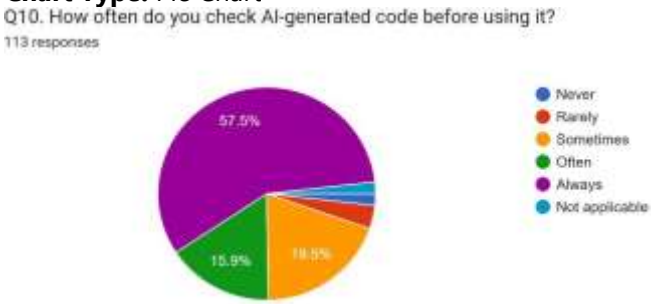

Figure 6: Frequency of AI-Generated Code Verification (N = 113

1. Analysis:

The survey results show that 65 respondents (57.5%) always verify AI-generated code before using it. Another 18 respondents (15.9%) often verify the code, while 22 respondents (19.5%) verify it sometimes. Only 4 respondents (3.5%) rarely verify the code, and 2 respondents (1.8%) never verify it. Two respondents selected Not Applicable.

These findings indicate that most participants understand the importance of checking AI-generated code before using it in programming tasks.

2. Result:

The majority of respondents follow code verification practices. This suggests that although AI tools are widely used, human checking remains an important part of software development.

4. Methods Used For Code Verification Chart Type: Pie Chart



Figure 7: Methods Used for AI-Generated Code Verification

1. Analysis:

The survey results show that testing the code with different inputs is the most commonly reported verification method, selected by 25 respondents (22.1%). Checking logical errors and security issues were each selected by 22 respondents (19.5%). Comparing the output with official documentation or trusted sources was selected by 18 respondents (15.9%). Running the code to check whether it works was reported by 12 respondents (10.6%), while reading and understanding the code was selected by 9 respondents (8.0%). Additionally, 2 respondents (1.8%) ask another person to review the code, and 3 respondents (2.7%) selected Not Applicable.

These findings indicate that respondents use different methods to verify AI-generated code, including testing, logical checking, security checking, and referring to trusted sources.

2. Result:

Testing code with different inputs is the most frequently reported verification method. The findings highlight the importance of testing, logical verification, security awareness, and understanding AI-generated code to ensure its correctness and reliability**.**

5. Issues And Errors In Ai-Generated Code

1. **Frequency of Incorrect or Incomplete AI-Generated Code**

Chart Type: Pie Chart

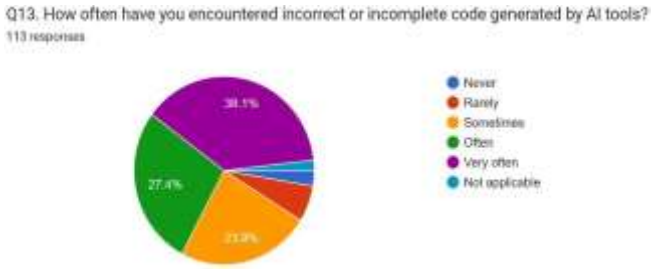

Figure 8: Frequency of Incorrect or Incomplete AI-Generated Code (N = 113)

1. Analysis:

The survey results show that 43 respondents (38.1%) very often encounter incorrect or incomplete AI-generated code. Another respondents (27.4%) often face such issues, while 27 respondents (23.9%) sometimes experience them. Only 7 respondents (6.2%) rarely encounter these problems, and 3 respondents (2.7%) never face them. Two respondents selected Not Applicable .Overall, 74 respondents (65.5%) reported that they often or very often encounter errors in AI-generated code. This indicates that AI-generated code may require careful checking and testing before implementation.

2. Result:

The findings show that incorrect or incomplete AI-generated code is a common issue among the surveyed participants. Therefore, developers and students should

not depend entirely on AI-generated outputs without proper verification.

6. Importance Of Understanding Ai-Generated Code Chart Type: Bar Chart

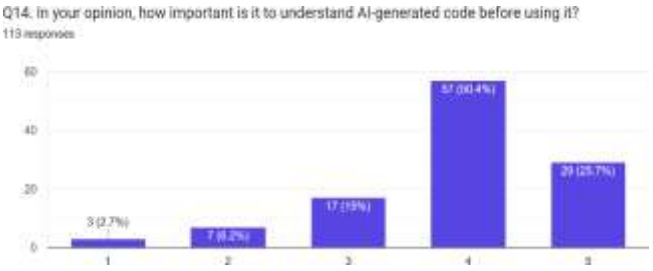

Figure 9: Importance of Understanding AI-Generated Code (N= 113)

1. Analysis:

The survey indicates that 86 respondents (76.1%) gave a rating of 4 or 5 for the importance of understanding AI-generated code. This shows that most participants believe programmers should understand the code generated by AI rather than simply copying and using it.

2. Result:

Understanding AI-generated code is considered important by the majority of respondents. This highlights the continued importance of programming knowledge and logical thinking when working with AI tools.

7. Career Readiness And Future Skills

1. **Self-Assessment of Career Preparedness Chart Type:** Bar chart

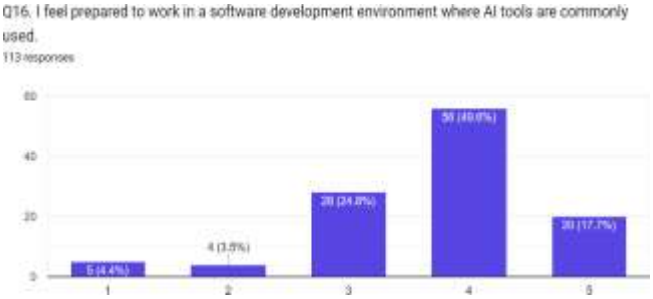

Figure 10: Self-Assessment of Career Preparedness (N = 113)

1. Analysis:

The results show that 56 respondents (49.6%) rated their career preparedness as 4, while 20 respondents (17.7%) gave the highest rating of 5. A total of 28 respondents (24.8%) selected rating 3. Only 9 respondents (7.9%) selected ratings 1 and 2.

Overall, 76 respondents (67.3%) rated their career preparedness as 4 or 5. This indicates that most participants feel reasonably prepared for future career opportunities in an AI-influenced software development environment.

2. Result:

The majority of respondents reported a positive level of career preparedness. However, some participants selected moderate or low ratings, indicating a need for continuous learning and skill development.

8. Skills Required For Future Software Development Chart Type: Bar Chart

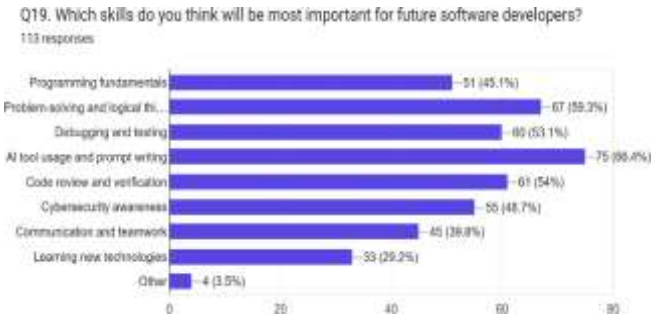

Figure 11: Skills Required for Future Software Development

1. Analysis:

The survey results show that AI tool usage and prompt writing received the highest number of responses (75). Problem solving and logical thinking were selected by 67 respondents. Code review and verification received 61 responses, followed by debugging and testing with 60 responses. Cybersecurity awareness was selected by 55 respondents, while programming fundamentals received 51 responses. Communication and teamwork received 45 responses, and learning new technologies received 33 responses.

These findings indicate that respondents consider both

technical and AI-related skills important for future software development careers.

2. Result:

AI tool usage, problem solving, code verification, and debugging are among the most preferred future skills. The findings suggest that future software professionals will need a combination of programming knowledge, AI awareness, and analytical skills.

9. Expected Changes In Software Development Skills Chart Type: Pie Chart

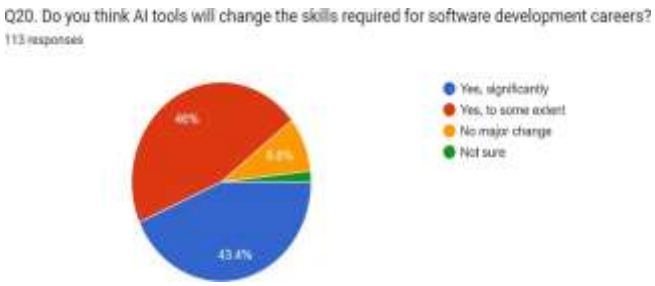

Figure 12: Expected Changes in Software Development Skills (N = 113)

1. Analysis:

The results show that 49 respondents (43.4%) believe that AI will significantly change the skills required in software development. Another 52 respondents (46.0%) expect changes to some extent. Only 10 respondents (8.8%) believe there will be no major change, while 2 respondents (1.8%) are not sure.In total, 101 respondents (89.4%) expect AI to bring changes to future software development skills.

2. Result:

The majority of respondents believe that AI will influence the skills required in the software development industry. Therefore, continuous learning and adaptation to new technologies will be important for future career growth.

10. Comparative Analysis Across Respondent Categories

1. **Comparison of AI Tool Usefulness and Career Preparedness**

The average ratings were calculated based on the responses provided on a five-point rating scale. These averages were used to compare AI tool usefulness and career preparedness across the four respondent categories.

Table 2: Comparison of AI Tool Usefulness and Career Preparedness

Respondent Category	AI Tool Usefulness (Average)	Career Preparedness (Average)
Undergraduate Students	4.00	3.60
Postgraduate Students	3.52	3.33
Working Software Developers	4.04	3.96
IT Professionals	3.94	3.94

Chart Type: Grouped Bar Chart

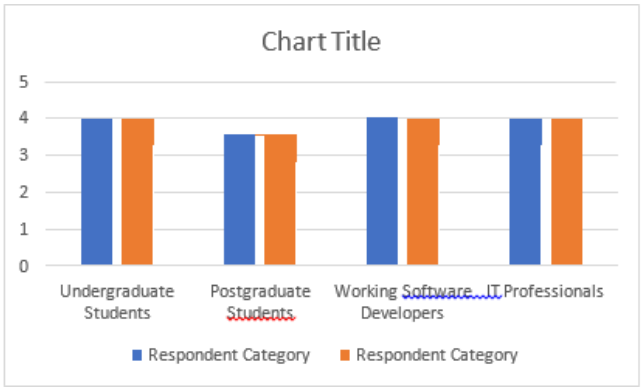

Figure 13: Comparison of AI Tool Usefulness and Career Preparedness Across Respondent Categories

1. Analysis:

The comparison shows that working software developers gave the highest average rating for AI tool usefulness (4.04), followed by undergraduate students (4.00). IT professionals gave an average rating of 3.94, while postgraduate students gave 3.52.

Regarding career preparedness, working software developers and IT professionals reported higher average ratings of 3.96 and 3.94, respectively. Undergraduate students reported an average rating of 3.60, while postgraduate students reported 3.33.

The results indicate some differences in the opinions of students and working professionals. Working professionals reported higher average career preparedness ratings than both student groups.

2 Result:
The comparative analysis suggests that working professionals generally feel more prepared for AI-related changes in software development than computing students. However, all respondent categories gave positive average ratings for AI tool usefulness.

11. Overall Comparative Findings
The survey findings indicate that AI tools are widely used among the four respondent categories. The comparative analysis shows differences in the average ratings of AI tool usefulness and career preparedness across these categories.

The findings also highlight the importance of programming fundamentals, problem-solving, code verification, and continuous learning in AI-assisted software development.

12. Result:
The comparison shows that working software developers and IT professionals reported higher average career preparedness ratings than the student groups. The findings emphasize the importance of developing technical knowledge and adapting to changes in AI-assisted software development.

13. Summary Of Results And Findings
The analysis of 113 responses provides several important findings regarding AI adoption, code verification practices, and future career readiness among computing students and software professionals.

- **AI Adoption:** A total of 109 respondents (96.5%) reported using AI tools for programming-related activities.
- **Popular AI Tools:** ChatGPT and Google Gemini were the most commonly selected tools, with 89 responses (78.8%) each.
- **Frequency of Usage:** Most respondents use AI tools often (46.0%) or sometimes (38.9%).
- **AI Applications:** Understanding programming concepts, debugging code, and generating test cases were the most commonly reported activities.
- **Code Verification:** Around 57.5% of respondents always verify AI-generated code before using it.
- **AI-Generated Errors:** A total of 74 respondents (65.5%) often or very often encounter incorrect or incomplete AI-generated code.
- **Understanding AI Code:** Around 76.1% of respondents rated understanding AI-generated code as important (rating 4 or 5).
- **Career Preparedness:** Approximately 67.3% of respondents rated their career preparedness at 4 or 5 on a five-point scale.
- **Future Skills:** AI tool usage, problem-solving, code verification, and debugging were among the most selected future skills.
- **Future Changes:** Around 89.4% of respondents expect AI to change the skills required in software development.

14. Overall Result
The overall findings indicate that AI tools have become an important part of programming and software development activities. However, the results also highlight the need for proper code verification, programming fundamentals, problem-solving abilities, and continuous learning. Both computing students and working professionals recognize the growing importance of AI-related skills for future career development.

VII. CONCLUSION AND FUTURE SCOPE

1. Conclusion
This study examined AI adoption, code verification practices, and future career readiness among computing students and software developers in India. The study was based on 113 responses collected through a questionnaire.

The findings show that AI tools are widely used for programming-related activities. ChatGPT, Google Gemini, Claude, and GitHub Copilot were among the commonly selected tools. Respondents use these tools for understanding programming concepts, debugging, generating test cases, improving code, and documentation.

The study also found that code verification is an important practice when using AI-generated code. Many respondents reported encountering incorrect or incomplete AI-generated outputs. Therefore, testing, logical checking, and understanding the generated code are necessary before implementation.

The results further indicate that most respondents believe AI will bring changes to future software development skills. AI tool usage, problem-solving, code verification, debugging, and cybersecurity awareness are considered important skills.

Overall, AI can support software development by improving productivity and assisting programmers in different tasks. However, it should be used as a supporting tool rather than a complete replacement for programming knowledge and human judgment.

2. Limitations Of The Study

The study has the following limitations:

- The study is based on a sample of 113 respondents.
- Convenience sampling was used for collecting responses.
- The survey was conducted through Google Forms.
- The findings are based on respondents' opinions and self-reported experiences.
- The study focuses on selected computing students and working professionals in India.
- The study does not experimentally measure the actual productivity or accuracy of AI-generated code.

3. Future Scope

The study can be extended in the following ways:

- Future research can include a larger number of respondents from different regions of India.
- A detailed comparison can be conducted between different AI coding tools.
- Future studies can examine the impact of AI tools on actual software development productivity.
- Research can focus on the security, reliability, and accuracy of AI-generated code.
- The role of AI in software development education and professional training can be studied further.
- Future research can examine how AI adoption influences job opportunities and the skills expected from software developers.

REFERENCES

1. H. Sofian, N. A. Md. Yunus, and R. Ahmad, "Systematic Mapping: Artificial Intelligence Techniques in Software Engineering," IEEE Access, vol. 10, pp. 51021–51040, 2022. https://doi.org/10.1109/ACCESS.2022.3174115
2. S. Martínez-Fernández et al., "Software Engineering for AI-Based Systems: A Survey," ACM Transactions on Software Engineering and Methodology, vol. 31, no. 2, 2022. https://doi.org/10.1145/3487043
3. S. Peng, E. Kalliamvakou, P. Cihon, and M. Demirer, "The Impact of AI on Developer Productivity: Evidence from GitHub Copilot," arXiv preprint arXiv:2302.06590, 2023. https://doi.org/10.48550/arXiv.2302.06590
4. E. Tabassi, "Artificial Intelligence Risk Management Framework (AI RMF 1.0)," National Institute of Standards and Technology, NIST AI 100-1, 2023. https://doi.org/10.6028/NIST.AI.100-1
5. Y. Fu, P. Liang, A. Tahir, Z. Li, M. Shahin, J. Yu, and J. Chen, "Security Weaknesses of Copilot-Generated Code in GitHub Projects: An Empirical Study," ACM Transactions on Software Engineering and Methodology, vol. 34, no. 8, Article 218, 2025. https://doi.org/10.1145/3716848
6. D. R. Fudholi and A. Capiluppi, "Artificial Intelligence for Source Code Understanding Tasks: A Systematic Mapping Study," Information and Software Technology, vol. 189, Article 107915, 2026. https://doi.org/10.1016/j.infsof.2025.107915