ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

Utilization of Github Copilot And Perceived Improvement in Debugging Skills among College Students at Quezon City University

Carissa Jade M. Carinan[1], Harold R. Lucero[2], Marvin N. Navarette[3], Kristine U. Ortiz[4], John Kyle S. Paroni[5], Charles Bryan C. Duyag [6]

College of Computer Studies, Quezon City University

DOI: https://doi.org/10.51583/IJLTEMAS.2026.150500182

Received: 13 May 2026; Accepted: 18 May 2026; Published: 12 June 2026

ABSTRACT

This study examined the utilization of GitHub Copilot and its relationship with the perceived improvement in debugging skills among Bachelor of Science in Information Technology students at Quezon City University. Using a quantitative descriptive-correlational design, the study involved 150 BSIT students selected through purposive sampling based on their prior experience using GitHub Copilot. Data were gathered through a structured online questionnaire administered via Google Forms and analyzed using weighted mean and Pearson Product-Moment Correlation Coefficient (Pearson r). Findings revealed that respondents often utilized GitHub Copilot in programming-related tasks, particularly when encountering coding errors and debugging code, with an overall weighted mean of 3.94 interpreted as "Often." The respondents also demonstrated a proficient level of perceived improvement in debugging skills, with a grand mean of 3.00 interpreted as "Proficient." Correlation analysis further revealed a strong positive and statistically significant relationship between GitHub Copilot utilization and perceived improvement in debugging skills ($r = 0.6094$, $p < .001$), indicating that increased utilization of the tool is associated with higher perceived debugging proficiency. The study concludes that GitHub Copilot can serve as an effective AI-assisted programming tool that supports debugging and problem-solving activities while highlighting the importance of maintaining critical thinking and independent coding skills in programming education.

Keywords: AI-assisted Programming, Debugging Proficiency, GitHub Copilot, Programming Education, Artificial Intelligence

INTRODUCTION

In this modern technology-based world, digital technologies are widely utilized to support teaching and learning. Artificial Intelligence (AI) encompasses a range of advanced computer systems that mimic human intelligence to assist with learning, problem-solving, and programming, using these tools it can generate, debug, and explain code (Mae et al., 2025). AI integration enhances productivity and quality within software development, particularly for code generation and bug detection (Daniel Ajiga et al., 2024). Generative AI supports software engineering by automating software development tools and processes, which saves time and resources (Sauvola et al., 2024).

Artificial intelligence (AI) coding assistants, such as GitHub Copilot, function as AI pair programmers that use machine learning and natural language processing to synthesize code from natural language explanations (Sarah Nadi, 2022). GitHub Copilot is frequently used by students to generate code and to repeatedly fix code or generate new code when it does not work, suggesting that students trust it to better understand code than to generate it (Shah et al., 2025) Additionally, recent studies underscore an increasing reliance on AI assistants among Information Technology undergraduates, with the majority of third-year students using these tools for core programming tasks (Mae et al., 2025). While platforms like GitHub Copilot are recognized for enhancing productivity and improving debugging efficiency, research underscores a concurrent hindrance in conceptual learning, emphasizing a strategic and deliberate approach to integrating AI tools within the educational framework (Kumar Gupta, 2025).

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

While AI-assisted programming tools like GitHub Copilot are increasingly utilized in both professional software development and academic settings, existing literature primarily evaluates their efficacy through technical performance metrics such as productivity, code accuracy, and task efficiency (Sarah Nadi, 2022; Zhang et al., 2023). Although empirical data suggest that AI integration facilitates faster debugging and higher task completion rates, concerns persist regarding student over-reliance and a potential reduction in the conceptual depth required for independent problem-solving (Zviel-Girshin, 2024). Although previous studies have explored GitHub Copilot usage and debugging effectiveness separately, limited research has focused on the relationship between Copilot utilization and students' perceived improvement in debugging skills.

To address this gap, the study aims to examine the utilization of GitHub Copilot and its relationship with perceived improvement in debugging skills among BSIT students at Quezon City University, focusing on usage factors such as frequency of use and time spent using the tool in relation to students' perceived ability to detect and resolve coding errors.

The findings of this study may enhance students' understanding of how GitHub Copilot usage influences their perceived improvement in debugging skills, thereby supporting the effective integration of AI-assisted tools in programming education. Moreover, the results might also be helpful for professors, as they could give useful insights for designing teaching strategies that successfully balance the assistance of AI with the progress of independent debugging and problem-solving skills. This is particularly significant when considering concerns of over-reliance and diminished independent exercise (Mae et al., 2025). Overall, the study aims to establish a basis for understanding the role of GitHub Copilot in improving students' debugging skills.

Statement of the Problem

The purpose of this study is to determine the utilization of GitHub Copilot and its relationship to the perceived improvement in debugging skills among college students at Quezon City University. The utilization of GitHub Copilot is examined in terms of frequency of use and time spent using the tool, while perceived improvement in debugging skills is evaluated based on students' ability to identify code errors, fix bugs, and understand code issues.

Specifically, this study seeks to answer the following questions:

1. How do the respondents be described in the utilization of GitHub Copilot in terms of:

a. Frequency of GitHub Copilot usage; and

b. Time spent using GitHub Copilot?

2. What is the level of students' perceived improvement in debugging skills in terms of:

a. Identifying code errors

b. Fixing bugs

c. Understanding code issues

3. Is there a significant relationship between GitHub Copilot usage and students' perceived improvement in debugging skills?

Related Studies

A structured debugging course suggests that students can significantly improve their debugging performance and confidence through systematic training (Wilkin, 2025). Additionally, research on AI-supported debugging environments indicates that students engage in problemsolving by interacting with AI-generated suggestions and perceive these interactions as beneficial to their learning (Yang et al., 2024). These findings collectively indicate

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

that structured instruction and AI-assisted support have distinct effects on students' development of debugging skills. Structured training enhances performance and confidence, while AI support fosters engagement and perceived learning experiences.

There have been many studies which show that students tend to have excessive confidence in their own coding results, even when they do not perform well. This is evident in the weak or moderate correlation between students' self-confidence and their actual performance, suggesting that many learners tend to overestimate the accuracy of their solutions (Strickroth, 2024). The use of the artificial intelligence tool GitHub Copilot, on the other hand, has been shown to improve programming productivity, with users completing tasks significantly faster when assisted by the tool (Peng et al., 2023). However, these improvements are primarily measured in terms of speed and do not necessarily reflect code accuracy or debugging ability. Additionally, not all outputs generated by Copilot are correct and still require proper evaluation (Wermelinger, 2023). This suggests that the increased efficiency and confidence students may experience when using such tools do not necessarily translate to improved debugging skills.

Students view AI tutors as helpful tools for learning how to debug programs; however, interaction with these systems may lead to increased dependence on AI-generated suggestions during the troubleshooting process (Yang et al., 2024). Similarly, studies on GitHub Copilot indicate that it can support students in understanding and applying programming concepts by generating potential solutions to coding problems. However, the effectiveness of these tools depends on the students' ability to critically evaluate, modify, and refine the generated outputs (Wermelinger, 2023). Overall, the literature suggests that while AI tools such as GitHub Copilot can enhance students' learning experiences, the development of debugging skills still relies on their capacity for critical thinking and independent problem-solving.

While many studies have emerged regarding the role of GitHub Copilot in programming learning, they have mostly focused on students' perceptions, coding assistance, and interactions with the tool (Wermelinger, 2023). These studies suggest that students engage with Copilot in different ways, but they rarely explore how variations in its utilization may influence learning outcomes. Students interact with AI tools such as GitHub Copilot at varying levels, particularly in terms of how frequently they use the tool and how much time they spend engaging with it. However, these studies rarely examine how the utilization of GitHub Copilot—particularly in terms of frequency of use and time spent using the tool—relates to students' perceived improvement in debugging skills.

This highlights a gap in the literature on the utilization of GitHub Copilot and students' perceived improvement in debugging skills. Addressing this gap is important for understanding whether variations in the utilization of GitHub Copilot (in terms of frequency of use and time spent using the tool) are associated with variations in students' self-perceived improvement in debugging among Bachelor of Science in Information Technology students at Quezon City University

DESIGN AND METHODOLOGY

Research Design

This study will utilize a descriptive-correlational research design to examine the utilization of GitHub Copilot and its relationship with the perceived improvement in debugging skills among Bachelor of Science in Information Technology (BSIT) students at Quezon City University. The descriptive approach will be employed to determine the level of GitHub Copilot utilization in terms of frequency and duration of use. It will also assess the respondents' perceived improvement in debugging skills, particularly in identifying code errors, fixing bugs, and understanding code issues. Meanwhile, the correlational approach will be used to determine whether a significant relationship exists between GitHub Copilot utilization and the respondents' perceived improvement in debugging skills.

This research design is appropriate for the study because it does not involve the manipulation of variables and focuses on examining existing conditions and relationships within a natural academic environment.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

Data Gathering

Data for this study will be collected through a structured online questionnaire administered using Google Forms, which will serve as the primary research instrument. The questionnaire is designed to gather relevant information regarding the respondents' utilization of GitHub Copilot and its perceived impact on their debugging skills. To ensure the collection of accurate and relevant data, the questionnaire is divided into three major sections.

The first section, Screening Profile, is intended to determine the eligibility of the respondents for participation in the study. Specifically, this section confirms whether the participants have prior experience using GitHub Copilot in programming or coding-related tasks. Only individuals who have actual experience utilizing GitHub Copilot for coding, debugging, software development, or related activities will be included in the study. Respondents who have not previously used the tool will be excluded to ensure that the gathered data accurately reflects the utilization and effectiveness of GitHub Copilot among actual users.

The second section focuses on the Utilization of GitHub Copilot. This part of the questionnaire aims to measure the extent to which respondents use GitHub Copilot in their programming-related activities. It includes Likert-scale items that assess both the frequency of use and the amount of time respondents spend utilizing the tool. The questions are designed to determine how often participants rely on GitHub Copilot during coding, debugging, problem-solving, and software development tasks, as well as the degree to which the tool is integrated into their programming practices.

The third section, Perceived Improvement in Debugging Skills, evaluates the respondents' perceptions regarding the influence of GitHub Copilot on their debugging abilities. This section also utilizes Likert-scale questions to assess how the use of GitHub Copilot contributes to improvements in identifying, analyzing, and resolving programming errors. Furthermore, it examines whether the tool assists respondents in troubleshooting code more efficiently, enhancing problem-solving strategies, and improving overall debugging performance during software development activities.

Sampling Technique and Ethical Considerations

A purposive sampling technique was employed in selecting the respondents for this study. The selection of participants was based on their prior experience in using GitHub Copilot and their willingness and availability to participate in the research process. The study involved a population size of 150 respondents, composed of Bachelor of Science in Information Technology students from Quezon City University who had previously utilized GitHub Copilot for coding, programming, debugging, or other software development activities. This sampling approach ensured that only individuals with relevant experience using the tool were included in the study, thereby increasing the reliability and relevance of the collected data.

Prior to participation, all respondents were informed about the purpose and objectives of the study. The researchers assured the participants that their responses would be treated with strict confidentiality and anonymity, and that participation in the study was entirely voluntary. Data collection was conducted through a Google Forms questionnaire, allowing respondents to conveniently answer the survey online. Participants were also given the option to provide their names if they wished to do so; however, no restrictions, obligations, or pressure were imposed on them regarding the disclosure of their identity.

Statistical Treatment of Data

The data gathered in this study will be analyzed using appropriate statistical tools to address the research questions concerning the utilization of GitHub Copilot and the students' perceived improvement in debugging skills among Bachelor of Science in Information Technology students at Quezon City University. The statistical treatments employed in the study are discussed as follows:

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

Weigthed Mean

The weighted mean will be utilized to determine the level of GitHub Copilot utilization among the respondents in terms of frequency of use and time spent using the tool for programming-related activities. Furthermore, it will also be used to assess the level of students' perceived improvement in debugging skills, particularly in identifying code errors, fixing bugs, troubleshooting, and understanding programming issues. The weighted mean is appropriate for this study since the questionnaire uses Likert-scale responses that measure the degree of agreement and frequency of utilization.

The following interpretive scale will be used to analyze the respondents' level of GitHub Copilot utilization based on the 5-point Likert scale:

Table 1. 5-Point Likert Scale Used by the Researchers

Range	**Verbal Interpretation**
4.21 – 5.00	Always (A)
3.41 – 4.20	Often (O)
2.61 – 3.40	Sometimes (S)
1.81 – 2.60	Rarely (R)
1.00 – 1.80	Never (N)

Meanwhile, the following interpretive scale will be used to determine the respondents' perceived improvement in debugging skills using the 4-point scale:

Table 2. 4-Point Likert Scale Used to Determibe Respondent's Perceived Improvement in Debugging Skills

Range	**Verbal Interpretation**
3.26 – 5.00	Always (A)
2.51 – 3.25	Sometimes (S)
1.76 – 2.50	Rarely (R)
1.00 – 1.75	Never (N)

The computed weighted mean scores will provide a quantitative basis for interpreting the respondents' utilization of GitHub Copilot and their perceived debugging proficiency.

RESULT AND DISCUSSION

This section presents and discusses the data gathered from 150 Bachelor of Science in Information Technology students at Quezon City University regarding their utilization of GitHub Copilot and their perceived improvement in debugging skills. The presentation of results focuses on the extent of GitHub Copilot utilization, the respondents' self-assessed debugging proficiency, and the relationship between the use of the tool and the enhancement of debugging skills.

Utilization of Github Copilot

The utilization of GitHub Copilot was assessed based on the frequency with which the respondents used the tool during various programming-related activities, including debugging, solving logical problems, and writing code.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

Table 3. Weighted Mean and Interpretation of GitHub Copilot Utilization in Terms of Frequency of Use

Indicators	**Weighted Mean**	**Interpretation**
Use when encountering coding errors	4.11	Often
Use when debugging code	4.01	Often
Use when solving problems/logic	3.96	Often
Use when working on programming activities	3.88	Often
Use when writing new code from scratch	3.76	Often
Category Mean	**3.94**	**Often**

Table 1 shows that the respondents often utilize GitHub Copilot in different programming-related tasks, as reflected by the overall category mean of 3.94, interpreted as "Often." This finding indicates that GitHub Copilot has become a commonly used support tool among the respondents during software development and debugging activities.

Among the indicators, the statement "Use when encountering coding errors" obtained the highest weighted mean of 4.11, suggesting that respondents primarily rely on GitHub Copilot when troubleshooting and resolving coding-related issues. This result implies that students perceive the tool as particularly helpful in identifying possible solutions and guiding them when they encounter programming errors.

Similarly, the indicators "Use when debugging code" (4.01) and "Use when solving problems/logic" (3.96) also received high weighted means, indicating that respondents frequently integrate GitHub Copilot into their debugging and problem-solving processes. These findings suggest that the tool is not only used for code generation but also serves as an aid in analyzing logical structures and improving coding efficiency.

Meanwhile, the indicator "Use when working on programming activities" obtained a weighted mean of 3.88, which further indicates that respondents often use GitHub Copilot as a supplementary learning and development tool during programming exercises and academic coding tasks. On the other hand, "Use when writing new code from scratch" garnered the lowest weighted mean of 3.76, although still interpreted as "Often." This may imply that while students use GitHub Copilot to assist in code generation, they may still prefer to manually construct programs and use the tool mainly for support, verification, or enhancement purposes rather than complete code dependency.

Overall, the findings reveal that GitHub Copilot is frequently utilized by the respondents, particularly in debugging and troubleshooting situations, highlighting its perceived usefulness in programming-related activities.

Perceived Improvement in Debugging Skills

The study also assessed the respondents' perceived improvement in debugging skills in terms of understanding code issues, fixing bugs, and identifying code errors.

As presented in Table 4 below, the respondents demonstrated a "Proficient" level of perceived improvement in debugging skills, with an overall grand mean of 3.00. This finding suggests that the respondents believe that their use of GitHub Copilot contributes positively to the enhancement of their debugging abilities.

Table 4. Summary of Weighted Means for Perceived Improvement in Debugging Skills

Variable Category	**Weighted Mean**	**Interpretation**
Understanding Code Issues	3.05	Proficient
Fixing Bugs	2.99	Proficient
Identifying Code Errors	2.96	Proficient
Grand Mean	**3.00**	**Proficient**

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

Among the three dimensions, "Understanding Code Issues" obtained the highest weighted mean of 3.05, interpreted as "Proficient." This indicates that respondents perceive GitHub Copilot as beneficial in helping them analyze, interpret, and understand programming problems and code-related issues more effectively. The tool may provide explanations, suggestions, or alternative coding approaches that assist students in comprehending the root causes of errors.

The indicator "Fixing Bugs" yielded a weighted mean of 2.99, while "Identifying Code Errors" obtained a weighted mean of 2.96, both verbally interpreted as "Proficient." These findings imply that respondents perceive improvement in their ability to locate and resolve programming errors with the assistance of GitHub Copilot. The relatively close values among the three dimensions also indicate that the respondents consistently perceive the tool as supportive across different aspects of debugging.

Overall, the results suggest that GitHub Copilot contributes positively to the respondents' debugging proficiency by assisting them in understanding programming issues, identifying errors, and implementing appropriate fixes during coding activities.

Relationship Between GitHub Copilot Utilization and Perceived Improvement in Debugging Skills

To determine whether a significant relationship exists between GitHub Copilot utilization and students' perceived improvement in debugging skills, the Pearson Product-Moment Correlation Coefficient (Pearson r) was computed and tested at the 0.05 level of significance.

Table 5. Correlation Between GitHub Copilot Utilization and Debugging Skills

Variables	Pearson r	p-value	Strength	Statistical Decision
Utilization vs. Debugging Skills	0.6094	< .001	Strong	Significant Relationship

Table 3 presents the correlation analysis between GitHub Copilot utilization and the respondents' perceived improvement in debugging skills. The computed Pearson r coefficient of 0.6094 indicates a strong positive relationship between the two variables. This means that higher levels of GitHub Copilot utilization are associated with higher levels of perceived debugging proficiency among the respondents.

Furthermore, the obtained p-value of less than .001 is lower than the 0.05 level of significance, leading to the rejection of the null hypothesis. This result confirms that the relationship between GitHub Copilot utilization and debugging skills is statistically significant.

The findings imply that students who frequently utilize GitHub Copilot during programming activities tend to perceive greater improvement in their debugging abilities. The strong positive relationship also suggests that the tool may serve as an effective support mechanism in enhancing students' understanding of code issues, error detection, and bug resolution processes.

Overall, the results of the study indicate that GitHub Copilot is not only widely utilized by the respondents but is also significantly associated with the improvement of their perceived debugging proficiency.

CONCLUSION

This study examined the utilization of GitHub Copilot and its relationship with the perceived improvement in debugging skills among Bachelor of Science in Information Technology students at Quezon City University. Specifically, the study focused on the respondents' frequency of use and time spent using GitHub Copilot, as well as their perceived improvement in identifying code errors, fixing bugs, and understanding code issues.

Based on the findings of the study, the respondents were found to often utilize GitHub Copilot in various programming-related activities. The results revealed that students primarily use the tool when encountering

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

coding errors and during debugging tasks, indicating that GitHub Copilot serves as a significant support mechanism in troubleshooting and problem-solving activities. The respondents also frequently utilized the tool during programming exercises and logic-related tasks, suggesting that GitHub Copilot has become integrated into their coding practices as both a development and learning aid.

In terms of perceived improvement in debugging skills, the respondents demonstrated a proficient level across all measured dimensions, including identifying code errors, fixing bugs, and understanding code issues. Among these dimensions, understanding code issues obtained the highest mean, indicating that students perceive GitHub Copilot as particularly helpful in analyzing and interpreting programming problems. These findings suggest that AI-assisted tools may contribute positively to students' confidence and perceived competence in debugging-related activities.

Furthermore, the study established a statistically significant strong positive relationship between GitHub Copilot utilization and perceived improvement in debugging skills. The results imply that students who frequently utilize GitHub Copilot and spend more time engaging with the tool tend to report higher levels of perceived debugging proficiency. Consequently, the null hypothesis stating that there is no significant relationship between GitHub Copilot utilization and perceived improvement in debugging skills was rejected.

The findings of the study support existing literature emphasizing the potential benefits of AI-assisted programming tools in enhancing students' learning experiences and programming productivity. However, the study also reinforces the importance of critical thinking and independent problem-solving, as discussed in previous studies, since AI-generated outputs still require evaluation, verification, and refinement by the users. While GitHub Copilot may serve as an effective supplementary tool for debugging and programming assistance, the development of genuine debugging competence still depends on the students' active engagement, analytical thinking, and understanding of programming concepts.

Overall, the study concludes that GitHub Copilot is widely utilized among BSIT students and is significantly associated with students' perceived improvement in debugging skills. The findings highlight the potential of AI-assisted programming tools to support programming education when used appropriately and strategically. Moreover, the study provides valuable insights for educators and academic institutions regarding the integration of AI tools into programming instruction while maintaining a balance between technological assistance and the development of independent debugging and problem-solving skills.

REFERENCES

1. Daniel Ajiga, Patrick Azuka Okeleke, Samuel Olaoluwa Folorunsho, & Chinedu Ezeigweneme. (2024). Enhancing software development practices with AI insights in high-tech companies. Computer Science & IT Research Journal, 5(8), 1897–1919. https://doi.org/10.51594/csitrj.v5i8.1450
2. Kumar Gupta, M. A. (2025). IMPACT OF GITHUB COPILOT USAGE ON PROGRAMMING PRODUCTIVITY AMONG UNDERGRADUATE COMPUTER SCIENCE STUDENTS. Asian And Pacific Economic Review, 18(2), 1. https://doi.org/10.65985/APER.2026632886
3. Mae, J., Boitizon, G., Barte, B. P., Layco, J. P., Nicole, A., Zoleta, J., & Balmes, I. L. (2025). The Influence of AI Code Assistants on Programming Learning: A Descriptive Study of Student Dependence.
4. Peng, S., Kalliamvakou, E., Cihon, P., & Demirer, M. (2023). The Impact of AI on Developer Productivity: Evidence from GitHub Copilot. http://arxiv.org/abs/2302.06590
5. Sarah Nadi, N. N. (2022). An empirical evaluation of GitHub copilot's code suggestions. Association for Computing Machinery.
6. Sauvola, J., Tarkoma, S., Klemettinen, M., Riekki, J., & Doermann, D. (2024). Future of software development with generative AI. Automated Software Engineering, 31(1). https://doi.org/10.1007/s10515-024-00426-z

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

7. Shah, A., Chernova, A., Tomson, E., Porter, L., Griswold, W. G., & Soosai Raj, A. G. (2025). Students' Use of GitHub Copilot for Working with Large Code Bases. SIGCSE TS 2025 - Proceedings of the 56th ACM Technical Symposium on Computer Science Education, 1, 1050–1056. https://doi.org/10.1145/3641554.3701800
8. Strickroth, S. (2024). Exploring Students' Self-Confidence in Their Programming Solutions. Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE, 1, 415–421. https://doi.org/10.1145/3649217.3653589
9. Wermelinger, M. (2023). Using GitHub Copilot to Solve Simple Programming Problems. SIGCSE 2023 - Proceedings of the 54th ACM Technical Symposium on Computer Science Education, 1, 172–178. https://doi.org/10.1145/3545945.3569830
10. Wilkin, G. A. (2025). "Debugging: From Art to Science" A Case Study on a Debugging Course and Its Impact on Student Performance and Confidence. SIGCSE TS 2025 - Proceedings of the 56th ACM Technical Symposium on Computer Science Education, 1, 1225–1231. https://doi.org/10.1145/3641554.3701893
11. Yang, S., Zhao, H., Xu, Y., Brennan, K., & Schneider, B. (2024). Debugging with an AI Tutor: Investigating Novice Help-seeking Behaviors and Perceived Learning. ICER 2024 - ACM Conference on International Computing Education Research, 1, 84–94. https://doi.org/10.1145/3632620.3671092
12. Zhang, B., Liang, P., Zhou, X., Ahmad, A., & Waseem, M. (2023). Practices and Challenges of Using GitHub Copilot: An Empirical Study. Proceedings of the International Conference on Software Engineering and Knowledge Engineering, SEKE, 2023-July, 124–129. https://doi.org/10.18293/SEKE2023-077
13. Zviel-Girshin, R. (2024). The Good and Bad of AI Tools in Novice Programming Education. Education Sciences, 14(10). https://doi.org/10.3390/educsci14101089