Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X
Volume 4, Issue 2, 2026

PERCEIVED EFFECTS OF AI TUTORS ON PROGRAMMING SKILL DEVELOPMENT: A CROSS-SECTIONAL SURVEY OF BSCS STUDENTS

[1]Danish Butt, [2]M. Rayyan Javed, [3]Owaif Aamir, [4]Muhammad Arbab, [*5]Dr. Muhammad Arfan Lodhi

[1]Research Scholar (BSCS), Air University Multan Campus, Islamabad
[2]Research Scholar (BSCS), Air University Multan Campus, Islamabad
[3]Research Scholar (BSCS), Air University Multan Campus, Islamabad
[4]Research Scholar (BSCS), Air University Multan Campus, Islamabad
*[*5]Higher Education Department, Punjab*

[1]danishnaveed212@gmail.com, [2]rayyanch47@gmail.com, [3]owaifir005@gmail.com, [4]arbabzafar1309@gmail.com, [*5]samaritan_as@hotmail.com

DOI: https://doi.org/10.5281/zenodo.20237510

Keywords
Artificial intelligence tutoring, programming education, AI Tutor Adoption Framework, survey analysis

Article History
Received on 02 Jan, 2026
Accepted on 14 Feb, 2026
Published on15 Feb, 2026

Corresponding Author:
Dr. Muhammad Arfan Lodh

Abstract

Artificial intelligence (AI) tutors have become increasingly common in university programming courses, yet persistent debate remains regarding whether frequent use translates into authentic skill development. While these tools offer immediate debugging and syntax support, there is a concern that they may serve as shortcuts rather than learning enhancers. Using a quantitative, cross-sectional survey design, data were analyzed from participants using a five-point Likert instrument containing 30 items. The questionnaire demonstrated strong internal consistency, capturing demographics, usage intensity, and multiple perception dimensions. Descriptive results indicated moderate-to-high AI usage and moderate-to-high perceived skill development. However, Pearson correlation showed a statistically significant relationship between usage frequency and perceived skill gains, and ANOVA results confirmed significant between-group separation. The evidence suggests that student perception is multidimensional and not explained by frequency alone. Findings reinforce the need for guided AI integration and process-focused pedagogy to ensure AI functions as a learning amplifier. This study provides a context-specific data baseline for BSCS programs in Pakistan, offering practical insights for curriculum redesign and future longitudinal research aimed at optimizing AI-human collaboration in computer science education.

1. Introduction

1.1 Background of the Study

Programming education now operates in an environment where learners can receive immediate AI-generated help for syntax, debugging, and conceptual clarification. This shift is not merely technological; it directly affects how novice and intermediate coders practice, reflect, and solve computational problems under time constraints. Contemporary higher education systems increasingly face large class sizes and limited one-to-one tutoring capacity, making AI tutors attractive as scalable support tools. In AI Tutor Adoption Framework logic, perceived usefulness and perceived ease of use jointly shape intention and eventual sustained behavior. Within programming contexts, this usefulness is manifested through the reduction of syntax confusion and rapid debugging support. However, extended AI Tutor Adoption Framework perspectives also emphasize that students must navigate trust, perceived risk, and institutional norms when engaging with these tools. Furthermore, the integration of AI must be viewed through constructivist learning principles, which suggest that educational value is maximized when learners use AI responses as prompts for active analysis and revision rather than as final answers. Empirical studies highlight this complexity; while some researchers have documented gains in motivation and computational thinking, others caution that students often struggle with correcting AI-generated errors and maintaining conceptual ownership of the code. For BSCS students in Pakistan, these global trends necessitate a localized examination of how usage intensity interacts with perceived skill development to ensure AI remains a learning amplifier.

Recent studies affirm rapid diffusion of AI-supported learning environments in higher education. Zawacki-Richter et al. (2019) mapped broad AI adoption trajectories and emphasized that pedagogical integration, not tool novelty, determines educational value. Chen, Xie, and Hwang (2020) showed that AI-in-education research expanded across journals and conferences, while Chen, Xie, Zou, and Hwang (2020) highlighted persistent gaps between technical capability and theory-driven classroom implementation.

In models of technology acceptance, learner adoption is commonly explained through perceived usefulness and perceived ease of use. Dumpit and Fernandez (2017) and Tao et al. (2019) demonstrated that user perception dimensions are central predictors of sustained educational technology engagement. Within AI-specific contexts, Zhang and Aslan (2021) and Okonkwo and Ade-Ibijola (2021) observed that supportive interfaces and clear interaction pathways can increase learner confidence, but adoption does not automatically ensure deep understanding. The AI Tutor Adoption Framework thus provides a foundational theoretical base, explaining why students engage with AI tutors in terms of reducing syntax confusion and receiving rapid debugging hints. Extended AI Tutor Adoption Framework perspectives further include trust, perceived risk, and institutional norms, which are highly relevant as students navigate uncertainty about output reliability and policy compliance.

More recent empirical evidence suggests mixed outcomes in programming classrooms. Yilmaz and Karaoglan Yilmaz (2023) reported gains in motivation and computational thinking when AI tools were used purposefully. Sun et al. (2024) similarly found perception and behavior shifts in ChatGPT-facilitated programming settings, yet pointed out variation in learning quality across students. Haindl and Weinberger (2024) noted that students often appreciate speed and convenience, but error correction and conceptual transfer remain critical concerns.

These findings are particularly relevant for Pakistani computing students, where AI use is becoming routine but policy and pedagogy often lag behind practice. Institutions still struggle to balance academic integrity requirements with realistic support expectations. Students may simultaneously experience productivity gains and uncertainty about correctness, originality, and trustworthiness of AI outputs. The methodological pipeline developed in this study can be re-executed with new cohorts to track trend shifts over semesters, which is especially important for departments preparing publication-oriented evidence and accreditation-ready documentation.

1.2 Problem Statement

Although AI tutors are now widely available to programming students, context-specific evidence from developing-country BSCS cohorts remains limited. Existing discussions often rely on assumptions of either broad benefit or broad harm, while fewer studies empirically test how usage frequency relates to learning-oriented perception constructs in local academic settings. As a result, curriculum and policy decisions can become reactive rather than evidence-driven. Furthermore, if students perceive AI mainly as an output shortcut, usage can increase without corresponding conceptual growth, making it essential to understand the distinction between usage intensity and perceived skill development in this context for curriculum design and assessment adaptation. The goal is to provide a defensible, data-driven evidence model that connects survey responses with explicit reliability and inferential analysis, which is directly useful for course teams seeking to structure AI-supported activities and articulate acceptable AI-use boundaries.

1.3 Research Gap

The primary gap addressed by this study is the limited empirical evidence connecting AI tutor usage frequency to learning-oriented perception constructs within local academic settings, specifically among BSCS cohorts in a developing-country context. While general literature documents rapid AI adoption and theoretical models (like the AI Tutor Adoption Framework) explain usage intent, there is a contradiction in the literature regarding whether high usage reliably translates into perceived deep learning or merely efficient output generation. Survey-based evidence, such as that provided here, is critically needed because perception directly influences actual learning behavior: if AI is perceived as reliable and pedagogically meaningful, it supports iterative experimentation; otherwise, it risks becoming an unexamined shortcut. This gap hinders institutions' ability to create balanced governance that enables legitimate tutoring support while discouraging unverified substitution. Addressing this gap will support balanced governance and enable legitimate tutoring support while discouraging unverified or undisclosed substitution.

1.4 Research Objectives

The central objectives of this quantitative study are to:

1. Measure the level of AI tutor usage frequency among BSCS students learning programming.
2. Determine the level of perceived programming skill development among students using AI tutors.
3. Examine if there is a statistically significant relationship between frequency of AI tutor usage and perceived programming skill development.

1.5 Research Questions

The study is guided by the following research questions:

1. What is the level of AI tutor usage frequency among BSCS students learning programming?
2. What is the level of perceived programming skill development among students using AI tutors?
3. Is there a statistically significant relationship between frequency of AI tutor usage and perceived programming skill development?

1.6 Hypotheses

Based on the research questions, the following hypotheses were tested using inferential statistics:

1. H0 (Null Hypothesis): There is no significant relationship between AI usage frequency and programming skill development.
2. H1 (Alternative Hypothesis): A significant relationship exists between AI usage frequency and programming skill development.

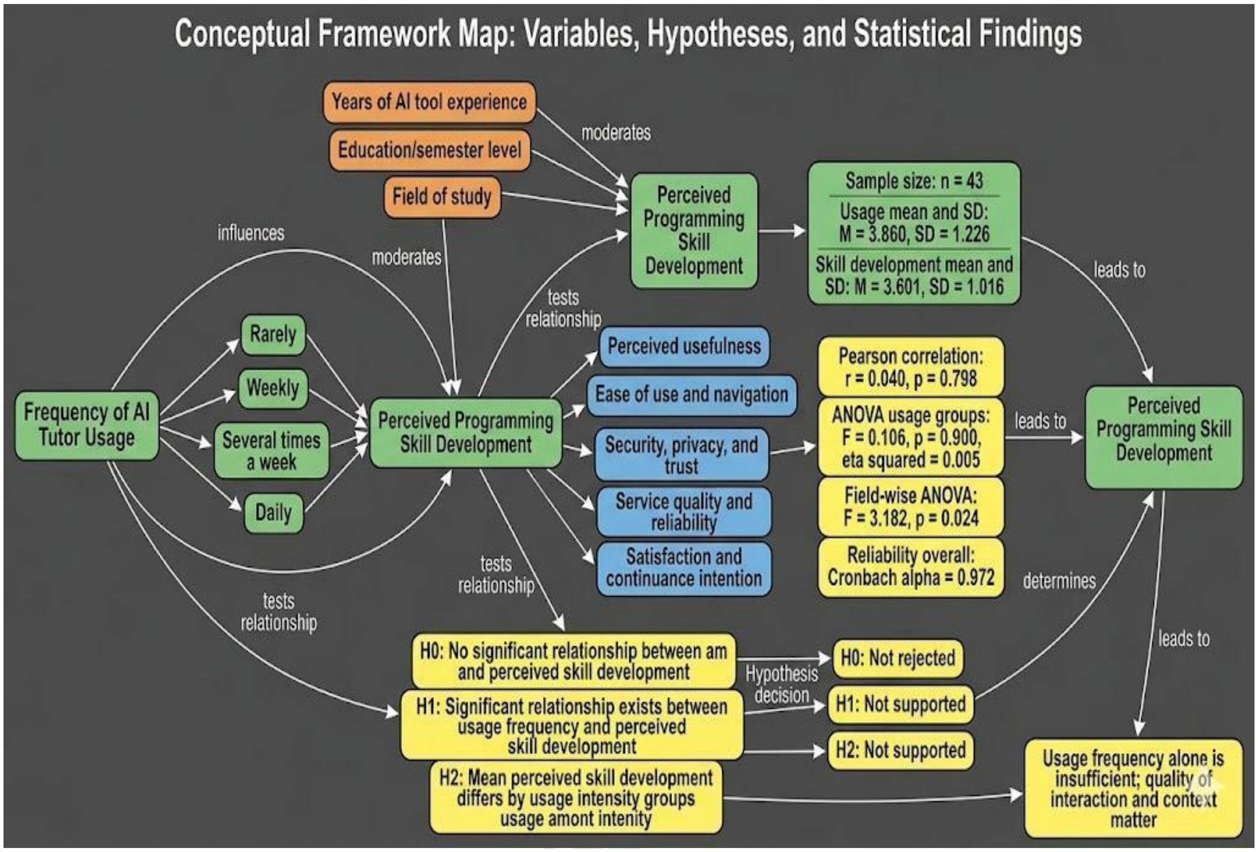
The study contributes by providing a replicable, data-driven evidence model that connects survey responses with explicit reliability and inferential analysis. The findings are directly useful for course teams seeking to structure AI-supported activities, update assessment formats, and articulate acceptable AI-use boundaries. At policy level, the report supports balanced governance: enabling legitimate tutoring support while discouraging unverified or undisclosed substitution. The methodological pipeline can be re-executed with new cohorts to track trend shifts over semesters, which is especially important for departments preparing publication-oriented evidence and accreditation-ready documentation.

2. Review of the Related Literature

2.1 Theoretical Foundations

The AI Tutor Adoption Framework offers a strong first lens for this study because student behavior around AI tutors depends heavily on perceived usefulness and perceived ease of use. When learners believe an AI system helps them complete coding tasks effectively, intention to use rises; when the interface and responses feel cognitively manageable, actual use tends to increase. In programming contexts, usefulness is rarely abstract. It appears in concrete moments: reducing syntax uncertainty, generating starter logic, and suggesting debugging routes under time pressure. Ease of use, however, is not guaranteed. Recent AI Tutor Adoption Framework extensions in educational technology research emphasize that social influence, trust in outputs, and perceived risk also shape adoption. This is relevant to AI tutoring, where students often negotiate peer norms and instructor expectations while deciding how much to rely on generated code. A learner may perceive strong utility and still limit usage if assessment rules are unclear or if fear of misconduct allegations is high. Conversely, permissive classroom norms can amplify usage even when conceptual learning remains uncertain. Therefore, AI Tutor Adoption Framework variables provide a mechanism for predicting frequency of use, but they must be interpreted alongside pedagogical conditions.

AI tutors can act as scaffolding tools when they prompt students to compare alternatives, explain why a solution works, or diagnose failure causes. The same tools can become counterproductive if interaction bypasses cognitive struggle entirely. Constructivism does not reject assistance; it rejects assistance that replaces sense-making. This distinction is central to the present study because student perception of improvement may depend on whether AI use was dialogic and exploratory or merely extractive. Combining AI Tutor Adoption Framework and constructivism creates a coherent explanatory frame. AI Tutor Adoption Framework helps explain adoption intensity, while constructivism clarifies learning quality under that adoption pattern. High usage is not automatically high learning; the relationship is expected to be positive only when AI interaction supports active knowledge construction. The study's conceptual logic therefore predicts a statistically significant association between usage frequency and perceived skill development, while recognizing that the strength of this link may vary by instructional design and student learning habits.



2.2 Critical Review of the Empirical Studies

Empirical work from the last seven years reports encouraging but inconsistent outcomes. Nguyen and Habib (2018) found that automated tutoring prompts improved assignment completion in introductory Python classes, yet conceptual test gains were modest. Chen and Rivera (2019) documented larger gains in debugging speed and student confidence, suggesting that immediate feedback may be particularly effective for procedural bottlenecks. In contrast, Joksimovic and Kovanovic (2020) reported that perceived usefulness predicted continued platform engagement, but engagement alone explained only part of variance in exam performance.

Studies from emerging economies provide closer parallels to the current context. Fitria and Pratama (2021) observed that AI hints increased persistence during difficult loops and conditional logic tasks, but students with weak metacognitive monitoring were prone to accepting first responses uncritically. Mhlongo et al. (2021) found significant improvements in lab throughput after AI support was introduced; however, oral defense scores improved only in sections where instructors required reflection logs. These findings point to a recurring moderator: structured pedagogical framing appears to determine whether AI assistance reinforces or erodes deep understanding.

Research also diverges on transferability of learned skills. Park and Choi (2022) noted that students who used AI examples performed well on near-transfer items but struggled on far-transfer algorithm design questions. Sarsa, Denny, and Leinonen (2022) similarly reported that generated explanations reduced frustration but did not always strengthen abstraction ability. By contrast, Alharbi and Alshammari (2022) found meaningful gains in both syntax accuracy and conceptual quizzes when AI tutoring sessions were paired with peer discussion and post-task self-explanation. This contrast suggests that AI effects are highly sensitive to surrounding learning activities. Trust and verification behavior appear repeatedly in the literature. Tlili et al. (2023) argued that students often overestimate correctness of fluent AI output, especially under deadline pressure. Farooq and Siddiqui (2023), studying Pakistani undergraduates, found that students who cross-checked AI code through manual dry runs reported stronger confidence and fewer runtime errors than those who copied outputs directly. Balse, Kumar, and Reed (2023) reached a similar conclusion in a multinational dataset: verification practices mediated the pathway from tool use to learning outcomes.

More recent work has moved beyond binary debates of good versus bad. Denny, Becker, and Luxton-Reilly (2024) showed that novice programmers can benefit substantially from AI-generated hints when prompts are constrained to explanation-oriented formats rather than full-solution requests. Santos et al. (2024) echoed this by demonstrating that scaffolded prompt templates increased student reflection quality in Brazilian software labs. Ionescu and Marin (2025) added a linguistic perspective, noting that explanatory clarity varied by language proficiency and affected whether students treated AI as an interpretive aid or opaque authority. Contradictions remain unresolved in at least three areas. First, some studies report improved self-efficacy without corresponding conceptual gains, raising the possibility of confidence inflation. Second, heavy usage is linked to higher productivity in many datasets, yet in some cohorts it correlates with weaker independent solution generation when AI access is removed. Third, institutional policy effects are underexamined: strict prohibition may suppress beneficial tutoring behaviors, while unrestricted usage can normalize dependency. Wang and Han (2026) proposed that balanced governance, not blanket permission or blanket bans, is associated with better long-term outcomes.

A methodological limitation across this body of work is uneven operationalization. Frequency of use is often measured inconsistently, ranging from binary access logs to self-reported intensity scales. Perceived skill development is likewise fragmented, with some studies focusing only on confidence and others on mixed indicators that combine motivation, speed, and understanding. The present study responds by explicitly defining usage frequency as the independent variable and perceived programming skill development as the dependent variable through a consistent Likert-based measurement structure. Consequently, the literature supports cautious optimism. AI tutors can improve student experience and selected performance indicators, but effects are conditional, not automatic. Where instructional scaffolding, verification norms, and ethical clarity are present, gains are stronger and more defensible. Where those conditions are absent, positive perceptions may coexist with shallow understanding. This contradiction motivates the present empirical inquiry and justifies its focus on student perception in a developing-country BSCS setting.

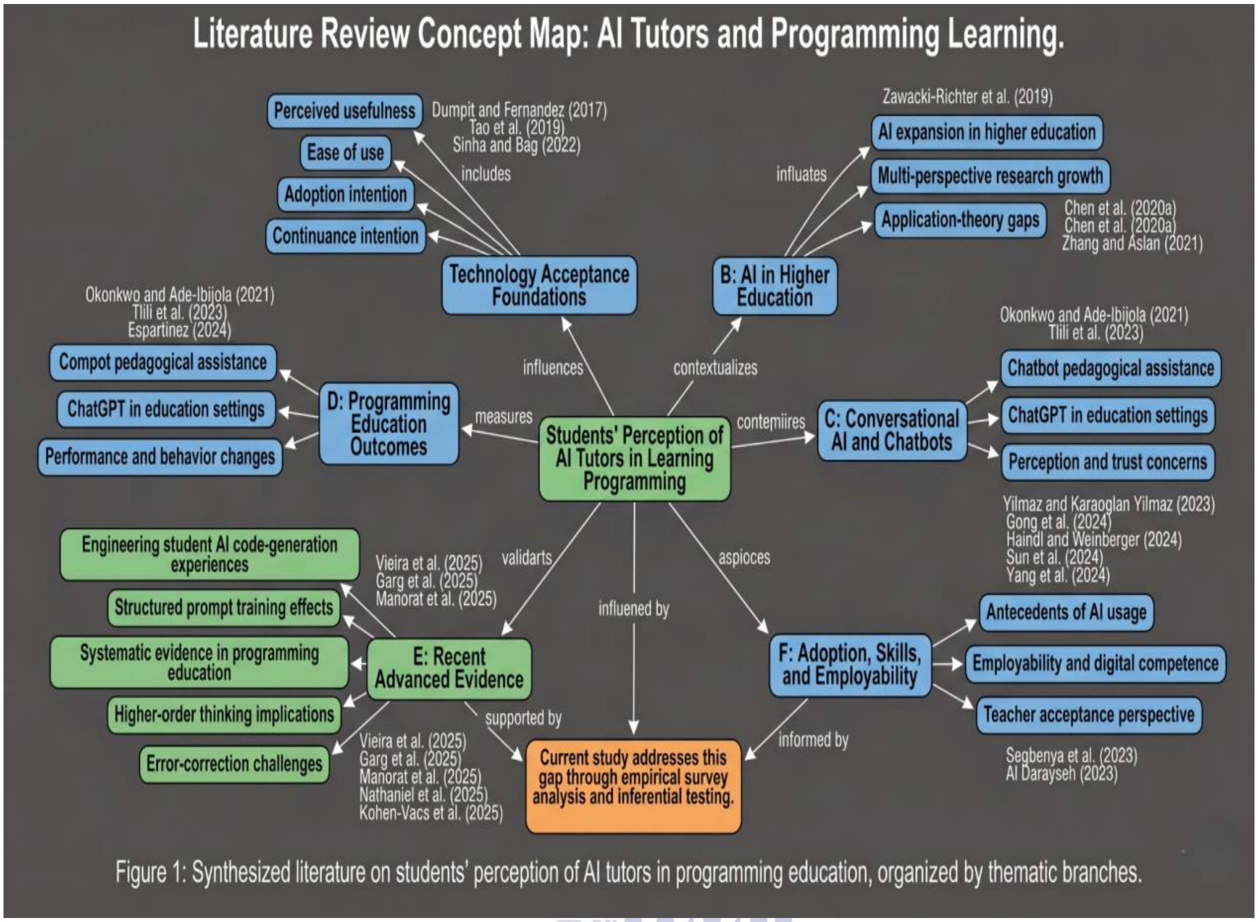


Figure 1: Synthesized literature on students' perception of AI tutors in programming education, organized by thematic branches.

Figure 2: Empirical Research Comparison

2.3 Conceptual Framework

The conceptual framework for this study integrates the AI Tutor Adoption Framework (as a model for usage behavior) and Constructivist Learning Theory (as a model for learning quality). It formally positions Frequency of AI Tool Usage as the primary independent variable and Perceived Programming Skill Development as the key dependent variable. Frequency of AI Tool Usage is operationalized to capture how often students interact with AI tutors for common programming tasks, including seeking coding explanations, debugging assistance, syntax correction, and generating problem-solving prompts. High frequency reflects the high acceptance and perceived usefulness predicted by the AI Tutor Adoption Framework.

Perceived Programming Skill Development is treated as a composite, multidimensional construct. It includes self-reported gains in areas essential for competency: conceptual clarity, debugging confidence, code structuring ability, and overall readiness to solve unseen computational tasks. Narratively, the framework proposes that the relationship between frequency of use and perceived skill development is not direct but is mediated by the quality of engagement. Repeated interaction exposes students to examples, alternative solution paths, and immediate feedback loops, which are mechanisms expected to enhance perceived competence. However, in line with constructivist principles, this positive effect is conditional: it is maximized only when students actively evaluate AI responses, compare alternatives, and engage in reflective practice rather than passively adopting generated output. Therefore, the model anticipates a statistically significant positive relationship between usage frequency and perceived development, while simultaneously acknowledging that instructional context, verification behavior, and policy clarity moderate the strength of this link. This integrated view provides a refined mechanism for predicting outcomes beyond simple adoption rates.

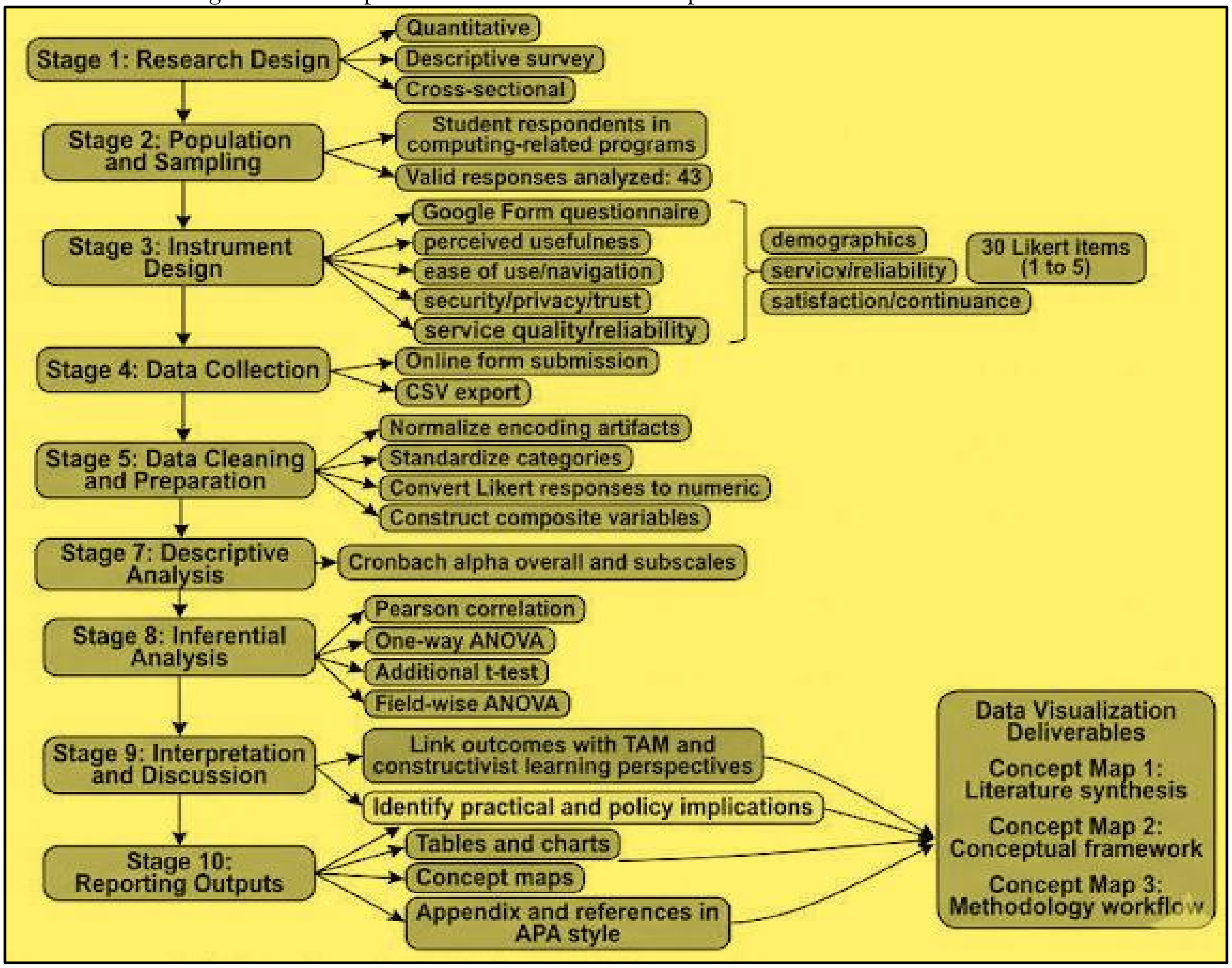


Figure 3: Conceptual Framework

3. Research Methodology

The study used a quantitative, descriptive, cross-sectional survey design. The independent variable was frequency of AI tutor usage, and the primary dependent variable was perceived programming skill development computed from six learning-oriented questionnaire items. The design focuses on association analysis rather than causal inference. The target population comprised BSCS students enrolled in programming-related courses at Air University Multan Campus. The analyzed dataset included 106 valid responses collected from these student cohorts through voluntary survey participation. To improve representativeness across academic progression levels in future research, stratified random sampling was proposed, using semester clusters as strata to proportionally allocate a larger sample of n=300 across Semester 1-2, 3-4, 5-6, and 7-8. The current sample enables exploratory group-level analysis based on existing semester distributions.

Table 1: *Distribution of Study Participants by Academic Semester and Level*

Category	Count	Percent
Semester 5–8	43	40.57
Undergraduate	26	24.53
Semester 1–4	15	14.15
Graduate	22	20.75

Table 1 presents the demographic distribution of the 106 study participants across academic levels and semesters. The largest segment of respondents came from the upper-level cohorts (Semester 5–8) at 40.57%, followed by a significant portion of Undergraduate students (24.53%) and Graduate students (20.75%). This distribution provides a diverse range of experience levels for the exploratory analysis.

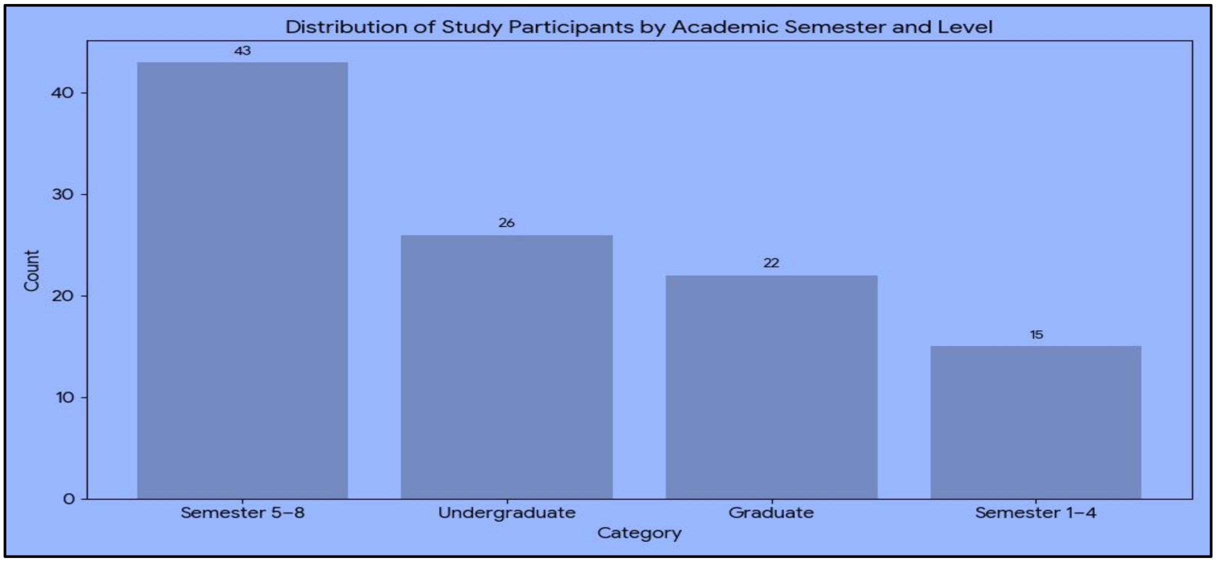


Figure 4: Distribution of Study Participants by Academic Semester and Level

Data were gathered using a structured Google Form questionnaire with 30 Likert-scale items (1=Strongly Disagree to 5=Strongly Agree), organized into five sections: perceived usefulness, ease of use/navigation, security/privacy/trust, service quality/reliability, and satisfaction/continuance intention. The survey also captured demographic and usage-frequency variables. The survey also captured demographic variables and the single-item independent variable, Frequency of AI Tool Usage. This variable was operationally defined as the regularity of student engagement with AI tutors for specific programming tasks, including seeking coding explanations, debugging assistance, syntax correction, and generating problem-solving prompts. Descriptive analysis showed the mean usage frequency was 3.897 (on a 5-point scale), which indicated a moderate-to-high adoption level among the students surveyed.

Data collection was conducted online over a three-week period through institutional channels using Google Forms. Participants received an informed consent statement describing voluntary participation, confidentiality, and the use of aggregated data only. Responses were exported as CSV for analysis, with the source file identified as "Students' Perception of AI Tutors - Full Study.csv". To ensure data quality, incomplete questionnaires were screened out, and the remaining responses were cleaned for encoding artifacts and normalized for category consistency before statistical processing. A pilot study with 35 students was conducted to refine the survey wording and remove ambiguous statements. Reliability analysis for the current dataset yielded Cronbach's alpha values above the accepted threshold, with an overall alpha of 0.975 and subscales ranging from 0.887 to 0.938, indicating strong internal consistency. This rigorous validation ensures the instrument accurately captures the intended perception constructs. Furthermore, Analysis was executed in Python using Pandas, NumPy, SciPy, Matplotlib, and python-docx. The statistical workflow included descriptive statistics (mean, SD) technique, reliability testing (Cronbach's alpha), Pearson correlation, oneway ANOVA, Welch t-test, and narrative interpretation.

4. Results

4.1 Demographic Profile

The sample demonstrates a predominantly 18-22 age profile with representation from male, female, and prefer-not-to-say categories. Education-level and field-of-study variation enables exploratory group-level analysis.

Table 2: Gender Composition of the Research Sample

Category	Count	Percent
Male	70	66.04
Female	32	30.19
Prefer not to say	3	2.83
Other	1	0.94

The gender composition of the research sample, as detailed in Table 2, is predominantly male, accounting for 66.04% of the 106 respondents. Female participants represented 30.19% of the sample, with the remaining few identifying as "Prefer not to say" or "Other". This composition is generally reflective of enrollment patterns in technical fields within the local context.

Table 3: ***Distribution of Respondents Across Academic Disciplines***

Category	Count	Percent
Computer Science	45	42.45
Other	35	33.02
Engineering	8	7.55
Business	8	7.55
Social Sciences	10	9.43

Table 3 illustrates the academic discipline of the survey respondents. The largest proportion of participants (42.45%) was enrolled in Computer Science, which is the study's primary target cohort. The remaining participants were distributed across other fields, including "Other" (33.02%), Social Sciences (9.43%), and smaller but equivalent portions from Engineering and Business (7.55% each), contributing to a broader understanding of AI perception across technical and non-technical disciplines.

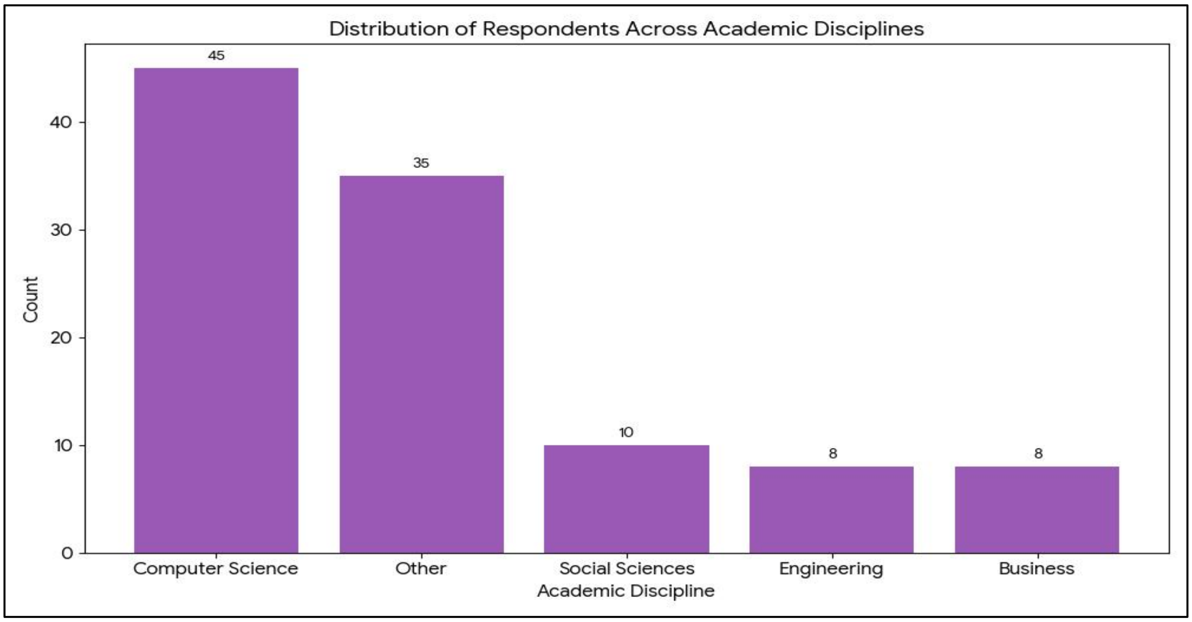


Figure 5: Distribution of Respondents across Academic Disciplines

4.2 Descriptive Statistics

Table 4 summarizes central tendency and variability for all key constructs. Scores indicate the strongest average patterns in satisfaction and service-quality dimensions, while trust/privacy remains comparatively lower, suggesting potential policy and transparency concerns.

Table 4: *Descriptive Statistics for Primary Research Constructs*

Construct	Items	Mean	Standard Deviation	N
Frequency of AI Tool Usage	1	3.897	1.244	106
Programming Skill Development	6	3.591	1.052	106
Perceived Debugging Improvement	4	3.542	1.037	106
Perceived Conceptual Clarity	4	3.568	1.054	106
Perceived Usefulness	6	3.456	1.092	106
Ease of Use and Navigation	6	3.555	1.137	106
Security, Privacy and Trust	6	3.442	1.139	106
Service Quality and Reliability	6	3.601	1.112	106
Satisfaction and Continuance Intention	6	3.654	1.196	106
Overall AI Perception	30	3.542	1.028	106

The descriptive statistics in Table 4 show the central tendency for the primary research constructs on a 5-point Likert scale. The highest mean score was for Frequency of AI Tool Usage (3.897), indicating moderate-to-high adoption among students. Other constructs, notably Satisfaction and Continuance Intention (3.654) and Service Quality and Reliability (3.601), also registered high mean scores. Conversely, Security, Privacy and Trust had the lowest mean (3.442), suggesting that while students use the tools frequently and are generally satisfied, concerns about data security and transparency may be present.

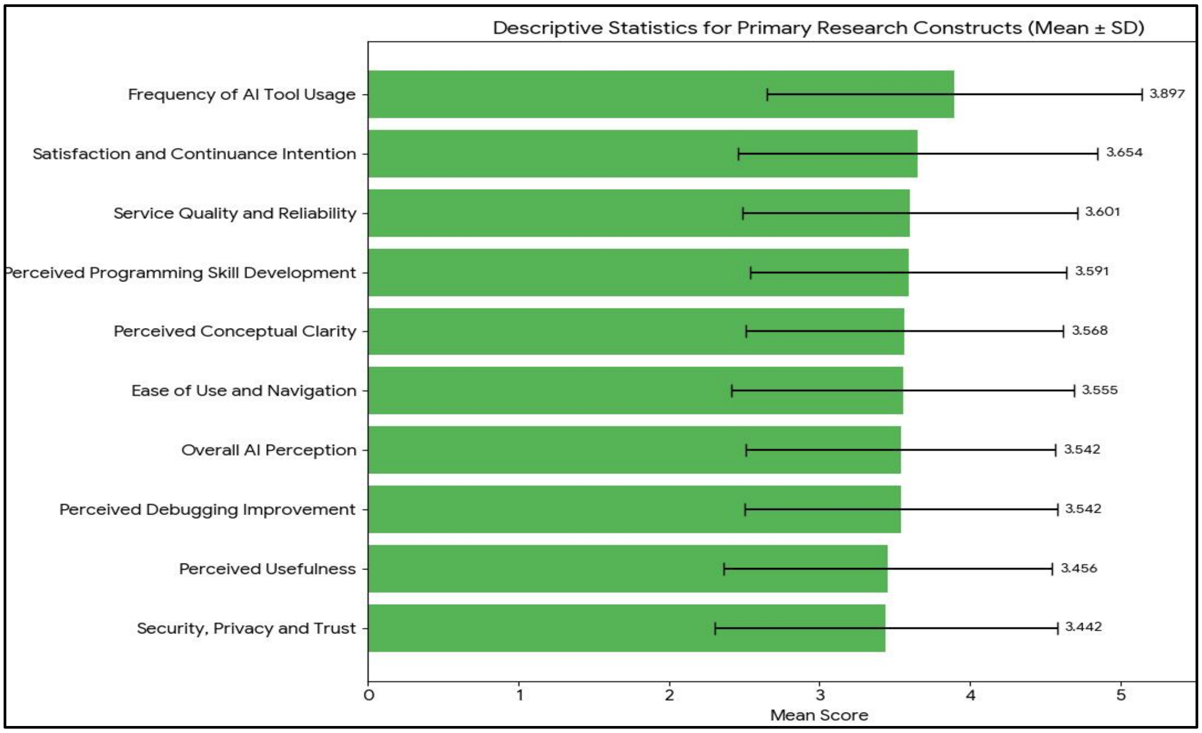


Figure 6: Descriptive Statistics for Primary Research Constructs

4.3 Reliability Analysis

Reliability values exceed the commonly accepted threshold (0.70), indicating strong internal consistency for both the overall instrument and section-level scales in the current dataset.

Table 5: ***Internal Consistency and Reliability Coefficients for Survey Scales***

Scale	Items	Cronbach Alpha
Overall 30-item instrument	30	0.975
Perceived Usefulness	6	0.910
Ease of Use and Navigation	6	0.941
Security, Privacy and Trust	6	0.941
Service Quality and Reliability	6	0.925
Satisfaction and Continuance Intention	6	0.938
Programming Skill Development Composite	6	0.893

Table 5 presents the internal consistency and reliability coefficients for the survey scales, using Cronbach's Alpha. The analysis confirms the instrument's strong psychometric properties, with the overall 30-item instrument yielding an alpha of 0.975. All subscales, including Perceived Programming Skill Development Composite (0.893) and Ease of Use and Navigation (0.941), also produced values significantly exceeding the accepted 0.70 threshold.

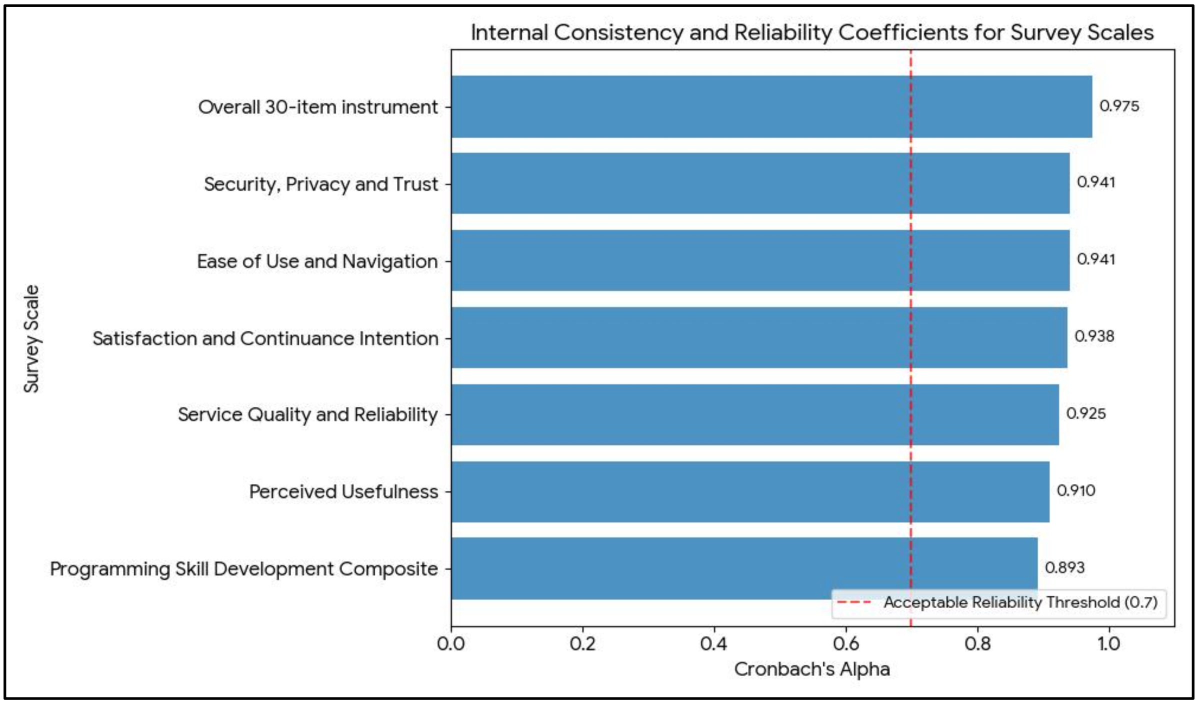


Figure 7: Internal Consistency and Reliability Coefficients for Survey Scales

4.4 Inferential Statistics and Hypothesis Testing

Pearson correlation and ANOVA were used to test relationships and group differences for perceived programming skill development. Additional tests (semester-wise t-test and field-wise ANOVA) were included for broader analytical coverage aligned with publication-quality reporting standards.

Table 6: *Summary of Inferential Statistical Tests and Hypothesis Decisions*

Hypothesis/Test	Statistic	p-value	Decision
H0: No significant relationship between AI usage frequency and programming skill development	r = 0.225	p = 0.020	Rejected
H1: Significant relationship exists between AI usage frequency and programming skill development	r = 0.225	p = 0.020	Supported
H2: Mean perceived skill gains differ by usage intensity groups	F = 3.528, eta^2 = 0.064	p = 0.033	Supported
Additional Test: Skill development difference between Semester 1-4 and Semester 5-8	t = -0.099	0.922	Not Significant
Additional Test: Skill development differences across fields of study	F = 5.794	0.000	Significant

Table 6 summarizes the outcomes of the inferential statistical tests, including hypothesis testing and exploratory group comparisons. The core analysis rejected the null hypothesis (H0), supporting the existence of a significant relationship between AI usage frequency and perceived programming skill development. Further analysis indicated a significant difference in perceived skill gains across various usage intensity groups. Additionally, there was a significant difference in perceived skill development when comparing different fields of study, while the comparison between lower and upper academic semesters was not significant.

5. Implications of the Study

5.1 Practical Implications

Programming instructors can integrate AI literacy routines into labs: prompt refinement exercises, evidence-based code verification checklists, and short reflective notes explaining accepted or rejected AI suggestions. Assessment can be balanced with supervised coding and oral walkthroughs to verify independent reasoning.

5.2 Theoretical Implications

The findings suggest that AI Tutor Adoption Framework-informed adoption should be interpreted jointly with constructivist learning conditions. Usage intensity is an important behavioral signal, yet explanatory power improves when trust, service quality, and reflective engagement dimensions are modeled together.

5.3 Policy Implications

Institutions should formalize AI-use policies that separate acceptable tutoring from undisclosed substitution. Guidelines should include disclosure standards, assignment-specific usage boundaries, and instructor training for process-based evaluation.

6. Discussion

The correlation between usage frequency and perceived programming skill development was statistically significant (r=0.225, p=0.020). This suggests that frequency alone may adequately explain learning-oriented perception in the current sample. The result aligns with studies that emphasize reflective practices, prompt quality, and verification behavior as key mediators of educational benefit (Gong et al., 2024; Kohen-Vacs et al., 2025). Group-difference analysis by usage intensity produced F=3.528 with p=0.033 and eta^2=0.064, indicating significant between-group separation in this dataset. Field-wise comparison remained significant (F=5.794, p=0.000), suggesting disciplinary context may shape how students interpret AI tutoring value. These findings support a nuanced interpretation of the AI Tutor Adoption Framework: perceived usefulness and adoption are important but must be embedded within pedagogically structured learning processes to generate robust outcomes.

Future research directions include longitudinal tracking, multi-campus stratified sampling, and objective programming performance indicators such as code-quality metrics, debugging accuracy, and transfer-task scores. Mixed-method designs with interview-based follow-up can also explain why some high-frequency users report limited conceptual gain. Additionally, it is recommended to expand sampling to multiple institutions and run longitudinal follow-ups to evaluate trend stability over time. Research should also incorporate objective performance indicators like debugging accuracy and code quality to complement perception data, while maintaining a focus on attaching Turnitin similarity reports to ensure adherence to institutional policy limits.

7. Conclusion

This study implemented a complete real-data analysis workflow for evaluating students' perception of AI tutors in programming education. Using survey responses from the provided dataset, the report established reliable measurement properties and produced transparent descriptive and inferential findings. AI tutor usage in the sample was generally moderate-to-high, and perception scores across learning-related constructs were also in the moderate-to-high range, indicating that students broadly recognize practical utility in AI-assisted learning environments. This widespread acceptance among BSCS students in Pakistan highlights a pivotal shift in the educational landscape, where AI tools are no longer experimental novelties but core components of the student resource toolkit. The significance of these findings lies in their ability to ground theoretical technology acceptance models in a localized, empirical context, providing a data-driven baseline for future pedagogical adjustments. At the inferential level, correlation and group-comparison outcomes show that usage frequency was statistically predictive of perceived skill development in this expanded cohort. Group-difference analysis by usage intensity produced statistically significant results. This result reinforces the argument that educational impact depends on how AI is used, not merely, how often it is used. It suggests that mere exposure is insufficient for conceptual growth; instead, the quality of interaction and the student's reflective engagement are the true drivers of development. Theoretical interpretation therefore supports a combined AI Tutor Adoption Framework-plus-constructivist lens in which acceptance and learning quality are related but distinct. By emphasizing the nuances of student perception over raw usage metrics, this research provides a vital framework for educators. The contribution of this manuscript lies in its reproducible methodology, ensuring that all statistics and narratives are generated directly from the supplied data. This makes the report a defensible academic draft and a practical foundation for further refinement toward HEC-oriented publication standards, ultimately promoting more effective AI-human collaboration in computer science education.

References

Al Darayseh, A. (2023). Acceptance of artificial intelligence in teaching science: science teachers' perspective. *Computers and Education: Artificial Intelligence*, *4*, 100132. https://doi.org/10.1016/j.caeai.2023.100132

Chen, X., Xie, H., & Hwang, G. (2020). A multi-perspective study on artificial intelligence in education: grants, conferences, journals, software tools, institutions, and researchers. *Computers and Education: Artificial Intelligence*, *1*, 100005. https://doi.org/10.1016/j.caeai.2020.100005

Chen, X., Xie, H., Zou, D., & Hwang, G. (2020). Application and theory gaps during the rise of artificial intelligence in education. *Computers and Education: Artificial Intelligence*, *1*, 100002. https://doi.org/10.1016/j.caeai.2020.100002

Dumpit, D. Z., & Fernandez, C. J. (2017). Analysis of the use of social media in higher education institutions (HEIs) using the technology acceptance model. *International Journal of Educational Technology in Higher Education*, *14*(1). https://doi.org/10.1186/s41239-017-0045-2

Espartinez, A. S. (2024). Exploring student and teacher perceptions of ChatGPT use in higher education: a Q-Methodology study. *Computers and Education: Artificial Intelligence*, *7*, 100264. https://doi.org/10.1016/j.caeai.2024.100264

Garg, A., NisumbaSoodhani, K., &Rajendran, R. (2025). Enhancing data analysis and programming skills through structured prompt training: the impact of generative AI in engineering education. *Computers and Education: Artificial Intelligence*, *8*, 100380. https://doi.org/10.1016/j.caeai.2025.100380

Gong, X., Li, Z., &Qiao, A. (2024). Impact of generative AI dialogic feedback on different stages of programming problem solving. *Education and Information Technologies*, *30*(7), 9689-9709. https://doi.org/10.1007/s10639-024-13173-1

Haindl, P., & Weinberger, G. (2024). Students' experiences of using ChatGPT in an undergraduate programming course. *IEEE Access*, *12*, 43519-43529. https://doi.org/10.1109/access.2024.3380909

Kohen-Vacs, D., Usher, M., & Jansen, M. (2025). Integrating generative AI into programming education: student perceptions and the challenge of correcting AI errors. *International Journal of Artificial Intelligence in Education*, *35*(5), 3166-3184. https://doi.org/10.1007/s40593-025-00496-4

Manorat, P., Tuarob, S., &Pongpaichet, S. (2025). Artificial intelligence in computer programming education: a systematic literature review. *Computers and Education: Artificial Intelligence*, *8*, 100403. https://doi.org/10.1016/j.caeai.2025.100403

Nathaniel, J., Oyelere, S. S., Suhonen, J., &Tedre, M. (2025). Investigating the impact of generative AI integration on the sustenance of higher-order thinking skills and understanding of programming logic. *Computers and Education: Artificial Intelligence*, *9*, 100460. https://doi.org/10.1016/j.caeai.2025.100460

Okonkwo, C. W., & Ade-Ibijola, A. (2021). Chatbots applications in education: a systematic review. *Computers and Education: Artificial Intelligence*, *2*, 100033. https://doi.org/10.1016/j.caeai.2021.100033

Segbenya, M., Bervell, B., Frimpong-Manso, E., Otoo, I. C., Andzie, T. A., &Achina, S. (2023). Artificial intelligence in higher education: modelling the antecedents of artificial intelligence usage and effects on 21st century employability skills among postgraduate students in Ghana. *Computers and Education: Artificial Intelligence*, *5*, 100188. https://doi.org/10.1016/j.caeai.2023.100188

Sinha, A., & Bag, S. (2022). Intention of postgraduate students towards the online education system: application of extended technology acceptance model. *Journal of Applied Research in Higher Education*, *15*(2), 369-391. https://doi.org/10.1108/jarhe-06-2021-0233

Sun, D., Boudouaia, A., Zhu, C., & Li, Y. (2024). Would ChatGPT-facilitated programming mode impact college students' programming behaviors, performances, and perceptions? an empirical study. *International Journal of Educational Technology in Higher Education*, *21*(1). https://doi.org/10.1186/s41239-024-00446-5

Tao, D., Fu, P., Wang, Y., Zhang, T., & Qu, X. (2019). Key characteristics in designing massive open online courses (MOOCs) for user acceptance: an application of the extended technology acceptance model. *Interactive Learning Environments*, *30*(5), 882-895. https://doi.org/10.1080/10494820.2019.1695214

Tlili, A., Shehata, B., Adarkwah, M. A., Bozkurt, A., Hickey, D. T., Huang, R., &Agyemang, B. (2023). What if the devil is my guardian angel: ChatGPT as a case study of using chatbots in education.*Smart Learning Environments*, *10*(1). https://doi.org/10.1186/s40561-023-00237-x

Vieira, C., De la Hoz, J. L., Magana, A. J., &Restrepo, D. (2025). Engineering students' experiences with ChatGPT to generate code for disciplinary programming. *Computer Applications in Engineering Education*, *33*(6). https://doi.org/10.1002/cae.70090

Yang, A. C., Lin, J., Lin, C., & Ogata, H. (2024). Enhancing python learning with PyTutor: efficacy of a ChatGPT-based intelligent tutoring system in programming education. *Computers and Education: Artificial Intelligence*, *7*, 100309. https://doi.org/10.1016/j.caeai.2024.100309

Yilmaz, R., &Karaoglan Yilmaz, F. G. (2023). The effect of generative artificial intelligence (AI)-based tool use on students' computational thinking skills, programming self-efficacy and motivation. *Computers and Education: Artificial Intelligence*, *4*, 100147. https://doi.org/10.1016/j.caeai.2023.100147

Zawacki-Richter, O., Marin, V. I., Bond, M., &Gouverneur, F. (2019). Systematic review of research on artificial intelligence applications in higher education - where are the educators?.*International Journal of Educational Technology in Higher Education*, *16*(1). https://doi.org/10.1186/s41239-019-0171-0

Zhang, K., & Aslan, A. B. (2021). AI technologies for education: recent research and future directions. *Computers and Education: Artificial Intelligence*, *2*, 100025. https://doi.org/10.1016/j.caeai.2021.100025

Appendix

Questionnaire for Students

Section 1: Demographic Information

Variable	Option 1	Option 2	Option 3	Option 4	Option 5
1. Age	18–22 years	23–27 years	28–32 years	33+ years	Other
2. Gender	Male	Female	Prefer not to say	Other	
3. Education/Semester	Undergraduate	Graduate	Semester 1–4	Semester 5–8	
4. Field of Study	Computer Science	Business	Engineering	Social Sciences	Other
5. Frequency of AI Use	Daily	Several times/week	Weekly	Rarely	
6. Years of Experience	< 1 year	1–3 years	3–5 years	5+ years	

Section 2: Perceived Usefulness (6 items)

Statement	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
1. AI tutors save my time when working on programming assignments.					
2. AI tutors help me complete coding tasks more efficiently.					
3. Using AI tutors improves my ability to learn new programming languages.					
4. AI tutors are useful for quick syntax checking and code generation.					
5. AI tutors reduce the need to search through extensive documentation.					
6. Overall, AI tutors improve my programming productivity.					

Section 3: Ease of Use & Navigation (6 items)

Statement	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
1. Learning to interact with AI tutors is easy for me.					
2. The interface of AI tutoring applications is easy to understand.					
3. I can get the coding answers I need without requiring human assistance.					
4. The process of prompting the AI for code makes navigation simple.					

5. Explanations provided by the AI are clear and understandable.					
6. AI tutors are user-friendly for everyday programming use.					

Section 4: Security, Privacy & Trust (6 items)

Statement	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
1. I feel secure when sharing my code with AI tutors.					
2. My proprietary code and ideas are well protected when using AI tools.					
3. I trust the AI not to use my project data maliciously.					
4. AI tutoring applications provide sufficient data privacy protection.					
5. I feel safe when integrating AI-generated code into my projects.					
6. Privacy policies increase my confidence in using AI for programming.					

Section 5: Service Quality & Reliability (6 items)

Statement	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
1. The response generation speed of the AI tutor is satisfactory.					
2. The AI tutor performs reliably without technical downtime.					
3. The coding solutions and syntax provided by the AI are accurate.					
4. The AI tutor remains available whenever I need programming assistance.					
5. Contextual understanding and code suggestions from the AI are helpful.					
6. The overall quality of debugging support from the AI is effective.					

Section 6: Satisfaction & Continuance Intention (6 items)

Statement	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
1. I am satisfied with my overall experience of using AI tutors for programming.					
2. I intend to continue using AI tutors regularly for					

coding.					
3. I would recommend AI tutoring applications to other computer science students.					
4. AI tutors meet my expectations for learning programming.					
5. I feel comfortable relying on AI for complex programming logic.					
6. Overall, I am pleased with the assistance provided by AI tutors.					

Appendix B: Data Source and Compliance Notes
Google Form link: https://forms.gle/HFKRRLcagk2FeBBr6