Enhancing Software Engineering Education with Emotion-Aware AI Chatbots

Hadeel Sawalha[1] Hadeel Jebara[1] Mamoun Nawahdah[1]

***Abstract*— Despite the growing adoption of AI chatbots in engineering education, most research remains centered on English-speaking learners and overlooks the unique cultural, linguistic, and emotional needs of Arabic-speaking software engineering students. Additionally, many educational chatbots focus on delivering full solutions rather than guiding learners through critical thinking and problem-solving, which undermines engagement and cognitive growth. Emotional dynamics in engineering programming education, known for inducing stress and confusion, are also frequently neglected. This study addresses these gaps through a real-world controlled experiment involving 60 Arabic-speaking software engineering students assigned to one of four instructional setups: Text-Based interaction, Voice-Based interaction, Voice-Real Teacher interaction and Voice-Avatar interaction. Each participant completed the same programming task while emotional and learning-related outcomes were measured. Results showed that the avatar and voice-based setups yielded higher emotional stability, stronger engagement, and improved code quality. These findings demonstrate that emotionally aware, culturally adaptive chatbot systems can significantly enhance engineering programming learning experiences, by supporting both cognitive and affective needs.**

***Index Terms*— AI chatbots, engineering programming education, Arabic-speaking learners, software engineering students, voice interactions, emotion-aware chatbots.**

I. Introduction

Artificial Intelligence (AI) is reshaping higher education by enabling adaptive learning and scalable support [6] [4]. In engineering programming education, AI chatbots serve as virtual assistants, offering instant guidance, debugging help, and personalized explanations [1]. They address common coding challenges and reduce reliance on constant instructor supervision [1].

Despite growing interest in chatbot-assisted education, prior research remains limited in several critical ways. Most existing studies evaluate either text-based or voice-based chatbot, without directly comparing their effectiveness in real-world programming tasks [1] [2]. Some research explores emotion-aware features theoretically or through surveys, but lacks empirical validation [12] [19]. Even when chatbot experiments are conducted, they tend to emphasize surface-level metrics such as engagement or satisfaction, without analyzing deeper learning outcomes like code quality, functional accuracy, or sustained engagement [1]. Moreover, most of this research has centered on English-speaking learners, overlooking the unique linguistic, cultural, and pedagogical challenges faced by Arabic students [3] [1].

To address these gaps, we conducted a controlled experiment at Birzeit University with 60 Arabic-speaking second-year software engineering students. Participants were randomly assigned to one of four support setups: text-only chatbot, voice-only chatbot, voice chatbot with an animated avatar, and a human instructor simulating a chatbot. Each student completed a one-hour programming task, during which we tracked engagement, facially recognized emotions using MorphCast[1], and collected self-reported mood ratings. Two experts evaluated the code for maintainability, readability, and functional accuracy. Results showed that the voice-avatar condition led to better emotional regulation, interaction quality, and code quality. These findings highlight the value of tailored AI chatbots that mimic the role of real teacher and meet both cognitive and emotional needs in culturally underrepresented settings [16].

II. Literature Review

A. *The Role of AI Chatbots in Programming Education*

AI chatbots play a growing role in programming education by offering personalized, real-time support, especially in large or dynamic classrooms [4] [9]. They help students understand concepts, debug code, and complete tasks by analyzing code in real time, identifying errors, and providing structured guidance [5] [8]. Chatbots are particularly valuable for students who are hesitant to seek help and have been widely adopted to deliver on-demand feedback, enhance problem-solving skills, and maintain engagement through interactive learning [6].

B. *Text Based Chatbots: Potential and Limitations*

Text-based chatbots allow students to type questions and receive written responses, helping them review information flexibly and return to explanations whenever necessary [7]. Studies show they improve engagement and comprehension and provide consistent support [5], but their effectiveness is limited by the need for clear query formulation, which can challenge students with limited writing skills or learning disabilities [7] [10]. Moreover, text-based interaction often lacks emotional expression and personalization, causing responses to feel robotic or impersonal and placing additional cognitive load on students during stressful or confusing tasks [2] [15]. They also tend to offer complete solutions rather than guiding students step by step, which can discourage critical thinking and deeper learning [2].

[1]Birzeit University, Birzeit, Palestine hsawalha@birzeit.edu, hjebara@birzeit.edu, mnawahdah@birzeit.edu

[1]https://demo.morphcast.com/

978-9950-8599-0-6

C. Voice Based Chatbots in Programming Education

Voice-based chatbots provide a more natural, interactive engineering programming learning experience by enabling verbal communication that mimics human conversation [3] [1]. They can convey empathy and encouragement through variations in tone, pitch, and rhythm, supporting learners who struggle with typing or formulating written queries and helping to reduce stress during challenging tasks [2] [14]. Additionally, voice interfaces lower cognitive load by delivering spoken feedback, which is critical in time-sensitive activities such as debugging or implementing complex logic [11]. Research indicates that users of voice chatbots experience higher engagement, faster response times, and improved code quality [1] [14], although current systems still face challenges in emotion detection and in accurately interpreting complex, informal queries across diverse dialects and languages [12] [11].

D. Challenges in Current Chatbots

Both text based and voice based chatbots struggle with providing genuine emotional support, handling complex or lengthy queries, and adapting to different languages and cultural contexts [1] [2]. They often deliver direct answers rather than guided, step by step assistance, leading to over-reliance and reduced critical thinking among students [14]. These limitations underscore the need for more emotionally and culturally responsive chatbot designs [15].

E. Emotion Aware Chatbots for Personalized Support

Emotion aware chatbots detect, interpret, and respond to learners' affective states using multimodal recognition. They analyze vocal tone, text, facial expressions, and interaction patterns to adapt explanations dynamically and offer emotional encouragement [12] [15]. Integrated avatars further enhance empathy through human-like expressions [18], improving resilience, engagement, and code quality. However, their real-world classroom efficacy and cultural adaptability remain underexplored [20] [17].

F. Research Gap

Most existing studies focus on either text- or voice-based chatbots in education, often without rigorous experimental comparison during authentic engineering programming tasks [14]. Although some research explores emotion-sensitive features, it is largely theoretical or survey-based, lacking evaluation of learning outcomes such as code quality or maintainability [2]. Moreover, Arabic-speaking learners remain underrepresented, despite their distinct cultural and linguistic needs [1]. This study addresses these gaps through a controlled experiment comparing four instructional modes with Arabic-speaking software engineering students in a real engineering programming context [14].

Our study fills these gaps by experimentally comparing four chatbot setups, including an emotion-aware avatar mode, on an authentic programming task with Arabic-speaking software engineering students. We measure emotional detection, engagement, code quality. These insights will guide the design of custom AI chatbots that closely mimic real teachers, with emotional support tailored to the needs of Arabic learners [16] [14].

III. Methodology

A. Experimental Design

This study employed a controlled experimental design with 60 second-year Software Engineering students from Birzeit University, all of whom had basic object-oriented programming (OOP) skills. Participants registered online and were randomly assigned to one of four setups of 15 students each to balance background and experience. Each student completed a 60-minute programming task.

The independent variable was interaction mode (Text, Voice, Real Teacher, Avatar). We measured two dependent variables: (1) **Engagement**, quantified as the effective interaction time between each student and the chatbot during the engineering programming task—specifically the duration spent exchanging queries and responses while coding. And (2) **Code Quality**, evaluated by expert reviewers on code accuracy, readability and maintainability. One reviewer had 5 years of experience and the other had 8 years of experience in teaching and reviewing programming projects. Their extensive background added reliability, accuracy, and expert insight to the evaluation process. A pilot study was conducted to verify the consistency and reliability of these measures across all setups.

The following null hypotheses were tested to evaluate the effects of chatbot modality:

- **H0 (Engagement):** There is no significant difference in student engagement across the four interaction setups.
- **H1 (Code Quality):** There is no significant difference in student code quality across the four interaction setups.

B. Experimental Model

To ensure consistent and realistic chatbot behavior across all interaction setups, the Wizard of Oz (WoZ) technique was used. In this method, a human operator simulated chatbot responses from behind the scenes, allowing for natural interaction while maintaining control over the quality, timing, and emotional tone of responses. This approach was essential for accurately reflecting each setup's intended behavior, especially in emotion-aware conditions, without relying on autonomous chatbot systems that may vary in performance.

The Wizard of Oz model enabled the researcher to adapt replies in real time based on student inputs and emotional cues, ensuring the instructional intent of each condition was maintained throughout the task. It also allowed for uniformity in communication style, minimizing the risk of technical inconsistencies influencing the outcomes.

C. Experimental Setups

Setup 1: Text-Based Interaction: Students received structured written responses via text chat alongside the IDE. The teacher monitored each student's shared screen, replied in the chat, and used an AI assistant to support explanations. Emotional cues were inferred from text tone and emojis; the

Fig. 1: Wizard-of-Oz simulation model used to emulate chatbot interaction with consistency and control.

operator logged sentiment indications for qualitative analysis. Figure 2 shows the overall setup.

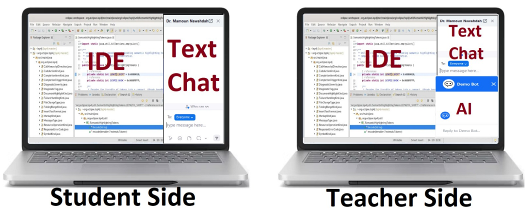

Fig. 2: Setup 1 – Text-based chatbot interaction.

Setup 2: Voice-Based Interaction: Students asked questions verbally via voice chat, while the teacher responded by voice and supplemented answers with chat messages when needed. An AI assistant supported quick formulation of detailed verbal explanations. The operator observed vocal tone and pacing to infer emotions, recording these cues in research notes for objective measures. Figure 3 shows the overall setup.

Fig. 3: Setup 2 – Voice-based chatbot interaction.

Setup 3: Voice–Real Teacher Interactions: Building on the Voice setup, this mode introduced real-time emotion detection using MorphCast. A live video feed of the teacher was displayed to the student. Facial expressions captured by MorphCast guided the teacher to adapt tone and content of responses according to detected emotions (e.g., confusion, frustration, satisfaction), enhancing empathy and support. MorphCast triggers allowed instant encouragement for frustration or advanced hints for positive emotions. Figure 4 shows the overall setup.

Setup 4: Voice-Avatar Interaction: Instead of showing the teacher directly, this mode used an animated avatar that mapped the teacher's voice and emotional tone. MorphCast

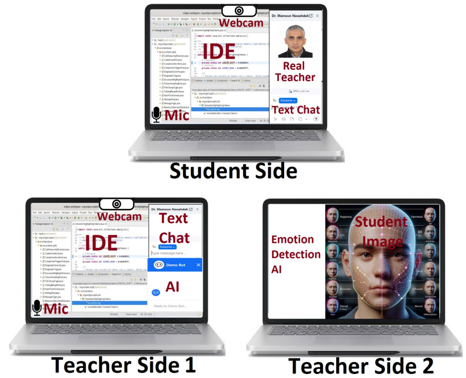

Fig. 4: Setup 3 – Voice–Real Teacher Interactions.

analyzed student emotions, and the avatar mirrored both the teacher's intended responses and appropriate facial expressions. This setup combined vocal clarity with visual emotional cues. The avatar's expressions and voice modulation dynamically reflected student affect to reinforce engagement. Figure 5 shows the overall setup.

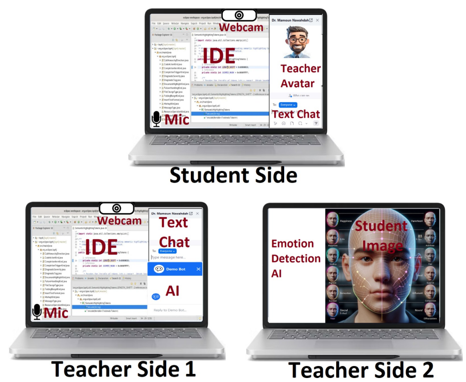

Fig. 5: Setup 4 – Avatar-assisted voice interaction.

D. Procedure

The experiment comprised three phases: Pre-Task, During-Task, and Post-Task.

Pre-Task Phase: Participants completed a survey on programming experience and familiarity with chatbot technologies, then received a project package with a base starter code for the Student Data Management System.

During-Task Phase: Students implemented the remaining components of the system within 60 minutes:

- Implement the `StudentList` class to support `Iterable<Student>`.
- Display student records in a read-only `TextArea`.
- Create an "Add Student" dialog.
- Add a "Save to File" feature.

Trained observers logged session times, question counts and types (syntax, logic, feature), and emotional responses at early, mid, and late stages.

Post-Task Phase: participants completed a survey assessing how the interaction modality influenced their engagement and how effectively the chatbot helped identify bugs or errors, positively or negatively. The survey also asked for their preferred interaction modality (Text, Voice, Real Teacher, or Avatar), their choice of assistance source (chatbot, teacher, or both), their overall perception of the chatbot's effectiveness during the task, and any suggestions for improving the interaction. Finally, all code submissions were collected for evaluation by two experts on code quality, focusing on code accuracy, readability, maintainability and the overall quality.

E. Data Collection and Observation

All sessions were recorded, capturing student and instructor screens, webcam video, audio, and chat windows, to create a complete interaction log. Observers took notes on session start and end times, the number and type of questions asked (syntax, logic, feature implementation), and how students phrased follow-up queries. Before the task, students completed a pre-task survey on their engineering programming experience and prior use of similar tools; immediately afterward they filled out a post-task survey on perceived engagement, preferred assistance source (chatbot, teacher, or both), and positive or negative effects. Objective engagement was computed using the formula in Equation 1.

$$\textbf{Engagement\%} = \frac{\textbf{Active Interaction Time}}{\textbf{Total Session Time}} \times \textbf{100\%} \qquad (1)$$

Emotional states during the learning sessions were classified into three primary categories positive, negative, and neutral, based on facial expression recognition outputs from the MorphCast Emotion AI system. The operational definitions were as follows:

- **Positive State:** Presence of facial expressions corresponding to joy, interest, or satisfaction, with a MorphCast confidence score ≥ 0.60 for any of these emotions.
- **Negative State:** Presence of expressions associated with frustration, confusion, boredom, sadness, or anger, with a MorphCast confidence score ≥ 0.60 for any of these emotions.
- **Neutral State:** Absence of detectable positive or negative indicators above the set threshold, or a balanced mix of low-intensity expressions across emotional categories, resulting in no dominant affective state.

Final code submissions were reviewed by two experts, who provided narrative feedback on readability (structure, naming, comments), maintainability (modularity, best practices), and functional accuracy (correctness and bug resolution). Observers also logged emotional cues such as emojis in text chat, variations in vocal tone in voice modes, and MorphCast software facial expression data in the Real Teacher and Avatar setups to contextualize how affect influenced problem solving and code production.

F. Data Analysis

One-way ANOVA was used because there was a single independent variable (interaction mode) with many conditions, allowing simultaneous comparison of group means. This test determines if differences among means are statistically significant at a 95% confidence level. Box plots were generated to visualize distributions, medians, interquartile ranges, and outliers for each condition, aiding interpretation of variance and group differences.

IV. Results

A total of 60 students (15 per setup: Text, Voice, Real Teacher, Avatar) completed the study and all corresponding measures. The following subsections present findings on self-reported experiences, objective engagement, emotional responses, and expert assessments of code quality.

A. Engagement

Engagement was evaluated using two measures: self-reported ratings from the post-task survey (subjective engagement) and the percentage of active interaction time during the session (objective engagement, see Eq. 1). A one-way ANOVA revealed significant differences across the four setups for both measures (subjective: $p=0.0071$, objective: $p=0.000004$). For subjective engagement, mean ratings were: Text (3.43), Voice (4.36), Real Teacher (3.79), and Avatar (4.57), with the Avatar and Voice setups achieving the highest levels of reported engagement. Figure 6 presents the self-reported engagement scores across the four instructional setups.

Objective engagement showed a similar trend, with mean interaction times of Text (15.15%), Voice (26.54%), Real Teacher (10.00%), and Avatar (24.46%). These results indicate that the Avatar and Voice setups sustained higher levels of active participation compared to Text and Real Teacher. Fig. 7 shows the objective engagement scores across the four instructional setups.

Fig. 6: Self-reported student engagement scores across the four instructional setups.

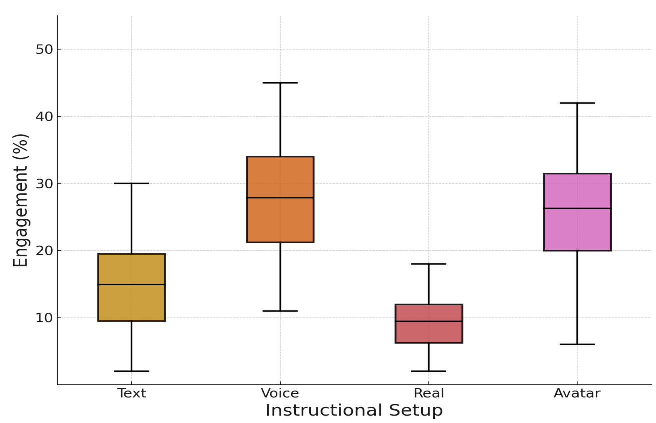

Fig. 7: Objective engagement scores across the four instructional setups.

B. Engagement Time

Engagement time (in minutes) was compared across the four instructional setups using a one-way ANOVA, which revealed a statistically significant difference ($p=0.0363$).

Mean engagement times were: Text (11.71), Voice (13.14), Real Teacher (8.86), and Avatar (12.57). Voice and Avatar showed the highest mean engagement times, indicating that students in these setups tended to remain actively engaged for longer durations than those in Text and Real Teacher. Figure 8 presents engagement time across the four instructional setups.

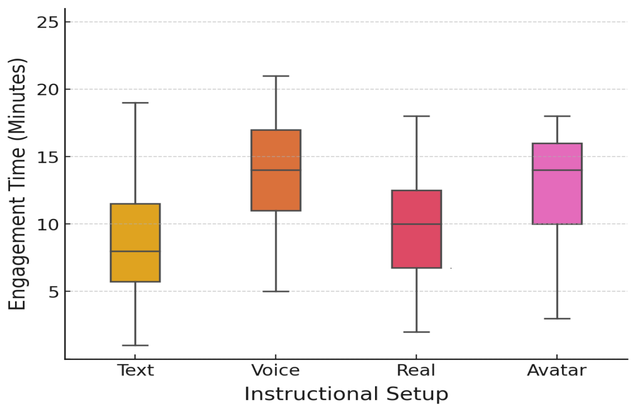

Fig. 8: Comparison of student engagement time across the four instructional setups.

C. *Emotional Valence Over Time*

Valence scores (1–5 scale) for each setup across three intervals (early: 0–20 min, middle: 20–40 min, final: 40–60 min) were as follows:

- **Text:** 2.7 → 3.0 → 3.3
- **Voice:** 3.0 → 3.4 → 3.9
- **Real Teacher:** 3.0 → 3.2 → 3.1
- **Avatar:** 2.8 → 3.8 → 4.6

The Avatar setup showed the greatest positive shift, while Voice and Text rose moderately, and Real Teacher remained near neutral. Fig. 9 shows the emotional valence over time across Real Teacher and Avatar setups.

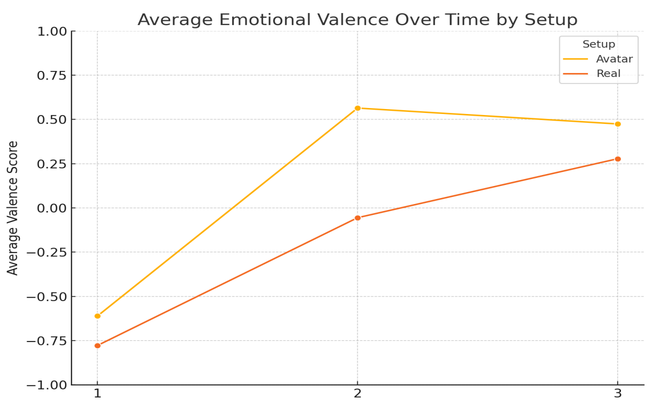

Fig. 9: Emotional valence progression across Real Teacher and Avatar setups.

D. *Emotional Valence Comparison*

Mean final-interval valence by setup was: Text (3.3), Voice (3.9), Real Teacher (3.1), Avatar (4.7). The difference between Avatar and Real Teacher was significant (p=0.0007), underscoring the avatar's superior support for positive emotional states.

Valence scores were calculated using Equation 2, which captures the net balance of positive versus negative cues:

$$\textbf{Valence Score} = \frac{\textbf{Positive Count - Negative Count}}{\textbf{Total Emotion Count}} \tag{2}$$

E. *Emotional Valence Comparison*

Figure 10 shows that students in the Avatar condition expressed overwhelmingly more positive emotions (40) than negative (16) or neutral (11), whereas in the Real Teacher condition negative cues (35) slightly outnumbered positive (22) and neutral (18). This distribution underscores the avatar's ability to evoke encouraging affective responses while live instructor interactions, though expert, tended to stimulate more negative reactions. By grouping emotional occurrences into positive, negative, and neutral categories, we can see that the emotionally adaptive design of the avatar created a noticeably more supportive engineering programming learning environment.

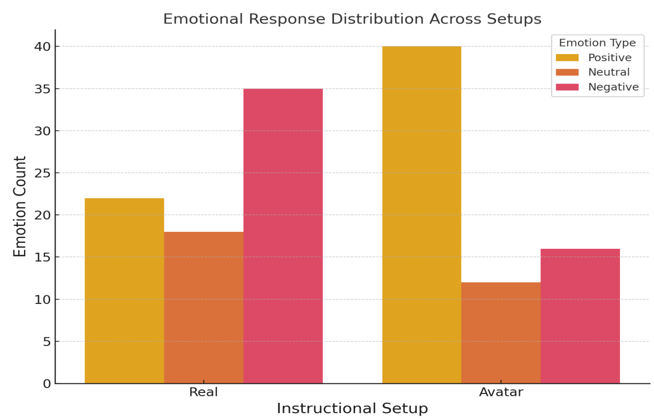

Fig. 10: Positive, neutral, and negative emotional occurrences across Real Teacher and Avatar setups.

F. *Code Quality*

Code quality was assessed by two expert reviewers with extensive experience in Java programming, teaching, and code review. Their combined insights highlighted how the four setups (Text, Voice, Real Teacher, Avatar) influenced both the structure and overall quality of students' code.

In terms of **task completion**, a one-way ANOVA showed a significant difference between setups (p=0.0053). Voice (83.85%) and Real Teacher (82.69%) achieved higher completion rates than Avatar (67.31%) and Text (63.23%), indicating that more direct or interactive approaches tended to help students finish a larger portion of the assigned task within the session. Fig. 11 shows task completion percentages across the four setups.

Submissions from the Avatar and Voice conditions tended to demonstrate clearer **maintainability**, with user interface

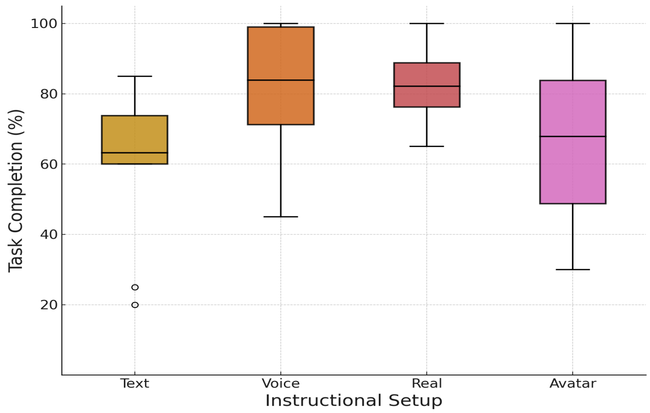

Fig. 11: Task completion percentage across the four instructional setups.

elements separated from core logic, helper classes used effectively, and methods designed with specific, well-defined responsibilities. In contrast, Text and Real Teacher submissions often exhibited tightly coupled code, mixing UI with logic and relying on longer, ad hoc methods, which reduced modularity and maintainability.

In terms of **readability**, Avatar and Voice submissions more consistently used descriptive identifiers, applied uniform naming conventions, and incorporated inline comments that clarified control flow. This made the code easier to follow and reduced the cognitive effort required for understanding. By contrast, Text and Real Teacher code frequently suffered from inconsistent naming, sparse documentation, and uneven indentation, which reviewers noted increased the time required to grasp the logic.

For **functional accuracy**, the Avatar and Voice groups generally included systematic error checks, input validation, and thorough handling of edge cases, while the other setups occasionally omitted these safeguards. This sometimes resulted in reduced robustness and a higher likelihood of runtime errors.

V. Discussion

A. *Engagement*

Engagement outcomes were assessed through both subjective self-reports and objective interaction time, with statistically significant differences observed across the four setups for both measures (subjective: p=0.0071, objective: p=0.000004).

For *subjective engagement*, mean post-task ratings were highest in the Avatar (4.57) and Voice (4.36) conditions, followed by Real Teacher (3.79) and Text (3.43). Participants in the Avatar and Voice setups consistently reported greater involvement and attention throughout the task.

For *objective engagement*, measured as the percentage of active interaction time, the Voice condition achieved the highest mean (26.54%), closely followed by Avatar (24.46%), with Text (15.15%) and Real Teacher (10.00%) trailing. These results indicate that, statistically, the Avatar and Voice modalities sustained higher levels of active participation compared to Text and Real Teacher.

From a practical perspective, these findings align with earlier research showing that multimodal feedback can help maintain focus and reduce cognitive load. Abdallah et al. [4] demonstrated that combining visual and auditory cues in classroom chatbots increases on-task behavior, while Groothuijsen et al. [5] found that integrated voice–visual interfaces promote continuous engagement in programming tasks. Our results extend this evidence to programming education, highlighting that expressive, interactive modalities such as Avatar and Voice can both statistically and practically enhance student engagement.

B. *Emotional Dynamics*

The Avatar condition showed the largest positive shift in emotional valence, rising from 2.8 at the start to 4.6 by the end of the task. Voice interactions increased from 3.0 to 3.9, Text from 2.7 to 3.3, while Real Teacher remained relatively stable ($3.0 \rightarrow 3.1$). The difference between Avatar and Real Teacher in final valence was statistically significant ($p = 0.0007$), highlighting the stronger support of the avatar for positive emotional states.

The emotional logs revealed 28 positive, 11 negative and 6 neutral signals in the Avatar setup, versus 17 positive, 19 negative, and 9 neutral in Real Teacher. These results suggest that a nonjudgmental, expressive avatar can create a more encouraging atmosphere, potentially mitigating performance anxiety. Although valence improvements in Voice and Text conditions were not statistically tested for significance, their upward trends indicate that real-time affective feedback, particularly facial expressions and vocal tone adjustments, may act as microinterventions that help students recover from moments of frustration and maintain motivation [2] [15].

C. *Code Quality*

Code quality analysis combined statistical evidence on task completion with expert reviewer insights. The findings indicate that setups providing direct or highly interactive feedback, such as Voice and Real Teacher, tended to support students in completing a greater portion of the assigned work within the session. This may be linked to the immediacy of responses and the presence of verbal interaction, which, as noted in prior studies [10] [13], can enhance persistence and maintain momentum in problem-solving. Still, with only 15 students per group, these patterns should be considered suggestive rather than conclusive.

In terms of **maintainability**, **readability**, and **functional accuracy**, submissions from the Avatar and Voice setups showed clearer structure and better coding practices. These included separating user interface from core logic, using helper classes and concise methods, applying consistent naming conventions, maintaining proper indentation, adding explanatory comments, and implementing systematic error handling and input validation. By contrast, code from the Text and Real Teacher setups more often exhibited tightly coupled components, inconsistent identifiers, minimal documentation, and weaker safeguards, which reduced clarity and robustness.

Overall, the findings suggest that emotionally supportive and feedback-rich environments, particularly those combining verbal and visual cues, can help students develop better coding habits and produce cleaner, more organized program structures. This effect may be further strengthened when learners feel motivated and confident during the coding process [10] [13] [21]. However, the small sample size limits statistical confidence, and further studies with larger participant groups are needed to determine the extent to which these advantages generalize across contexts and programming tasks.

VI. Conclusion and Future Work

This study evaluated four instructional setups: Text, Voice, Real Teacher, and Avatar, for teaching programming to Arabic-speaking software engineering students. Statistically validated results revealed significant differences in engagement ($p=0.0071$ subjective, $p=0.000004$ objective) and task completion ($p=0.0053$). The Avatar and Voice conditions achieved the highest reported engagement and maintained longer active interaction times, while the Voice and Real Teacher conditions completed a greater portion of the assigned task. Expert code reviews further showed that submissions from the Avatar and Voice groups demonstrated stronger maintainability, readability, and functional accuracy. These outcomes highlight the value of emotionally supportive, feedback-rich environments, especially those combining verbal and visual cues, in fostering cleaner program structure, sustained focus, and proactive help-seeking behaviors.

While promising, these findings should be interpreted with caution due to the study's limitations, including the small sample size (15 students per condition), single-institution scope, short 60-minute session, and reliance on a Wizard of Oz simulation rather than fully autonomous chatbot systems. Additionally, emotion detection relied primarily on facial-expression analysis, which may not capture the full complexity of internal emotional states or account for cultural differences in emotional expression.

In future work, we aim to extend this research to multiple Palestinian universities, include a larger and more diverse student population, and apply the approach across different programming and software engineering courses. We also plan to explore the real-time deployment of fully autonomous AI chatbots in authentic classroom settings to evaluate their long-term impact on programming learning outcomes, skill development, and student well-being.

References

[1] H. Sawalha, M. Nawahdah, and H. Jebara, "GUIDELINES FOR DEVELOPING PERSONALIZED AND EFFECTIVE VOICE CHATBOTS IN EDUCATION," in *19th International Technology, Education and Development Conference (INTED2025) Proceedings*, Valencia, Spain, Mar. 3–5 2025, pp. 5921–5929.

[2] H. Jebara, M. Nawahdah, and H. Sawalha, "EMOTION RECOGNITION IN EDUCATIONAL CHATBOTS: GUIDELINES FOR ENHANCING ENGAGEMENT AND PERSONALIZATION," in *19th International Technology, Education and Development Conference (INTED2025) Proceedings*, Valencia, Spain, Mar. 3–5 2025, pp. 5930–5937.

[3] A. Abdo and S. Yusof, "Exploring the Impacts of Using the Artificial Intelligence Voice-Enabled Chatbots on Customers Interactions in the United Arab Emirates," *IAES Int. J. Artif. Intell. (IJ-AI)*, vol. 12, no. 4, pp. 1920–1927, Dec. 2023.

[4] A. K. Abdallah, A. M. Alkaabi, D. A. F. Mehiar, and Z. A. J. Aradat, "Chatbots in Classrooms: Tailoring Education and Boosting Engagement," in *Cutting-Edge Innovations in Teaching, Leadership, Technology, and Assessment*, IGI Global, 2024, pp. 166–181.

[5] S. Groothuijsen, A. van den Beemt, J. C. Remmers, and L. W. van Meeuwen, "AI chatbots in programming education: students' use in a scientific computing course and consequences for learning," *Comput. Educ. Artif. Intell.*, vol. 7, p. 100290, 2024.

[6] C. W. Okonkwo and A. Ade-Ibijola, "Chatbots applications in education: A systematic review," *Comput. Educ. Artif. Intell.*, vol. 2, p. 100033, 2021.

[7] M. A. Kuhail, N. Alturki, S. Alramlawi, and K. Alhejori, "Interacting with educational chatbots: A systematic review," *Educ. Inf. Technol.*, vol. 28, pp. 973–1018, 2022.

[8] J. Nayak, T. Keane, T. Linden, and A. Molnar, "Teaching High School Students Artificial Intelligence by Programming Chatbots," in *Teaching Coding in K-12 Schools: Research and Application*, Springer, 2023, pp. 263–276.

[9] L. Labadze, M. Grigolia, and L. Machaidze, "Role of AI Chatbots in Education: Systematic Literature Review," *Int. J. Educ. Technol. High. Educ.*, vol. 20, no. 1, p. 56, 2023.

[10] S. Bobadilla, R. Glassey, A. Bergel, and M. Monperrus, "SOBO: A Feedback Bot to Nudge Code Quality in Programming Courses," arXiv:2303.07187, 2023.

[11] M. De La Roca, M. M. Chan, A. Garcia-Cabot, E. Garcia-Lopez, and H. Amado-Salvatierra, "The Impact of a Chatbot Working as an Assistant in a Course for Supporting Student Learning and Engagement," *Comput. Appl. Eng. Educ.*, early view, 2024.

[12] C. Zhu, "Research on Emotion Recognition-Based Smart Assistant System: Emotional Intelligence and Personalized Services," *J. Syst. Manag. Sci.*, vol. 13, no. 5, pp. 227–242, 2023.

[13] S. Palahan, "PythonPal: Enhancing Online Programming Education through Chatbot-Driven Personalized Feedback," arXiv:2503.16487, 2025.

[14] H. Sawalha, M. Nawahdah, and H. Jebara, "Evaluating the Effectiveness of Voice and Text Software Chatbots on Students' Programming Skills," in *Proc. 2025 IEEE Int. Conf. Smart Learn. Courses (SCME)*, Hebron, Palestine, Jun. 15–16 2025, pp. 126–131.

[15] H. Jebara, M. Nawahdah, and H. Sawalha, "Evaluating the Impact of Emotion-Aware AI Chatbots on Student Engagement and Programming Skills," in *Proc. 2025 IEEE Int. Conf. Smart Learn. Courses (SCME)*, Hebron, Palestine, Jun. 15–16 2025, pp. 132–137.

[16] M. Nawahdah, H. Sawalha, and H. Jebara, "An Interactive Emotional-Aware Chatbot for Enhancing Computer Programming Education Among Arabic-Speaking Students," in *Proc. 2025 6th Int. Conf. Artif. Intell. Educ. Technol. (AIET)*, Munich, Germany, Jul. 29–31 2025.

[17] N. Ahmad and A. Farooq, "Fine-Tuning LLMs for Emotional Interaction in Learning Platforms," in *Int. Conf. Artif. Intell. Educ.*, 2024.

[18] H. Park and Y. Lee, "Affective Interaction in Avatar-Based Learning: Design Considerations and Student Perceptions," *Educ. Technol. Res. Dev.*, vol. 70, pp. 1871–1890, 2022.

[19] Y. Chen, M. Li, and J. Zhang, "Emotion-AI in Education: Enhancing Student Engagement through Affective Feedback Loops," *IEEE Trans. Learn. Technol.*, vol. 16, no. 2, pp. 180–192, 2023.

[20] R. Winkler and A. Baltasar, "Social Presence in Emotion-Aware AI Tutors: Impacts on Learning Motivation and Interaction Quality," *Comput. Educ.*, vol. 195, p. 104677, 2023.

[21] M. Nawahdah, H. Sawalha, R. Salameh, and M. Taha, "Evaluating the Accuracy and Effectiveness of AI-Based Grading in Computer Science Education," in *Proc. 2025 IEEE Int. Conf. Smart Learn. Courses (SCME)*, Hebron, Palestine, Jun. 15–16 2025, pp. 13–18.