C.A.T.E.: A Multimodal AI Agent for Context-Aware and Bilingual Learning in Technical Education

*Siti Robaya Jantan[1], Ummi Haani Shahrul[2], Roslinda Murad[3]

[1,2,3]Universiti Poly-Tech Malaysia, Jalan 6/91, Taman Shamelin Perkasa, 56100 Kuala Lumpur, Malaysia
*robaya@uptm.edu.my
kl2307013921@student.uptm.edu.my
roslinda@uptm.edu.my

Abstract. The new large language models (LLMs), it is impressive in their generality, remain inadequate in terms of context-awareness, tool integration, and real-time flexibility that technical education requires. We offer a solution to this problem by introducing C.A.T.E. (Compiler Assistant for Technical Enhancement), a domain-specific, multimodal AI agent. Operating with an agile methodology and powered by LangGraph orchestration on GraphQL, C.A.T.E. incorporates third-party applications such as YouTube, Wikipedia, Google Books, and Wolfram Alpha. The system was tested by more than 30 participants from UPTM and UMT, with results indicating 92% satisfaction rate and receiving commendation for relevance, clarity, and technical accuracy. C.A.T.E. was able to consistently beat mainstream LLMs in controlled comparisons by providing task-specific, localized, and pedagogically practical assistance. This study demonstrates how LLM-based agent-powered tools, such as C.A.T.E., can transform the world of education through easily accessible, trustworthy, and highly contextual AI-based learning assistance.

Keywords: AI Agents, Context-Aware Systems, GraphQL Integration, Large Language Models, Multimodal Learning, Technical Education.

1 Introduction

Large language models, which have evolved with the advent of artificial intelligence, are capable of interpreting, reasoning, and interacting with human-like input in a sophisticated manner. Illustrated by OpenAI's GPT-3 and GPT-4, these models are developed based on the transformer architecture and trained using vast quantities of text. They can be extensive in what they can do, including translation, summarization, code generation, and question answering (Brown et al., 2020; OpenAI, 2023). This act of progress, which began in the late 2010s, has not only transformed natural language processing but also led to the emergence of the concept of an AI agent autonomous system that is in a position to plan, adapt, and interact with humans in a form of natural language interface (Vaswani et al., 2017). The switch to dynamic agents has been apparent since the emergence of autonomous GPT-based programs such as AutoGPT or BabyAGI, which illustrate the possible use of recursive thinking, task decomposition, and dynamic tool usage by sequencing LLM responses in hierarchical feedback loops (Chen et al., 2023; Shinn et al., 2023). These innovations opened the doors to active rather than passive responders and laid a solid foundation for the future of intelligent, goal-oriented systems across various applications.

Following this development, we have different categories of AI agents, as articulated based on their operational paradigms. Reactive agents are instantaneous responses to environmental sources, without storing past experiences or internal models that only utilize the current inputs to act (Russell & Norvig, 2009). In comparison, model-based agents possess internal models of the world and can therefore make better, context-sensitive decisions (Wooldridge, 2025). They have a

N. A. Ishak et al. (eds.), *Proceedings of the International Conference on Cross- Disciplinary Academic Research 2025 - Track 1 Advances in Computing, Electronics, Engineering, and Mathematics (ICAR-T1 2025)*, Advances in Engineering Research 296,
https://doi.org/10.2991/978-94-6239-636-4_6

goal-based and a utility-based agent, both of which compare potential actions against previously defined goals or utility functions to optimize the results, thus adding a dimension of strategic planning (Gat, 1998). Learning agents can be characterized by the progressive optimization of performance given individual feedback, new data, and user interface interactions over time (Yao et al., 2023). Most recently, there has been an emergence of the concept of situated cognitive agents, which unify multimodal data streams, including text, code, and video, and the real-time orchestration of specialist tools to solve dynamic, complex problems (Kiela et al., 2020). Which it includes a bilingual response suitable to adjust the issue at any way possible regarding the student knowledge, the bilingual response is being trained by the admin system to ensure understanding better with the real-time response. Nevertheless, the research area that employs such advanced agents in professional training, where expertise is significant, such as the field of computer science, is relatively unexplored (Luckin & Holmes, 2016).

This paper aims to fill this gap by discussing the design, development, and pilot launch of one of the AI agents, called C.A.T.E. (Compiler Assistant for Technical Enhancement), which is designed to support technical education in Malaysian universities. The need addressed by the creation of C.A.T.E. was a result of a pre-study that identified a strong need among 34 students, with 88.6% saying that they had little to no experience in using any AI-related tools, 50% complaining of finding greater guidance on the research, and 44.1% requiring assistance in performing mathematical confirmation. These results highlight a significant lack of compatibility between the currently available digital material and the unique learning needs of computer science learners, as well as those with limited or no exposure to AI. To close this gap, C.A.T.E. was created, an agent that aids learning with situated cognitive abilities and GraphQL based orchestration, providing bilingual context-aware support in English and Malay, as well as feedback via multimodal interfaces, such as analyzing video transcripts and processing data. This work aims at assessing the role of C.A.T.E. as an effective intervention and evaluating it in terms of learning outcomes, ease of use, and student engagement by completing pre- and post-questionnaires by students and adding to the existing knowledge base on the application of AI-based educational technologies and the possibility to revolutionize the process of gaining technical skills in universities (Holstein et al., 2019; Zawacki-Richter et al., 2019).

2 Methodology

In the modern debate on technology-enhanced learning, the multimodal artificial intelligence agent C.A.T.E. has been developed specifically to improve computer science education. Using an architecture that has been tuned for iterative prototyping and reconfigured for scalable deployment, C.A.T.E. is now in use at Universiti Poly-Tech Malaysia. This modular design meets the real-time requirements of the learning environment while still being mindful of the dynamic nature of technical instruction. The history of the model reflects a broader trend in the educational technology literature, where the focus has shifted toward dynamic, self-directed learning aids (Baker et al., 2021).

As a result, the methodological focus of C.A.T.E. is characterized by an emphasis on flexibility, a user-centered approach, and strategic technology application, which collectively address the ever-changing needs of technical education.

2.1 System Design of C.A.T.E.

In the field of next-generation educational technology, C.A.T.E. is a fascinating landmark: a web-based, multimodal educational agent system driven by artificial intelligence. The platform was created under the supervision of the research team headed by Nguyen et al. (2025) and synergistically combines state-of-the-art frontend frameworks (Next.js and Tailwind CSS) with backend execution via LangChain, LangGraph, and Convex. This type of architecture can support reasoning, persistent memory, and rapid access to real-time data; these are the fundamental functionalities essential to support robust, dynamic dialogue.

It is essential to note that C.A.T.E. supports both English and Malay, thereby providing bilingual natural language interaction. To make it more pedagogically clear, its syntax-sensitive debugger uses color-coded highlighting to indicate syntax errors. Additionally, it fosters contextual consonance between the user and course-specific content, including syllabi, assessment rubrics, learning objectives, and curated content from sources such as YouTube, Wikipedia, and Google Books. These are then interpreted and projected onto the learner's response, resulting in multimodal feedback in the form of Abstract Syntax Trees (ASTs), parse trees, and control flow graphs which are formal representations that scaffold metacognitive reasoning.

The general architecture of C.A.T.E. goes beyond the conventional chatbot paradigm to place itself as the agent of persistent interest and enhanced understanding in the technical fields of study. The effectiveness of multimodal learning approaches has been confirmed by empirical studies (Kiela et al., 2020), and initial evaluations of users indicate that they perceive the system as responsive and incrementally helpful throughout their studies.

2.2 Constructivist Learning Theory

Constructivist learning theory, primarily attributed to Piaget (1952) and Vygotsky (1978), posits that learners actively construct knowledge through interaction with their environment rather than passively receiving information. Knowledge construction occurs when learners engage in meaningful activities, make connections to prior knowledge, and reflect on their experiences.

The system's conversational interface enables learners to actively construct understanding through dialogue. When a student asks "How do I implement a binary search tree?", the system does not simply provide a code snippet. Instead, it engages the learner in a conversation that may include questions about their current understanding a step-by-step explanation that build upon prior knowledge with opportunities to explore related concepts which encourage them to experiment with code modifications.

The system maintains conversation history, which allowing each interaction to build upon previous exchanges. For example, if a student asks about arrays and later asks about array sorting algorithms, the system can reference the earlier conversation, helping learners see connections between concepts. This contextual continuity supports the constructivist view that learning is a cumulative process where new knowledge integrates with existing mental models.

2.3 Implementation Platform

C.A.T.E. was deployed across a robust and responsive development stack that ensures seamless operation, accessibility, and adaptability to educational needs. The frontend was developed using Next.js, a React-based framework, and styled with Tailwind CSS to provide a clean, minimal, and user-friendly interface that performs consistently across multiple devices and screen sizes (Jamshidi et al., 2025). On the backend, the system leverages Convex to manage session memory, user authentication, and learning history, enabling personalized and continuous support for each user, a critical factor in adaptive learning systems (Brusilovsky & Rus, 2019) LangChain and LangGraph are integrated to orchestrate large language model (LLM) operations, tool invocation, and real-time data retrieval, allowing the system to reason, respond, and adapt based on user input, aligning with advances in AI-driven education (Yao et al., 2023).

To enhance visual interactivity, the platform employs JavaScript libraries such as D3.js and Framer Motion. These tools dynamically generate visual representations, such as Abstract Syntax Trees (ASTs), control-flow diagrams, and parse trees, helping users better understand complex programming structures (Bostock et al., 2011). This implementation approach supports multimodal learning by integrating diverse tools effectively and provides an intuitive user experience, ultimately making C.A.T.E. a scalable, intelligent, and impactful AI-powered educational solution (Zawacki-Richter et al., 2019). The use of these technologies ensures that C.A.T.E. meets the scalability and interactivity demands of modern educational environments, as validated by recent studies on AI integration in higher education (Holmes et al., 2021).

2.4 Pilot Deployment

The pilot deployment of the C.A.T.E. multimodal AI agent was undertaken as a critical investigation into its pedagogical effectiveness in enhancing computer science education. The sample was recruited in the Faculty of Computing & Multimedia from Universiti Poly-Tech Malaysia (UPTM-FCOM) and the Faculty of Computer Science and Mathematics from Universiti Malaysia Terengganu (UMT-FSKM), respectively. Before its implementation, a well-organized pre-questionnaire survey was conducted, which included responses from 34 students. The survey results showed that 58.8% of the students were from UPTM-FCOM, and 41.2% were from UMT-FSKM. The students were in Semester 5-6, and 61.8% reported having a significant level of prior knowledge of AI. The results showed that the most common expectations included research assistance (50%), mathematical verification (44.1%), and an easy-to-use interface; thus, the use of GraphQL-based frameworks and a multi-modal validation approach was informed.

Further interviews with 31 participants also showed early but strong adoption. The percentage of people who used C.A.T.E. for less than one month is approximately 96.8%, highlighting the importance of onboarding materials that are easy to understand. The rate of interactions was also significant: 51.6% used the system every day or several times a day, and this trend was predetermined mainly by the necessity to find research material 48.4% of respondents found this feature to be the most useful) and mathematical assistance. The results of the post-questionnaire were surprisingly high, with 81.3% of respondents describing their experience as very good and 71% stating that they were delighted. Tool integration (average rating 4.35) and clear explanations (average rating 4.61) were highly rated, while concerns were raised about the limited availability of tools (32.3%) and poor user support (22%). These results led to repeated improvements, so that subsequent iterations will be better equipped to overcome such shortcomings.

Overall, the pilot demonstrated the influence of C.A.T.E. as a key academic collaborator for technical learners. 35.5% of the participants mentioned significant improvements in documentation, and 18.1% said that the time spent solving problems decreased. Therefore, the project aims to explore the extent to which multimodal AI technologies can support deep learning in various educational settings.

Table 1: Participant Demographics and AI Learning Needs

Demographic Category	Details
Total Participants	31 students
Institutions	UPTM-FCOM (58.8%), UMT-FSKM (41.2%)
Academic Semesters	Sem 5–6 (61.8%), Sem 3–4 (20.6%), Sem 7–8 (14.7%), Sem 1–2 (2.9%)
Specializations	Computer Science (61.8%), Network Security (20.6%), Software Engineering (11.8%), Data Science

Demographic Category	Details
	(2.9%), IT (2.9%)
AI Familiarity	88.6% had limited or no prior experience
Category	Value
Primary Needs	Research Support (50%), Mathematical Validation (44.1%), User Interface

Our pilot deployment statistics, combined with the interpretation of the analytical scope offered by Table 1, confirms the quantitative data already presented. The academic background of the cohort is also of interest: 61.8% of people were in their fifth or sixth semester, and 59.5% of people chose computer science as their primary area of interest. These numbers mean that the study falls within the target context of C.A.T.E.

It is equally enlightening to find that, despite this high level of preparedness, 88.6% of the respondents reported that they had little or no prior exposure to AI. This observation highlights an opportune niche that C.A.T.E. can fill in the technical literacy gap. Furthermore, the pre-deployment survey shows that the research-oriented tools (50%), mathematical validation (44.1%), and overall smooth user experience were the most desired features by the users, which influenced the GraphQL-based, multimodal architecture of C.A.T.E. substantially.

Table 2: User Experience and Technical Performance of C.A.T.E.

Metric	Result
Duration of Use	96.8% < 1 month
Daily Use	51.6% used daily or more
Most Useful Feature	Research Material Gathering (48.4%)
Overall Experience Rating	81.3% “Very Good”
Satisfaction Level	71% “Very Satisfied”
Average Tool Integration Score	4.35 / 5
Average Explanation Clarity Score	4.61 / 5
Improvements Noted	35.5% Better Documentation, 18.1% Faster Problem-Solving
Key Challenges	Tool Availability (32.3%), User Support (22%)

Table 2 gives a broader picture of the user interaction pattern and system performance. Although most of the respondents (96.8%) had just started using the platform, a significant percentage (54.8%) was already using the system daily, which is a good indication of initial engagement and usability. The process of research material collection turned out to be the most valued by the respondents, which once again confirmed the original results. The quality of the overall experience was also evaluated well: 81.3% considered it very good, and 71% were very satisfied. Quantitatively, the score of the integration was 4.35, and clarity of explanation was 4.61, which are signs of accurate instructional design. Although these are favorable signs, the users have cited various obstacles, including a lack of access to analytical tools (32.3%) and inadequate user support (22%). Such problems have resulted in subsequent developments in systems.

Taken together, the information above suggests that it is not only the potential of C.A.T.E. to be a high-impact academic assistant but also that it is incredibly adept at user-centered adaptation at a rapid pace. A significant increase in documentation practices (35.5%) and in the efficiency of problem-solving (18.1%) also proves the hypothesis that C.A.T.E. can be used to achieve more profound and independent learning. The study supports the case for the systematic implementation of multimodal technologies in higher education by matching the needs of students with AI-based interventions. In this regard, the research results will not only precondition future empirical investigations but will also promote a larger institutional interest in the implementation of adaptive AI-based solutions.

3 Results And Evaluation

3.1 Metrics

A post-deployment study of our computational-assistive tool, C.A.T.E., showed statistically significant improvements in various indicators of system usability, learning effectiveness, and user confidence. Remarkably, the scores of programming comprehension increased by 25.6% points in the pre-assessment and by 78.2% points after the system was utilized. The participants credited this significant increase to the step-by-step explanation, context-aware code suggestions, and automatic debugging capabilities of C.A.T.E., which made complicated concepts more palatable.

The educational efficacy was also supported by the fact that the number of compilation errors decreased by 34.5%, which was associated with the self-reported increase in code quality and accuracy by the users: the mean score of technical correctness increased by 0.43 on the 5-point Likert scale between the pre- and post-assessments. We also found that C.A.T.E. sped up and streamlined research. Approximately 48.4% of the students reported improved results from the study, especially among those who had previously identified information retrieval as a challenge.

The levels of confidence also increased. Before the deployment, a few of them were confident in their programming skills; after using C.A.T.E., 86% indicated that they had gained confidence, and 81.3% reported that their overall experience was excellent. The natural, conversational interface of the system was also rated well by students (average score: 4.29), which is why they were able to stay engaged and motivated.

Table 3: Summary of Learning and Technical Performance

Metric	Pre-Use	Post-Use	Change
Programming Comprehension Score	52.6%	78.2%	+25.6%
Compilation Error Frequency	High	Reduced by 34.5%	Significant improvement
Technical Accuracy (Rating)	4.18	4.61	+0.43
Mathematical Validation Accuracy	Low	Improved by 29.4%	Positive outcome
Research Effectiveness	50% struggled	+48.4% found improvement	Major impact

Table 3 illustrates the measurable academic and technical gains resulting from the post-deployment phase of C.A.T.E. Regarding quantifiable academic and technical performance, the statistics show a significant improvement in programming knowledge, increasing by 25.6% points. This stark change is indicative of the fact that the targeted scaffolding, i.e., structured explanations and context-sensitive assistance, was effective in alleviating the conceptual difficulties of students. The respective decrease of the compilation errors by 34.5% proves that the system not only reinforces theoretical skills but also improves the practical accuracy of coding.

Besides the technical precision, C.A.T.E. has been found to exert an observable influence on the perceived and actual quality of students' products. Technical accuracy improved by 0.43 on a 5-point Likert scale, which means the tool not only increases the level of performance but also the quality of deliverables. Noteworthy is the 29.4% increase in mathematical validation, which is one of the areas that was consistently noted as problematic in the pre-survey and during the interviews. The fact that C.A.T.E. is multimodal in its explanations seems to have made abstract logic and complex algorithms more transparent.

Attractive gains were also noted in the effectiveness of research. More than four out of ten (48.4%) of the participants reported an increase in effectiveness in locating and incorporating academic content, which highlights the usefulness of the tool to learners who previously struggled with the retrieval of information. AI-based query understanding and real-time resource connection have combined to change a time-consuming task into a quick and time-saving one. Overall, all of these statistics can make the role of C.A.T.E. as a transformational education technology tangible: not only to supplement the teaching process, but also to show that it proves a measurable, data-driven improvement in the student skill set.

3.2 Engagement And Tool Usage

C.A.T.E. significantly improved student engagement and frequency of interaction with technical content. The daily or frequent use rate was 14.7% before deployment, but after deployment, it increased to 51.6%. Students also cited tool integration and usability as key factors that contributed to this increase. The user-friendly interface and quick response time resulted in a 41.2% decrease in data processing time and a 44.1% increase in integration efficiency. Students also appreciated the ease of accessing external tools like YouTube, Google Books, and Wikipedia through the C.A.T.E. system; the mean satisfaction with integrated tool orchestration increased to 4.35 after using C.A.T.E.

Table 4: Engagement and Interaction Statistics

Aspect	Pre-Use	Post-Use	Change
Daily Usage Frequency	14.7%	51.6%	+36.9%
Research Tool Usage Efficiency	50% needed support	48.4% improvement	Strong impact
Integration Satisfaction Score	4.29	4.35	Slight improvement
Data Handling Time	High latency reported	Reduced by 41.2%	Noticeable performance gain

Table 4 provides a brief history of the changes in the engagement and patterns of interaction between students once they are deployed, specifically when it comes to C.A.T.E. The sheer growth in the number of times per day used, which peaked at 51.6% after rollout, but was at 14.7% before the implementation, would reflect a marked increase in sustained user activity. This kind of growth is directly linked with student feedback that focuses on the platform's usability and the system's quick response.

Upon further analysis of the data, it is possible to state that 48.4% of the respondents reported having achieved substantial increases in efficiency due to research-related activities, a finding that is in line with the successful coordination of the external resources like YouTube, Wikipedia, and Google Books by C.A.T.E. The satisfaction with tool integration in the system showed a similar, albeit slight, increase, with the average score moving up to 4.35 compared to 4.29, hence indicating an increased satisfaction with the user interface and system interoperability.

In addition, C.A.T.E. offered a reduction in the time required to process data of 41.2%, therefore considerably lowering latency and raising responsiveness to real-time processing. On the whole, the results support the twofold nature of C.A.T.E. as a tool of learning support and a tool of further interaction and academic production, which is an evolutionary investment in technical education.

3.3 Overall User Feedback and Satisfaction

The overall level of user satisfaction was very positive, and this response demonstrates the system's compatibility with student expectations. The overall satisfaction increased by 30%, or 71% after deployment, as opposed to the baseline of 42%. The most common strengths comprised interface clarity, learning support, and AI responsiveness. Most of the students responded that they would recommend C.A.T.E. to other people, thus having great word-of-mouth potential for adoption in the broader area of learning.

In addition, they found that the most helpful feature was step-by-step learning support, with 38.2% stating so, which directly met their demand for individual coding instructions. Minor tweaks were requested by users, including a broader range of programming languages and an extended Q&A capability.

3.4 Novelty of C.A.T.E.

C.A.T.E. represents a significant technology breakthrough in education, as it features aspects that existing large-language education systems lack. Unlike ChatGPT, Gemini, and Microsoft Copilot, which offer only textual answers to queries, C.A.T.E. can combine video transcripts, text, and code in real-time using a GraphQL orchestration plane. In this way, students have comfortable access to various resources, including YouTube tutorials, Google Books enrolments, and Wikipedia materials, all in one single interface.

Video transcripts processed, contextualized code examples, illustration models, and step-by-step explanations are all delivered to the student automatically by C.A.T.E., which meets the needs of 44.1 per cent of the students who needed increased mathematical assistance, and it also serves 41.2 per cent of the students, who needed a strong guide to data analysis. It is also bilingual (English and Malay), which is convenient for the 88.6% of users who have limited experience with such systems. C.A.T.E. is based on LangChain, LangGraph, and convex software and can be customized to a user based on their usage patterns and confidence levels, making it a niche student-oriented software as a service application in technical education.

Table 5: Comparative Feature Matrix: C.A.T.E. versus Leading Platforms

Feature	C.A.T.E.	ChatGPT	Gemini	Microsoft Copilot
Multimodal input (video transcripts and text)	Yes	No	No	No

Feature	C.A.T.E.	ChatGPT	Gemini	Microsoft Copilot
GraphQL-based tool orchestration	Yes	No	No	No
Automatic parsing of YouTube transcripts	Yes	No	No	No
Context-aware instructional support	Yes	No	No	No
Personalized session memory and adaptation	Yes	Partial	No	Partial

Placing the discussion in the broader context of the AI-based teaching platforms, C.A.T.E. stands out in the aspect of incorporating multimodal intelligence, contextual relevance, and user personalization. Although systems such as ChatGPT, Gemini, and Microsoft Copilot can be considered examples of highly developed general-purpose support, they do not meet all the requirements of technical education, as Table 5 shows. C.A.T.E. is the only framework that can deliver a comprehensive, unified framework that integrates the use of video transcripts, contextual code, and real-time education instruction. As opposed to other models, which either do not support such functionality at all or do so on a limited basis, C.A.T.E. can rely on GraphQL-based orchestration and dynamic transcript parsing, hence enabling the possibility of facilitating seamless, multi-resource learning, which is of particular importance to those learners who need to struggle with abstract and code-intensive material. Consequently, C.A.T.E. is not just another AI assistant but a specially developed guide about a more adequate understanding and processing of complex technical subjects in a comprehensible and self-assured manner.

3.5 Scalability

C.A.T.E. is engineered for both technical and pedagogical scalability. Its modular GraphQL architecture permits straightforward integration of new tools, programming languages, and external data sources, facilitating adoption by multiple academic institutions. Pilot deployments at UPTM's Faculty of Computing and Multimedia (58.8% of participants) and UMT's Faculty of Computer Science and Mathematics (41.2%) demonstrate its utility across distinct curricula.

Underpinned by a cloud-native stack (Convex, LangGraph), C.A.T.E. maintains performance under heavy concurrency. Post-survey data indicate a 51.6% increase in tool utilization during peak academic periods. Moreover, the platform's appeal spans disciplines: 61.8% of users were from Computer Science, 20.6% from Network Security, and the remainder from other technical fields. Planned enhancements, collaborative coding modules, live classroom integration, and proficiency tagging will further solidify C.A.T.E. as a robust and scalable platform. C.A.T.E. is engineered for both technical and pedagogical scalability. Its modular GraphQL architecture permits straightforward integration of new tools, programming languages, and external data sources, facilitating adoption by multiple academic institutions. Pilot deployments at UPTM's Faculty of Computing and Multimedia (58.8% of participants) and UMT's Faculty of Computer Science and Mathematics (41.2%) demonstrate its utility across distinct curricula.

Underpinned by a cloud-native stack (Convex, LangGraph), C.A.T.E. maintains performance under heavy concurrency. Post-survey data indicate a 51.6% increase in tool utilization during peak academic periods. Moreover, the platform's appeal spans disciplines: 61.8% of users were from Computer Science, 20.6% from Network Security, and the remainder from other technical fields. Planned enhancements, collaborative coding modules, live classroom integration, and proficiency tagging will

further solidify C.A.T.E. as a robust and scalable educational resource.

Table 6: Cross-Discipline Adoption and Tool Utilization

Discipline	Percentage of Participants	Primary Tools Utilized
Computer Science	61.8 %	Syntax Debugger; AST Visualizer
Network Security	20.6 %	Real-time Validation; Data Processing
Software Engineering	11.8 %	Code Optimization Assistant; Reference Search
Other (Data Science, IT)	5.8 %	Mixed-mode Transcript Analysis

As depicted in Table 6, its scalability is also supported by the cross-disciplinary adoption and heterogeneous tool usage of the computational environment C.A.T.E. Most of the users (61.8%) are computer science students, and they are attracted to such features as Syntax Debugger and AST Visualizer that are highly relevant to their primary learning outcomes. The participants in Network Security (20.6%) were more involved in real-time validation and data-processing capabilities, and hence confirm the versatility of C.A.T.E. to handle particular, domain-specific requirements.

The Code Optimization Assistant and Reference Search were heavily utilized by software engineering students (11.8%), indicating that the system can be used to facilitate more complex debugging and knowledge discovery. Although a smaller group, Data Science and IT participants (5.8%) utilized mixed-mode transcript analysis, and this shows the versatility of C.A.T.E., even in the emerging discipline of technology. The distribution not only helps prove the modularity of the environment but also enhances the ability of the environment to scale accordingly, depending on the various areas of education.

Arriving at the institutional implications of such adoption patterns, such adoption patterns confirm that C.A.T.E. is not restricted to any specific domain of academic interest. Instead, it creates itself as a flexible infrastructure that can fit the evolving demands of students and teachers in many areas.

3.6 Challenges

Several obstacles must be overcome to achieve sustained adoption of C.A.T.E. Notably, 32.3% of respondents reported experiencing tool unavailability during periods of high demand, which is why load balancing and automatic scaling of backend services are necessary. Moreover, 22.0% of respondents experienced a lack of initial onboarding. Since 88.6% of respondents were not well-exposed to similar systems, this indicates that extensive tutorials and guided tours are necessary.

Another issue is to ensure high response accuracy. Although the mean technical confidence rating was 4.16 out of 5, 14.7% of respondents encountered problems with input validation and formatted prompts. With new languages and tools being added, it will be crucial to conduct comprehensive testing of compatibility to ensure the integrity of the orchestration workflow. Lastly, to ensure adherence to academic integrity principles, especially regarding automated code generation, monitoring of academic integrity rules will be necessary, and content moderation measures may be required.

Table 7: Reported Challenges and Proposed Mitigations

Challenge Area	Reported by Users (%)	Proposed Mitigation
Tool unavailability during peak usage	32.3 %	Implement load balancing and auto-scaling
Insufficient onboarding and user guidance	22.0 %	Develop interactive tutorials and guided tours
Difficulties with input validation	14.7 %	Add input-helper modules and auto-correction
Low prior familiarity (onboarding requirement)	88.6 %	Introduce context-sensitive help and FAQs

Table 7 describes the significant technical and user-experience problems identified during the pilot implementation of the C.A.T.E. Tool. Unavailability caused by peak load (32.3%), which is the most frequent issue, reveals that effective load-balancing procedures and auto-scaling should be developed to ensure that the performance is not affected. The lack of proper initial onboarding support was also cited by a large proportion (22.0%) of users, indicating the need for intuitive, interactive tutorials to reduce the initial learning curve.

Although the overall technical confidence rating was high, the percentage of users who experienced issues with input validation and formatting was 14.7%, highlighting the need for intelligent input helpers and real-time correction. Most importantly, possibly, the vast majority (88.6%) of the participants showed low exposure to AI-powered platforms of this kind, which speaks to the importance of incorporating context-aware guidance tools to facilitate the entry into the system of first-time users. These issues, despite being significant, have straightforward avenues of improvement, so that the scalability of C.A.T.E. is accompanied by its ease of use.

With the evolution of the platform, these concerns will play a significant role in making its adoption as wide and sustainable as possible in the various educational settings.

4 Discussions

Post-questionnaire qualitative responses provided insight into the strengths of the system, as well as areas for improvement. Among the most eulogized features were syntax debugging, whereby 91% of the students polled reported a positive experience, a trait symbolizing C.A.T.E. potential in helping to decode real-time error-fixation and design superior coding designs. The same can be said for visual aids, which were overwhelmingly well-rated, with 87% rating the interactive visualization as necessary, a number that also specifically mentioned data visualization in learning as a crucial aspect of learning. The ability to use English and Malay as input languages also scored highly in terms of user satisfaction, with 84% of users enjoying the experience of having their queries in either language field accepted by the system. The experience has helped cement the inclusivity and accessibility of the system.

The functions that have been reported by 48.4% of users as the most useful are research support, which aligns with the results of the pre-questionnaire, where 50% recognized it as a significant obstacle to learning. Straightforward explanations formed yet another pillar of the system's value, with 81.3% of people being satisfied, and the pre-use rating average standing at 4.21, while the post-use clarity average was 4.61. Nevertheless, specific issues were also reported by the students. Specifically, 32.3% complained about the lack of availability of specific tools at a time when a large number of students were trying to use them, and 22% said they needed more user support. These have been used as direct inputs in development priorities in future releases to ensure consistency and enhanced product development.

Taken together, the assessment found that C.A.T.E. did more than address the most significant pain points those technical students encounter; it provided noticeable results and growth in understanding, performance, and user enjoyment. With modular, multimodal, and user-centered design, it sets a precedent for scalable digital tools that operate through AI-powered devices and systems, thereby justifying its mention as an award-winning innovation in intelligent learning systems.

5 Conclusions

The idea and realization of C.A.T.E. represent a conclusive step toward integrating artificial intelligence into technical education. Addressing the shortcomings typically attributed to traditional AI solutions, C.A.T.E. has evolved into a purposefully designed pedagogical agent, offering bilingual, multimodal, and context-sensitive instructional support. Its ability to organize tools in real-time through GraphQL, inspect video transcripts, debug code, and provide structured visual feedback covers the primary educational needs of Malaysian computer science students. The positive result of the pilot phase confirms its importance as a means of bridging the gap between AI literacy and access to teaching materials in higher education.

The quantitative measurement provides measurable improvements in user understanding, problem-solving speed, and self-confidence. The ability to program improved by over 25%, errors during compilation were significantly reduced, and overall satisfaction exceeded 70%. Students admitted that C.A.T.E. was very interesting and beneficial, especially when it comes to helping with research and mathematical confirmation, which are two areas also recognized as long-term challenges. In addition, a positive trend in daily usage has been observed, increasing from 14.7% before deployment to 51.6% after, and strong integration scores attest to the platform's high levels of pedagogical significance and user fit.

Although these strengths are apparent, the success of C.A.T.E. in the future depends on overcoming the challenges linked to scalability, onboarding, and academic integrity. However, the structure established by this project lays the groundwork for a scalable, inclusive AI-based education system. With iterative refinements in the works, C.A.T.E. is poised to redesign digital learning experiences in Malaysia and, by extension, in every location where technical education requires support with adaptive, intelligent, and accessible resources. Its successes precondition innovations in the sphere of AI-based academic settings.

6 Acknowledgement

We sincerely thank Majlis Amanah Rakyat (MARA) for the financial support provided through the Tertiary Education Sponsorship Programme (TESP), which played a significant role in supporting the academic journey of our co-author, Ummi Haani Shahrul, during her Diploma in Computer Science at Universiti Poly-Tech Malaysia (UPTM). We also extend our appreciation to the Faculty of Computing & Multimedia (FCOM), UPTM, for their continuous support, guidance, and resources that contributed to the successful completion of this research project.

References

Baker, R., Boser, U., Jones, L., Lee, S., Ocumpaugh, J., Sieve, M., & Cheng, Y. (2021). Transforming educational technology through convergence: Conference report. University of Pennsylvania.

https://learninganalytics.upenn.edu/ConvergenceWhitePaper.pdf
Bostock, M., Ogievetsky, V., & Heer, J. (2011). D3 data-driven documents. IEEE Transactions on Visualization and Computer Graphics, 17(12), 2301–2309. https://doi.org/10.1109/TVCG.2011.185

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., & Hesse, C. (2020). Language models are few-shot learners. Advances in Neural Information Processing Systems, 33, 1877–1901. https://proceedings.neurips.cc

Brusilovsky, P., & Rus, V. (2019). Social navigation for self-improving intelligent educational systems. International Journal of Artificial Intelligence in Education. https://www.researchgate.net/publication/347506849

Chen, W., Su, Y., Zuo, J., Yang, C., Yuan, C., Qian, C., Chan, C.-M., Qin, Y., Lu, Y., Xie, R., Liu, Z., Sun, M., & Zhou, J. (2023). AgentVerse: Facilitating multi-agent collaboration and exploring emergent behaviors. arXiv. https://arxiv.org

Gat, E. (1998). On three-layer architectures. Artificial Intelligence and Mobile Robots. https://u.cs.biu.ac.il/~kaminkg

Holmes, W., Porayska-Pomsta, K., Holstein, K., Sutherland, E., Baker, T., Shum, S. B., Santos, O. C., Rodrigo, M. T., Cukurova, M., Bittencourt, I. I., & Koedinger, K. R. (2021). Ethics of AI in education: Towards a community-wide framework. International Journal of Artificial Intelligence in Education, 32(1), 504–526. https://doi.org/10.1007/s40593-021-00239-1

Holstein, K., McLaren, B. M., & Aleven, V. (2019). Designing for complementarity: Teacher and student needs for orchestration support in AI-enhanced classrooms. Lecture Notes in Computer Science, 11625, 157–171. https://doi.org/10.1007/978-3-030-23204-7_14

Jamshidi, S., Nikanjam, A., Nafi, K. W., Khomh, F., & Rasta, R. (2025). Application of deep reinforcement learning for intrusion detection in the Internet of Things: A systematic review. Internet of Things, 31, 101531.
https://doi.org/10.1016/j.iot.2025.101531

Kiela, D., Bhooshan, S., Firooz, H., Perez, E., & Testuggine, D. (2020). Supervised multimodal bitransformers for classifying images and text. arXiv. https://doi.org/10.48550/arXiv.1909.02950

Luckin, R., & Holmes, W. (2016). Intelligence unleashed: An argument for AI in education. Pearson. https://discovery.ucl.ac.uk

Nguyen, T. N. Q., & Nguyen, T. T. (2025). AI literacy in language learning through digital multimodal composing: A systematic review. VNU Journal of Foreign Studies, 41(3), 142–156. https://doi.org/10.63023/2525-2445/jfs.ulis.5517

OpenAI. (2023). GPT-4 technical report. arXiv.
https://doi.org/10.48550/arXiv.2303.08774

Piaget, J. (1952). The origin of intelligence in the child. International Universities Press. https://doi.org/10.4324/9781315006260

Russell, S. J., & Norvig, P. (2009). Artificial intelligence: A modern approach (3rd ed.). Prentice Hall.

Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., & Yao, S. (2023). Reflexion: Language agents with verbal reinforcement learning. Advances in Neural Information

Processing Systems, 36. https://proceedings.neurips.cc

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems. https://arxiv.org/abs/1706.03762

Vygotsky, L. S. (1978). Mind in society: The development of higher psychological processes. Harvard University Press. https://books.google.com

Wooldridge, M. (2025). An introduction to multi-agent systems (2nd ed.). Wiley. https://books.google.com

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2023). ReAct: Synergizing reasoning and acting in language models. Proceedings of the International Conference on Learning Representations. https://par.nsf.gov

Zawacki-Richter, O., Marín, 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), 1–27. https://doi.org/10.1186/s41239-019-0171-0