

**Artificial Intelligence in Personalized Education: An  
Exploration of Current Intelligent Tutoring Systems  
and the Design of an Enhanced Interactive  
Programming Tutor**

**Ejiroghene Philip Wilkie**

Submitted to the  
Institute of Graduate Studies and Research  
in partial fulfillment of the requirements for the degree of

Master of Technology  
in  
Information Technology

Eastern Mediterranean University  
September 2023  
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

---

Prof. Dr. Ali Hakan Ulusoy  
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Technology in Information Technology.

---

Asst. Prof. Dr. Ece Çelik  
Director, School of Computing and  
Technology

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Technology in Information Technology.

---

Asst. Prof. Dr. Hüsnü Bayramoğlu  
Supervisor

---

Examining Committee

1. Assoc. Prof. Dr. Emre Özen

2. Asst. Prof. Dr. Hüsnü Bayramoğlu

3. Asst. Prof. Dr. Fatma Tansu Hocanın

## **ABSTRACT**

This thesis aims to explore the influence and advancements of Artificial Intelligence (AI) in education, particularly focusing on Intelligent Tutoring Systems (ITS) for programming education. A comprehensive review of existing ITS models will be carried out, examining their implications on academic performance, student engagement, and motivation. Building upon this analysis, the research will conceptualize and design an innovative ITS, prioritizing interactivity, enhanced code evaluation, and accessibility. While empirical testing of this tool will not be within the scope of this study, a theoretical comparison against existing non-programming chatbot tutors and other ITS will be made. This work is expected to provide substantial contributions to educators, policymakers, and researchers interested in the intersection of AI and programming education.

**Keywords:** Artificial Intelligence, Intelligent Tutoring Systems, Programming Education, Interactive Learning, Educational Technology, Student Outcomes.

## ÖZ

Bu tez, Yapay Zekânın (YZ) eğitimdeki etkisi ve ilerlemelerini, özellikle programlama eğitimi için Akıllı Öğretim Sistemleri (AÖS) üzerine yoğunlaşarak incelemeyi amaçlamaktadır. Mevcut AÖS modellerinin kapsamlı bir incelemesi yapılacak, akademik performans, öğrenci katılımı ve motivasyon üzerindeki etkileri incelenecektir. Bu analizden yola çıkarak, araştırma, etkileşimliliği, gelişmiş kod değerlendirmeyi ve erişilebilirliği önceliklendiren yenilikçi bir AÖS tasarlayacak ve kavramsallaştıracaktır. Bu aracın ampirik testi bu çalışmanın kapsamı dışında olacak olsa da, mevcut programlama dışı chatbot öğretmenler ve diğer AÖS ile teorik bir karşılaştırma yapılacaktır. Bu çalışmanın, YZ ve programlama eğitiminin kesişimine ilgi duyan eğitimciler, politika yapıcılar ve araştırmacılara önemli katkılar sağlaması beklenmektedir.

**Anahtar Kelimeler:** Yapay Zeka, Akıllı Öğretim Sistemleri, Programlama Eğitimi, Etkileşimli Öğrenme, Eğitim Teknolojisi, Öğrenci Sonuçları.

... *Dedicated to K*

To my dearest family, to whom I owe the world and have always been there when I  
needed them.

## **ACKNOWLEDGMENTS**

First and foremost, I would like to express my deepest gratitude to my thesis advisor, Assist. Prof. Dr. Hüsnü Bayramoglu for his unwavering support, guidance, and invaluable insights throughout the course of this research. Your expertise and commitment to my academic and personal growth have been instrumental in shaping this work. Special thanks to the countless researchers and authors whose work laid the foundation for this thesis. I'd also like to thank my peers and colleagues, especially Danial Shafaei who sits across from me as I'm writing this, for the enlightening discussions, camaraderie, and always being ready to make a cookie run with me. And lastly, I am profoundly grateful to my family, for their unyielding love, patience, and encouragement.

# TABLE OF CONTENTS

ABSTRACT .....	iii
ÖZ.....	iv
DEDICATION .....	v
ACKNOWLEDGMENTS .....	vi
LIST OF FIGURES .....	ix
LIST OF ABBREVIATIONS.....	x
1 INTRODUCTION .....	1
1.1 Problem Statement .....	1
1.2 Objectives of the Study .....	1
1.3 Research Questions .....	2
1.4 Significance of the Study .....	2
1.5 Scope of the Study .....	3
1.6 Methodology Overview.....	3
2 LITERATURE REVIEW .....	4
2.1 Review of Existing ITS Models .....	6
2.2 ITS in Programming Education .....	7
2.3 Impact of ITS on Student Outcomes .....	8
2.4 Limitations and Challenges of Current Systems.....	10
3 THE PROPOSED ITS DESIGN .....	15
3.1 Architecture of the Proposed ITS .....	18
3.2 Intelligent Tutoring Models .....	19
3.3 ITS Flow and User experience .....	21
3.4 AI Features and Capabilities .....	23
3.5 Improvements over previous systems .....	26

4 ITS COMPARISON RESULTS .....	29
4.1 Methodology for Comparisons .....	29
4.2 Comparison Parameters.....	29
4.3 Comparative Analysis with Existing ITS Models .....	31
4.4 Discussion of Findings .....	32
5 DISCUSSION AND CONCLUSION.....	35
REFERENCES .....	37

## LIST OF FIGURES

Figure 3.1: Chat interface of the proposed ITS .....	21
Figure 3.2: New session interface of the proposed ITS .....	22
Figure 3.3: Welcome screen interface of the proposed ITS .....	23
Figure 3.4: Wrong solution attempt .....	24
Figure 3.5: Wrong solution attempt 2 .....	24
Figure 3.6: Tutor explanation .....	25

## **LIST OF ABBREVIATIONS**

AI	Artificial Intelligence
AIED	Artificial Intelligence in Education
API	Application Programming Interface
CA	Conversational Agents
CBR	Case Based Retrieval
EDM	Educational Data Mining
GPT-3	Generative Pre-trained Transformer 3
GPT-4	Generative Pre-trained Transformer 4
ITS	Intelligent Tutoring System
LA	Learning Analytics
LLM	Large Language Model
ML	Machine Learning
NLP	Natural Language Processing
STT	Speech-to-Text
TTS	Text-to-Speech

# Chapter 1

## INTRODUCTION

Artificial Intelligence (AI) is emerging as an exciting technology with the potential to completely transform several industries, including education. Traditional educational approaches often struggle to cater to the needs of individual students, such as diverse learning styles, different paces of learning, individual interests and motivations, varied backgrounds and prior knowledge, and special educational needs [1, 2]. Intelligent Tutoring Systems (ITS), particularly those designed to teach programming, bring forth the possibility of personalized and adaptive educational experiences that can significantly enhance student outcomes.

### 1.1 Problem Statement

Conventional educational practices face challenges in providing individualized instruction to students due to constraints such as resource limitations and restricted teacher capacity. This often leads to a one-size-fits-all approach that fails to effectively address the unique needs and interests of each learner. The integration of ITS in programming education presents an opportunity to circumvent these limitations and deliver tailored learning experiences that better engage and motivate students.

### 1.2 Objectives of the Study

The primary objective of this thesis is to investigate the impact and potential of AI in the field of education, with a specific focus on ITS for programming education. The study aims to achieve the following objectives:

- Conduct a comprehensive review and analysis of existing ITS to examine their effects on academic achievement, student engagement, and motivation.
- Explore the conceptualization and design of a novel ITS, presenting a detailed prototype that incorporates potential benefits and theoretical effectiveness.
- Design and implement a novel Intelligent Tutoring System aimed at addressing the identified limitations and enhancing interactivity, accessibility, and reliability in programming education.
- Enhance our understanding of AI's role in education, particularly in programming education, laying the groundwork for future practical implementations and research endeavors.

### **1.3 Research Questions**

The following research questions will be explored to guide the study and address the objectives mentioned above:

1. What is the impact of ITS on student outcomes, including academic achievement, student engagement, and motivation?
2. What are the challenges and limitations associated with current ITS for programming education?
3. How can a novel ITS be conceptualized, designed, and theoretically evaluated to enhance student outcomes and address the limitations of existing models?
4. What will be the resultant features and benefits of a novel Intelligent Tutoring System designed and implemented to enhance interactivity, accessibility, and reliability in programming education?

### **1.4 Significance of the Study**

This thesis is expected to offer valuable insights to educators, policymakers, and researchers interested in the crossroads of AI, education, and programming. By examining the impact and potential of ITS, the study can guide the development of

effective teaching methodologies and tools. The findings may aid in designing and implementing innovative ITS, thereby supporting the progression of programming education and enhancing student learning outcomes.

### **1.5 Scope of the Study**

This research will primarily focus on the exploration and analysis of existing ITS. While a novel ITS will be conceptualized and designed, its testing and evaluation will be beyond the scope of this study. The research will encompass a comprehensive literature review, examination of existing models, and the theoretical evaluation of the proposed tool to pave the way for future practical implementations and research undertakings.

### **1.6 Methodology Overview**

The study will employ a systematic methodology to fulfill its objectives. A literature review will be conducted to acquire an overview of the current state of AI in education, specifically its role in ITS for programming education. Data collection will involve procuring information on these models and their effects on student outcomes. Various analytical methods, such as qualitative and quantitative techniques, will be used to interpret the data and extract insights. Ethical considerations regarding the use of AI technologies in education will be addressed throughout the research process to ensure responsible and equitable practices.

## Chapter 2

### LITERATURE REVIEW

The field of education has witnessed a rapid expansion of AI in recent years, particularly within the last five years [3]. AI has taken on increasingly significant roles in instruction, learning, and decision-making [4, 5]. It has the ability to automate learning resources, provide automatic assessments, and even predict student performance [6–8]. The application of AI in education, including personalized instruction strategies, has unlocked new possibilities and challenges for the transformation of higher education [9].

Emerging AI applications are now shaping how we understand and approach primary and secondary education [10]. The convergence of Artificial Intelligence in Education (AIED), Educational Data Mining (EDM), and Learning Analytics (LA) is reshaping the instructional landscape to be increasingly personalized, dynamic, and responsive to learners' needs [11]. These transformative technologies are not confined to classroom instruction but have also penetrated online learning, a sector that has shown a remarkable increase in AI adoption from 2010 to 2021 [12]. A substantial portion of this rise can be attributed to the implementation of AI in a range of learning support tools such as ITS [13]. These systems, by leveraging machine learning, offer a high degree of personalization and adaptation to students' individual learning patterns and have thus revolutionized the teaching and learning experience.

A significant manifestation of this transformation is the ITS [13]. ITS leverage the

power of machine learning to offer highly personalized and adaptive learning experiences, revolutionizing traditional teaching and learning methods. Furthermore, AI applications, including AIED, EDM, and LA, are increasingly molding the way we understand and approach primary, secondary, and higher education [10].

Several studies have been carried out to capture insights into the significant applications of ITS. A systematic review of AI and chatbot integration in education, emphasized its potential for personalized learning [14]. Another study demonstrates how AI technologies, particularly large language models(LLM) such as GPT-3, coupled with gamification, can support the scaffolding of programming learning for international students. This results in an increased sense of belonging and reduced their apprehension towards programming lessons [15]. In another innovative research an ITS that utilizes dynamic graphic visualizations, making abstract programming concepts more tangible and understandable to learners was introduced [16]. The 'Dodona' project, detailed in another work, connects learning and assessment through real-time data and feedback, providing a practical example of an ITS acting as a 'virtual co-teacher' [17]. An overview of the current evidence-based tutoring chatbot approaches and their use in teaching, also highlighted the necessity for continuous improvements in chatbot interaction design and usability [18].

In light of these studies, it's apparent that AI and ITS play a transformational role in enhancing learning experiences, particularly in the field of programming education. However, it's also evident that there's an ongoing need for improvement in chatbot design and usability. As technology continues to evolve, it is vital that we stay ahead with comprehensive reviews and analyses of the implications of AI to maximize its benefits and address potential drawbacks. It is clear that the future of education will be

heavily influenced by our capacity to refine these systems, to harness them effectively to enhance educational experiences and outcomes.

## **2.1 Review of Existing ITS Models**

The landscape of ITS is continuously evolving, and different types of systems have been designed to improve educational outcomes through innovative approaches. For example, a rule-based chatbot was built to emulate the experience of in-person math instruction by offering explanations, practice questions, and tailored feedback. The chatbot demonstrated a significant enhancement over traditional learning methods, such as Khan Academy's video-based approach [19] and introduced a contextual bandit algorithm to personalize the pace of the lesson effectively addresses varying user needs and preferences. Similarly, An adaptive and personalized tutoring system 'Seis Tutor' was built and compared with a traditional tutoring system. The work showed that blending cognitive intelligence with personalization and adaptation features in an ITS can significantly enhance the learning experience [20]. The findings emphasize the need for more individualized, learner-centric approaches in the realm of ITS.

Other work has also been done, including an innovative chatbot, designed with a reactive nature, acting as an effective medium for user interaction across platforms [7]. This work approached chatbot design by blending gamification and adaptive learning practices, suggesting the potential of such systems in encouraging a playful yet effective learning environment. Another system that was presented was MetaTutor, a hypermedia-based ITS that helps foster and measure Self-regulated learning (SRL) among learners [21]. The research here also underscored the crucial role of pedagogical agents in scaffolding cognitive and metacognitive SRL strategies. By leveraging multimodal data, the study offers insights into identifying the

significance of cognitive, affective, metacognitive, and motivational (CAMM) processes in SRL, revealing the limitations and potential of real-time instructional interventions.

Despite the above advancements, the work is far from complete. One area where these systems are being increasingly deployed, and which shows considerable promise, is in the realm of programming education.

## **2.2 ITS in Programming Education**

The integration of ITS in programming education, as demonstrated by the influx of students in Computer Science (CS) and related courses, is an emerging area of interest in contemporary research [22]. These systems, being capable of offering personalized and adaptive learning environments, have shown to be advantageous for enhancing the educational experience. To this end research has been done to develop ITS for programming education.

In one of these studies, "Pyo" a chatbot created to assist beginners in programming. The study emphasized the crucial role chatbots play by providing constant support to a large amount of students. In addition to this, the chatbot encourages introverted students to ask questions freely. Despite Pyo proving beneficial to learning outcomes, the authors acknowledge room for improvement, particularly in providing more direct functionality explanations [22]. In another novel approach, an ITS is proposed for the exclusive use in online programming courses, integrating advanced algorithms like the optimized ant colony algorithm for educational support. This proposed system demonstrates that ITS can provide students with real-time, insightful feedback, thereby enhancing their understanding of complex programming principles [23]. Similarly, an asynchronous web-based ITS for C programming language was

designed to maintain student motivation during the learning process. The system used interactive motivational tactics and rules to assess and uphold student motivation and was found to be satisfactory in design, usability, functionality, and compatibility through an evaluation study. This highlights the essential role of motivation in asynchronous learning systems, suggesting that the interactive nature of an ITS can serve as a solution for maintaining student motivation throughout the learning process [24]. Further innovation in ITS was achieved by using graphic representations to aid programming learning. By leveraging 3D graphics in an augmented reality environment, the authors attempted to make abstract programming concepts become more tangible, promoting a more profound understanding [16].

In a somewhat different approach made possible by recent breakthroughs in LLMs, Cao uses LLMs in tandem with gamification to assist Chinese students in introductory programming courses. His work suggests that a gamified ITS, amplified by AI models like GPT-3, can considerably alleviate student anxiety and promote a sense of belonging, thus improving learning outcomes in programming [15]. A similar approach showcases the prowess of an AI assistant, ChatGPT, in problem-solving across a diverse range of undergraduate CS topics. However, it also raises concerns about its potential misuse by students [25].

### **2.3 Impact of ITS on Student Outcomes**

The impact of ITS on student outcomes and engagement is an area of considerable interest and exploration in the current body of literature. Several studies have investigated this, adopting a range of methodologies and examining varied instructional contexts.

One such study analyzed the effect of an AI-enabled chatbot in a 5th-grade science

course, and found no direct significant enhancement in students' academic achievements, but the application of the chatbot proved to influence positively the experimental group's online learning experience [26]. Another study took a different angle, investigating students' continuance intention to use ITS for English learning through an extended technology acceptance model (TAM) [27]. They discovered several factors that significantly contribute to the continued intention of using ITS. This included perceived usefulness, price value, learning goal orientation, and facilitating conditions. Interestingly, emotional factors such as enjoyment positively predicted the use of ITS, whereas anxiety negatively predicted its use. These findings underscore the importance of students' perceptions of ITS and the role of affective factors in the successful adoption of these systems.

In the context of chatbots, a study to investigate the efficacy of a dialogue-based ITS concluded that student performance under the instruction of the ITS was comparable to conventional instruction and was significantly superior to self-learning through material reading [28]. Additionally, the ITS increased learner motivation, hinting at the potential of ITS to engage students effectively and enhance their learning experiences. Similarly, a pioneering study within the Ghanaian higher education context explored the impact of a chatbot or virtual teaching assistant on student learning [29]. Impressively, the research demonstrated that students interacting with the chatbot outperformed those who received traditional instruction from a course instructor. Another research also provided an overview of how students and tutors perceive ITS in an online learning context. While the study revealed that ITS could potentially enhance the online learning experience, it also surfaced several concerns that need addressing, underlining the importance of considering potential challenges and barriers when integrating ITS into the learning environment.

Students, the ultimate beneficiaries of AI-driven education, have shown a similar trend of acceptance. The adoption of generative AI (GenAI) technologies in higher education has been generally positive, as students perceive these technologies as valuable tools for enhancing their learning experiences [30].

## **2.4 Limitations and Challenges of Current Systems**

As AI and ITS continue to expand into the educational sector, a variety of ethical considerations, challenges, and limitations surface which demand careful examination and navigation [31].

A pressing limitation of current ITS is the lack of comprehensive voice support. Voice chat is a feature that has been shown to significantly improve accessibility, as well as user experience and satisfaction. Despite the evident advantages, many current ITS fall short in integrating voice-support functionalities. The absence of this feature restricts the potential of ITS to deliver a fully immersive and inclusive learning experience. Voice chat offers a more immediate and intuitive way for users to acquire and process information. For example, a study highlighted the dynamic appeal of speech-based dialogues, which is even more suitable for situations that require multitasking or for other potential users who may find text-based interactions more tedious [32]. Additionally, information presented with the aid voice format is often perceived as more credible and trustworthy than when its just words on a screen. Systems possessing voice capabilities can break down barriers, especially for individuals who may be less familiar or engaged with a topic, thereby offering a more inclusive and accessible learning experience [33].

The growing ubiquity of conversational agents (CAs) in various domains underscores the potential benefits of voice-based interaction. Defined as systems that allow users

to communicate using natural language, CAs have become increasingly sophisticated thanks to advancements in artificial intelligence and natural language processing [34]. Furthermore, CAs have progressed significantly from their rudimentary versions and now possess sophisticated generative models, powered by technical advancements such as large language models and statistical computing. These agents have seen diverse applications, from personal spheres to areas like healthcare, business and most relevantly education reflecting their versatility and user-centric advantages [35].

However, while voice systems have evolved, their true potential in supporting complex dialogues and maintaining conversational context has often been underutilized. Found that most voice-based search systems are not tailored for extended natural interactions [36]. This limitation forces users to restrict their voice-based inquiries to simple navigational tasks. By integrating improved conversational mechanisms in ITS, it's plausible to enhance user satisfaction and streamline the learning process. As voice support systems become increasingly sophisticated, their interactions can mimic natural human conversations, making them more engaging and user-friendly. Additionally, these enhancements provide offerings that users have been shown to find important. These include increased efficiency, convenience, ease of use, and the reduced cognitive effort they require [37].

Introducing voice support in ITS is not without challenges. Voice and other AI enabled applications necessitate the collection of extensive data, often involving sensitive student information. While this collection can significantly enhance personalized learning experiences, it concurrently raises questions of data security and privacy, the potential for data misuse, and an imperative need for robust data protection measures, especially for younger learners [38, 39]. Alongside privacy, the

capacity for personalization presents a double-edged sword. On one hand, it can foster targeted and efficient learning experiences, but on the other, it may encroach upon student autonomy. The key challenge here is to strike a balance between technology-assisted learning and human intervention. This balance is crucial to ensure that the development of self-directed learners is encouraged and not inhibited by an over-reliance on technology [31,40]. Equity and accessibility also emerge as pivotal concerns in the context of ITS in education. ITS holds significant potential to enhance educational equity, but it also risks intensifying the digital divide. There is a possibility that the benefits offered might be disproportionately distributed, thereby inadvertently amplifying existing educational inequities. This is especially evident when affluent students have greater access to advanced tools compared to their less privileged counterparts [41,42].

In-depth research has also been conducted into the specific limitations of ITS. An assessment of 36 papers revealed an understanding of the recent interaction design techniques between chatbots and learners in the educational realm and identified key challenges, particularly inadequate dataset training and a noticeable lack of usability heuristics in current systems [18]. The study advocated the need to delve deeper into the improvements that personalization and localization of chatbots provide to increase learning effectiveness and student satisfaction. In another paper, the effectiveness of technology-based curricula and adaptive learning systems as supplements to conventional teaching methods in the realm of mathematical skill development was scrutinized [43]. The findings revealed a mixed level of adherence to the core components of such systems and a generally limited ability of the software to facilitate personalized instruction, underscoring a disconnection between the software and existing curricula, as they rarely inform each other, leading the authors to argue

for improved implementation models and better strategies for enabling technology-facilitated personalization.

Similar limitations can be observed in the ITS recently developed for use in programming education. Issues with dataset availability and reproducibility, a gap in understanding the relationship between AI techniques and ITS data, demonstrate an overall need for more comprehensive research on ITS for programming [44]. For instance Pyo, a chatbot assistant for introductory programming students was observed to facilitate student learning, but was evaluated to have a need for more straightforward explanations of its functionalities [22]. Educational chatbots using existing LLMs ability to handle multiple interconnected learning objectives and adapt to diverse user characteristics suggests great potential, but this approach isn't without limitations [45]. The most notable among being the constraints of sustaining a long conversational history and the rigid structure of responses that may occasionally prompt unexpected role transitions for the chatbot. A ITS based on fuzzy logic to provide programming tutoring was also proposed. The study revealed that while the fuzzy-based ITS can increase satisfaction of the learners and help them learn, it needs to improve the system's recommendation accuracy, the efficiency of its interactions, and improve engagement in the learning process [46]. Dodona, an ITS designed to aid learning of computer programming, similarly offers real-time data and feedback to enhance students' learning. However, it has faced certain practical challenges in fostering active learning and supporting challenge-based education, thus indicating the need for further improvements [17]. Another Programming-Tutor with core focus on user interaction was developed. Although the proposed ITS is expected to provide high-quality immediate feedback to students and potentially increase their learning efficiency, the systems reliance on a moodle and Case Based Retrieval (CBR) limits

the breadth of topics and feedback the student can explore [47].

Work done on analyzed data and modeling of beginner level C programming error logs exposed a limitation in current ITS models, as it suggested that many ITS models overlook other important aspects of programming [48]. Similar to this is Trace Table Tutor (T3), was built specifically to support effective learning of code tracing [49], and INTTuToR was built to correct API misuse by providing instant feedback [50]. These systems were designed to focus on a specific aspect, and while these aspects have been overlooked, these systems lack capabilities to handle problems outside of that specific area.

These studies show that the successful utilization of ITS in education requires considering the ethical dilemmas as well as technical limitations of current ITS. While these tools provide incredible opportunities for personalized learning and enhanced accessibility, they simultaneously introduce possible issues related to privacy, autonomy, equitable access, and issues of dataset availability and reproducibility [30]. Equally significant are the challenges of ensuring a balance between technology-assisted and human-led learning, the potential to exacerbate existing educational inequities, and ensuring that software informs and aligns with existing curricula. The readiness and flexibility of educators to effectively implement these sophisticated technologies also pose significant challenges [40, 42]. Additionally, there is need for the use of better models, better accessibility, and improved usability heuristics in the developed systems.

## Chapter 3

### THE PROPOSED ITS DESIGN

The proposed ITS is conceptualized as a web-based platform that aims to revolutionize programming education by leveraging cutting-edge AI and integrating user-friendly and responsive interfaces along with voice tutoring capabilities. This new model builds on the strengths of existing ITS while filling in the gaps identified during the comprehensive review of the current models previously mentioned. Previously designed ITS models for programming education, while useful, have room for improvement in providing a cohesive user experience, reducing their effectiveness as educational tools. Also voice chat capabilities are lacking in current programming ITS. In recognizing this, the proposed ITS was conceptualized with the explicit aim of ensuring seamless interactions between the user and the system. The proposed ITS brings about three main advancements that enhance and extend beyond what current ITS models offer:

1. **Conversation:** One on one tutoring has been shown to be one of the most powerful educational tools [51]. However, traditional ITS often lack an interactive component that enables a two-way, conversational learning experience. Our proposed system addresses this by harnessing the power of LLM's and implementing advanced prompt engineering techniques to instruct the model. Prompt engineering has been shown to greatly improve the usability and reliability of output from LLMs [52]. In this design, system prompts are dynamically fed to the system based on the parameters the user initialized the

chat session with, this helps the ITS to engage the user in a real-time, context-aware dialogue, simulating a personal tutor's role. Furthermore, instructional prompts have been implemented to enable the system to deliver complex programming concepts in a structured and easily digestible manner, providing users with deeper understanding of programming concepts. This approach allows the ITS to provide real-time, interactive guidance to the user, fostering an immersive learning environment that significantly enhances the user experience [53]. The conversational strategy not only enables immediate feedback but also helps expand the user's understanding of programming concepts, making the learning experience much more cohesive.

2. **Advanced Code Review:** Previous systems such as Prutor [54] and ITSCl [55] use CBR or other Retrieval based solutions to evaluate or compare users' code using pre-set answers. However, this approach can be restrictive and may not provide learners with a comprehensive understanding of their code's functionality or the nuances of programming. The proposed system introduces an innovative approach by first attempting to compile and execute the user's code in the application, and then analyzing the output, any warning or error logs and the code written by the user by taking advantage of the code understanding of recent LLM's. This method allows the ITS to provide a detailed walk-through of the code, offering valuable insights, explanations, and even suggestions for improvement. Additionally, this in-depth code review process not only aids learning but also enhances the system's utility as a tool for code debugging and optimization.
3. **Voice Interaction:** Research conducted on the difference between voice and text chatbots has found that voice shows lower cognitive effort, increased perceived efficiency, as well as increased enjoyment as well as reported

satisfaction than text-based interaction [56]. However, text-based interactions dominate most of the current ITS leaving a lot of room for improvement. A notable feature that sets this proposed ITS apart from existing models is the introduction of voice chat. This addition not only makes the system more accessible, but it allows it to accommodate different learning styles and preferences [57]. It also provides users the option to vocalize their thoughts and hear verbal explanations, which can be crucial in solidifying understanding and promoting active learning [58]. Moreover, voice interaction can significantly benefit users with visual impairments or those who prefer audio learning over text.

The ITS operates on a session-based model where users can sign in and create a new learning session based on their preferences. A session could be a chat or voice chat session, with difficulty levels ranging from easy to hard, and catering to three different programming languages: JavaScript, C++, and Python. This design allows the system to accommodate a wide array of learners, from beginners to more advanced coders, and provide them with a tailored learning experience. At the start of a session, the system's LLM-powered chatbot provides the user with a programming problem according to the selected options. The system features a unique, split-screen interface with a chat section on one side and a code sandbox on the other, creating a cohesive learning environment where users can simultaneously engage in a conversation with the chatbot and write/execute code.

For those who learn better through verbal explanations, the system offers a voice chat option powered by WhisperAI and Eleven Labs, where learners can talk through their ideas and responses. This component not only enhances the system's accessibility but

also encourages learners to vocalize their thought processes, a method often used to solidify understanding. In conceptualizing the proposed ITS, the overarching goal was to create a tool that could significantly contribute to the advancement of AI in programming education, enhancing learner engagement, academic performance, and fostering a deeper understanding of programming concepts through a highly interactive and personalized tutoring platform. Leveraging the integration of conversation, advanced code review, and voice interaction into one cohesive systems allow us to provide an optimized learning experience.

### **3.1 Architecture of the Proposed ITS**

The ITS proposed in this study is designed with a three-tier architecture leveraging the power of popular web technologies: a frontend developed using React.js, a backend built with Node.js, and a PostgreSQL database managed through Supabase.

React.js, a JavaScript library maintained by Facebook, forms the foundation of the frontend, providing the interface that the users interact with. React.js was chosen for its ability to create dynamic and high-performing User Interfaces (UIs), allowing for the development of high usability interfaces. It also enables the development of reusable UI components, resulting in a more efficient, readable, and maintainable codebase. Furthermore, React's virtual DOM feature ensures superior performance by updating and rendering components only when it's necessary. These strengths make it a suitable choice for an application expected to manage numerous simultaneous interactions.

The backend, serving as the bridge between the frontend and the database, is constructed using Node.js. Node.js is a JavaScript runtime that runs on the server-side, known for its scalability and efficiency. The system uses Node.js for handling requests from the frontend, executing all the logic, handling the different

APIs, interfacing with the database, and returning responses to the client. Node.js operates on a non-blocking, event-driven I/O model, making it particularly effective for real-time applications that require high throughput like this system.

The data layer is implemented using PostgreSQL, an open-source object-relational database system. PostgreSQL offers robustness, performance, and SQL compliance, thereby providing a reliable and efficient storage solution. It supports both structured and unstructured data, allowing the application to store and query diverse types of data from user accounts to chat logs. Supabase is used to manage PostgreSQL. Supabase streamlines the process of working with PostgreSQL, offering a straightforward API, Row Level Security for the database tables and easy-to-use client libraries. The choice to use Supabase helps ensure security and simplifies the creation and management of user profiles.

In terms of data flow, when a user interacts with the system via the frontend, the application sends a request to the backend. The backend processes this request, interacts with the database if necessary, and sends a response back to the frontend. This three-tier architecture creates a separation of concerns, enhancing the maintainability and scalability of the system, and forms the robust foundation of the proposed system.

### **3.2 Intelligent Tutoring Models**

The effectiveness of any Intelligent Tutoring System (ITS) hinges on its underlying models, which guide its ability to offer a personalized, adaptive, and immersive learning experience. The proposed ITS integrates four key models namely the Domain Model, Student Model, Pedagogical Model, and User Interface Model, and this is how these models help form a cohesive foundation for programming education:

## 1. **Domain Model**

The Domain Model in the proposed ITS covers the essentials of programming languages like JavaScript, C++, and Python. It comprises a hierarchical structure of programming concepts, algorithms, and problem-solving techniques. The model also integrates real-world coding challenges and exercises to help students understand the applicability of various concepts.

## 2. **Student Model**

Rather than capturing real-time data, the Student Model of the proposed system uses the underlying LLM to adapt dynamically during student interactions. It assesses the learner's needs based on their conversation and code submissions, adjusting its instructional strategies in real-time. This approach allows for a personalized, flexible learning experience without requiring complex analytics or data storage.

## 3. **Pedagogical Model**

For the Pedagogical model, the system determines the most suitable teaching strategies and learning activities based on the student's needs, as assessed by the Student Model. For example, if a student is struggling with loops in Python, the system might introduce easier exercises, multiple examples, or scaffolded solutions to help them grasp the concept better.

## 4. **User Interface Model**

The ITS also features a dual-pane interface with a conversational agent on one side and a code sandbox on the other. This User Interface Model is designed not just for ease of use but also to enhance the learning experience, allowing for seamless integration of text and voice interactions, code writing, and immediate feedback, thus making it easier for learners to absorb complex programming concepts.

### 3.3 ITS Flow and User experience

The effectiveness of any system, particularly an educational tool, is largely determined by the user experience it offers. A well-crafted user flow can significantly enhance engagement, retention, and overall learning. With this understanding, the proposed ITS has been designed to offer an intuitive, efficient, and user-friendly experience. The proposed ITS offers two straightforward authentication methods:

1. **Magic Link:** This is an email-based authentication process enabled by Supabase. A user provides their email address and receives a 'magic link' in their inbox. By simply clicking this link, users are authenticated and logged into the system. This method eliminates the need for password memorization, providing both convenience and a layer of security.
2. **One-click Google Sign-in:** For those familiar with Google's ecosystem, this option allows for an almost instantaneous access. Users can use their Google credentials to sign in with just one click, bypassing the traditional username-password method.

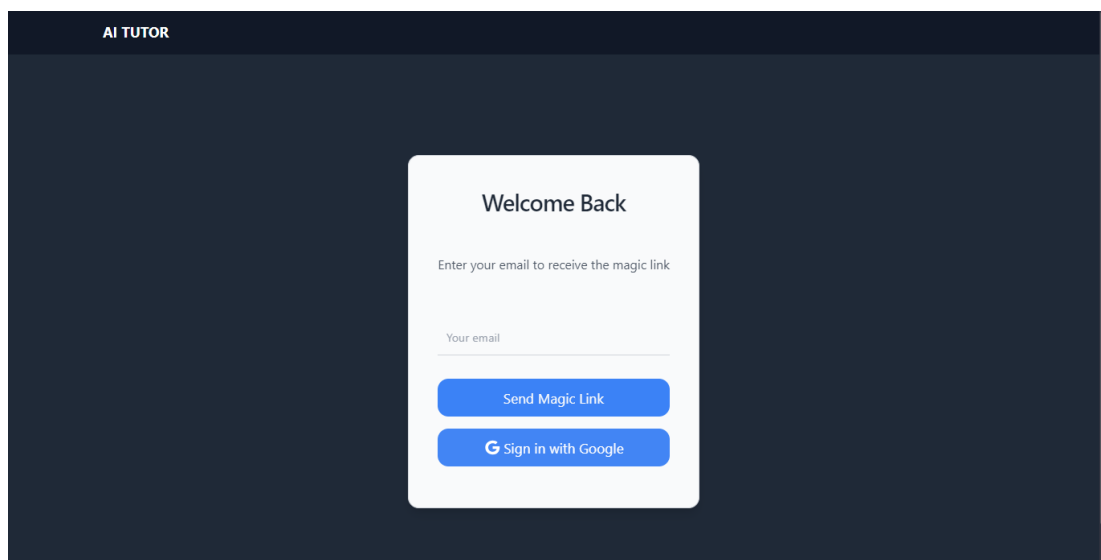


Figure 3.1: Chat interface of the proposed ITS

Once inside, users are met with a 'New Session' screen. Here, users can set up three personalization settings for their tutor session:

1. **Mode of Interaction:** Users can choose between a chat-based session or a voice chat session. The chat option offers a more textual, focused approach, while voice chat is also available.
2. **Difficulty Level:** There are three difficulty levels available: easy, medium, and hard. This ensures that both beginners and advanced students find challenges suitable for their level.
3. **Programming Language:** Users can select among three of the most popular and widely-used languages: JavaScript, C++, and Python.

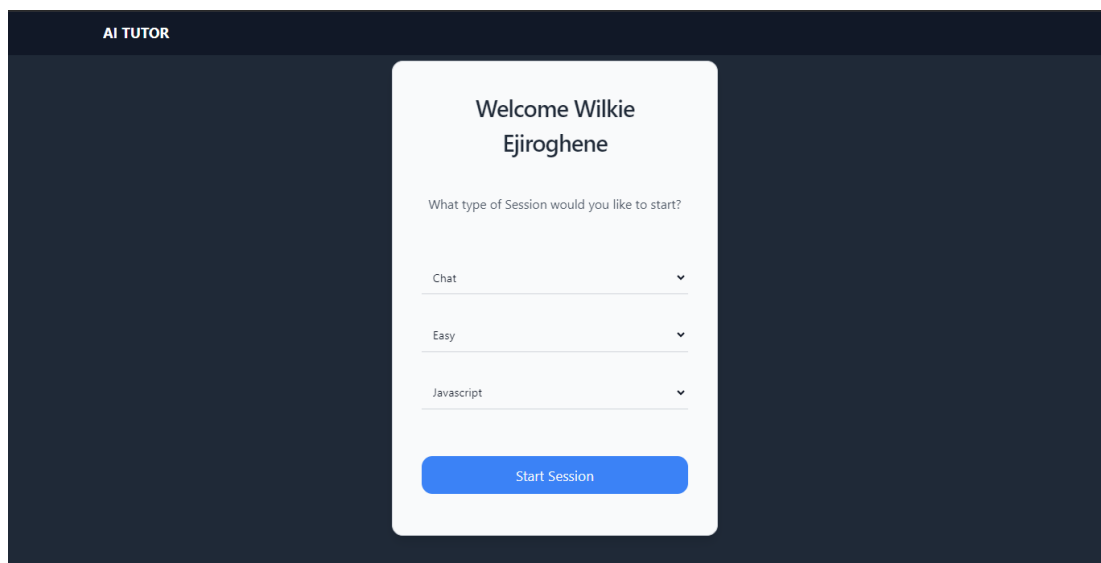


Figure 3.2: New session interface of the proposed ITS

Once the parameters are set, the user is greeted by the AI-powered tutor in an interactive chat session. This chat environment is divided into two main sections: the chat interface on the left and the code sandbox on the right. The AI tutor commences the session by providing a programming challenge that aligns with the user's previously selected difficulty and language preferences. This problem serves as the

central learning element of the session.

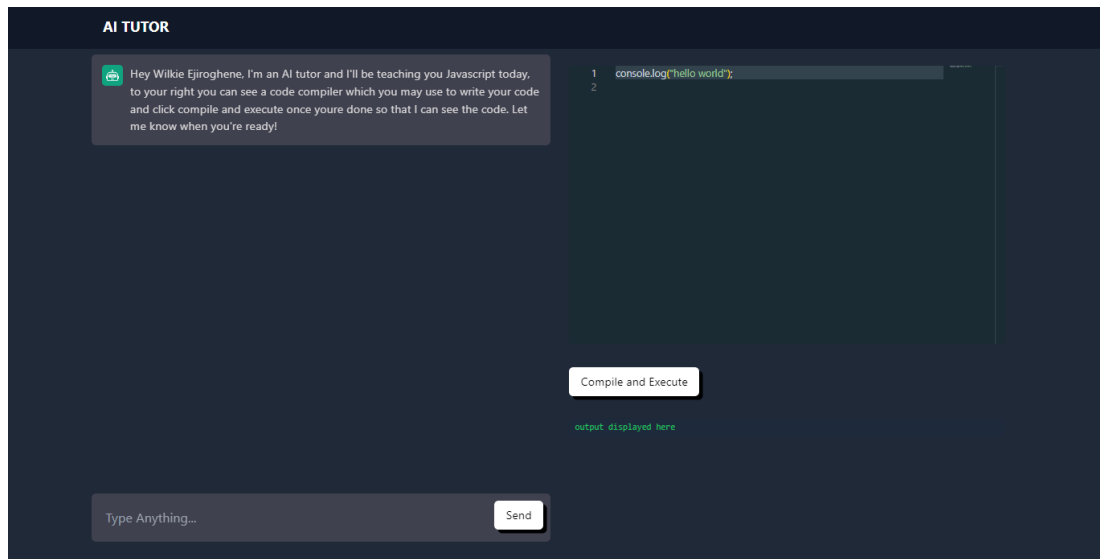


Figure 3.3: Welcome screen interface of the proposed ITS

Users then dive into problem-solving within the code sandbox. This integrated editor offers the capability to write, execute, and test code in real-time, creating a fluid conversation between the learner and the AI tutor.

As users work through the problem, the AI tutor stands ready to assist. If a user's solution contains errors or inefficiencies, the tutor provides targeted feedback, suggestions, and in-depth explanations, guiding the learner towards the correct solution. This interactive feedback mechanism ensures learners are not just handed solutions but are encouraged to understand their mistakes and improve their skills.

### 3.4 AI Features and Capabilities

The proposed ITS heavily relies on AI capabilities to deliver an interactive, personalized, and highly productive learning experience. The key component of the system is its chatbot, which is powered by an LLM. After carefully exploring different options such as Anthropic's Claude, Google's Bard, and Meta's Llama, along with other open source models built on Llama, OpenAI's GPT-3.5-turbo was

**AI TUTOR**

This code was submitted through the compiler by the student: 

```
function multiplyArray = (array) => {
  return
}
```

 and here is the output from the compiler after executing their code: Déjà

Good start, but it seems like you haven't implemented the actual logic to multiply the numbers together. Remember, you need to iterate through the array and update the product by multiplying each element with the current product value.

Let me give you a hint to help you complete your code:

Inside your function, declare a variable called 'product' and initialize it to 1. This variable will store the accumulated product.

Then, use a loop to iterate through each element of the array. Multiply the current element with the 'product' variable and update the 'product' variable with the result.

Finally, return the 'product' variable.

Try updating your code with this approach and let me know when you're ready to test it.

Type Anything...

```

1 function multiplyArray = (array) => {
2   return
3 }

/box/script.js:1
function multiplyArray = (array) => {
  return
}

SyntaxError: Unexpected token '-'
    at Module.compile (internal/modules/cjs/loader.js:895:18)
    at Object.Module._extensions..js (internal/modules/cjs/loader.js:995:18)
    at Module.load (internal/modules/cjs/loader.js:818:32)
    at Function.Module._load (internal/modules/cjs/loader.js:727:14)
    at Function.Module.runMain (internal/modules/cjs/loader.js:1047:10)
    at internal/main/run_main_module.js:17:11

```

Figure 3.4: Wrong solution attempt

**AI TUTOR**

```
result *= array[i];
}
```

 and here is the output from the compiler after executing their code: Déjà

Great progress! However, you're missing the return statement at the end of your function. Without the return statement, the function will not output the result correctly.

Update your code to include the return statement, like this:

```

'''javascript
function multiplyArray(array) {
  let result = 1;

  for (let i = 0; i < array.length; i++) {
    result *= array[i];
  }

  return result;
}
'''

```

Now, try running your code again and test it with some sample inputs. Let me know if you encounter any issues.

Type Anything...

```

1 function multiplyArray (array) {
2   let result = 1;
3
4   for (let i = 0; i < array.length; i++) {
5     result *= array[i];
6   }
}

```

Déjà

Figure 3.5: Wrong solution attempt 2

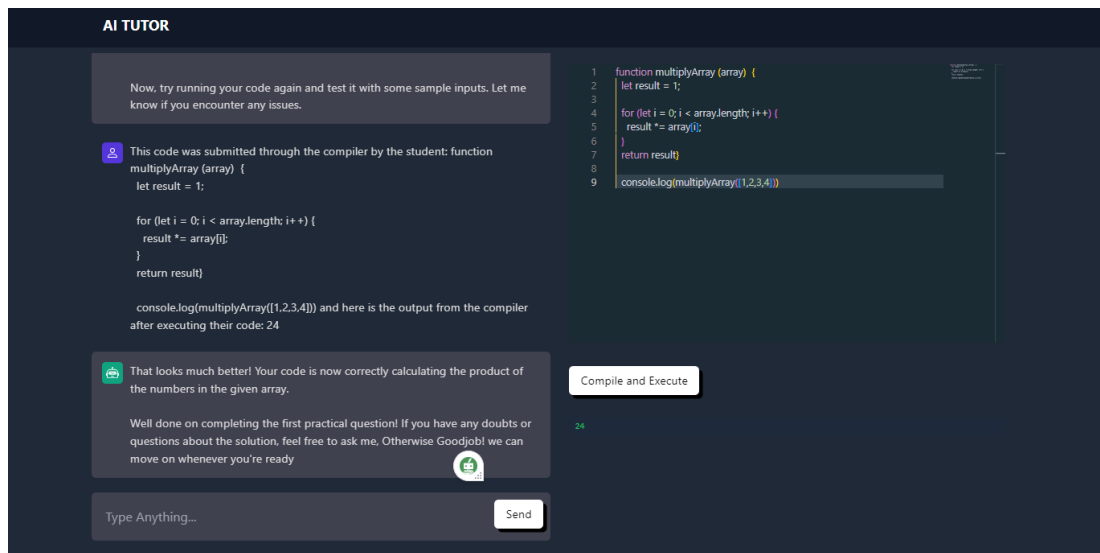


Figure 3.6: Tutor explanation

settled on as its LLM. Several factors contributed to this decision. First, GPT-3.5 was found to be more accessible and predictable at handling coding tasks compared to the other evaluated models [59]. It also demonstrated a robust understanding of programming languages and a consistent ability to generate relevant and accurate responses. Since the LLM acts as the backbone of the tutoring system, guiding the conversation in the chat sessions, GPT-3.5's superior language understanding and generation capabilities enable it to simulate natural and engaging conversations. This helps facilitate an interactive learning environment by providing real-time feedback to users' queries or coding attempts, culminating in a responsive and dynamic learning experience. Moreover, the LLM can generate in-depth explanations of complex programming concepts, promoting better understanding and skill acquisition. Additionally, GPT-3.5 facilitates the system's adaptability, adjusting its interaction based on the user's progress in the conversation. This personalization of the learning process, enabled by AI, significantly enhances learning efficacy. While GPT-4 was considered for its advanced logical capabilities and reasoning skills, which could potentially improve the coding ability of the chatbot, it was ultimately not chosen due

to GPT-3.5's speed, affordability, and proven consistency for this use case.

A key improvement of this ITS over previous systems is its integration of AI into its voice chat sessions. This is implemented using WhisperAI and Eleven Labs' technologies. WhisperAI, an advanced speech-to-text engine, is used to interpret the user's verbal input. The input is then processed by the same GPT-3.5 engine previously discussed to generate a text response. This response is then converted into speech using Eleven Labs' text-to-speech technology, facilitating an alternative mode of learning and encouraging learners to articulate their thoughts verbally.

The system's design also accommodates the possibility of swapping out the current LLM for newer and more advanced models as they become available. This design choice future-proofs the tutoring system, allowing it to evolve and adapt with the fast-paced progression of AI technology, and making it possible to leverage improvements in language models for enhanced tutoring capabilities. By maintaining this level of flexibility and adaptability, the system ensures its ability to stay at the forefront of AI-assisted programming education.

### **3.5 Improvements over previous systems**

The major new feature that this proposed ITS provides is the implementation of voice tutoring. Current programming ITS lack this feature, and having this alternative mode of learning provides opportunity for increased effectiveness of the system as voice tutoring has been shown to provide better results than text [56]. The system further enhances the interactive learning experience through user-centered design and a streamlined learning experience that seamlessly integrates AI technology. It does this in three main ways: promoting engaging interaction, providing code evaluation, and prioritizing accessibility.

The interactivity of the system primarily arises from its chat and voice chat functionalities. The chat interface, powered by the AI language model, not only provides users with a problem to solve based on their chosen options but also enables real-time feedback. Users can interact with the AI tutor, asking questions or seeking clarification, and receive instantaneous, in-depth explanations. This one-on-one engagement encourages active learning, as users have to articulate their thoughts, ask questions, and solve problems actively. The voice chat feature also provides an alternative mode of learning and allows users to articulate their thoughts verbally. This is particularly beneficial for auditory learners and encourages the practice of articulating ideas aloud, a key skill in collaborative coding environments. The combination of chat and voice chat interaction offers a multimodal learning environment, catering to different learning styles.

The system further enhances the learning experience by incorporating a robust code evaluation mechanism. The right section of the chat session window features a 'code sandbox' - an integrated development environment that allows users to write and execute their code directly within the session. This live coding environment is crucial for hands-on learning and immediate application of theoretical knowledge. Moreover, the system leverages Judge0 API for code compilation and execution, which supports multiple programming languages and provides standardized, reliable outputs. The code, its output, any error logs, and console logs are automatically fed into the chat. This instant feedback loop enables users to quickly identify and correct their errors, leading to efficient learning and problem-solving skills.

ITS also makes accessibility a priority. To begin with, the user-friendly interface, easy navigation, and the high degree of customization for preferred learning options - chat

or voice chat, difficulty level, and programming language - make the system welcoming for users at all proficiency levels while providing inclusivity and adaptability. Moreover, users can create an account using their email or by signing in with Google, further simplifying the login process. The choice of web-based application broadens accessibility as it can be used on any device with internet access, without the need for complex installation processes. Furthermore, the system takes care to ensure accessibility for individuals with disabilities. For instance, browser text to speech in collaboration with the voice chat feature can aid learners with visual impairments.

These features put together offer a transformative interactive learning experience. The user-centered design and integration of AI technology in the ITS coupled with its unique chat, voice chat, code sandbox and instant feedback promote engaging and personalized learning, effectively catering to different learning styles and facilitating active knowledge application. Additionally, the platform prioritizes accessibility, evident in its user-friendly interface, customization options, simplified login process, and device-agnostic nature. It also demonstrates a commitment to inclusivity, providing support for learners with disabilities. This system, therefore, represents a significant leap forward in creating an effective, inclusive, and accessible environment for learning programming.

## **Chapter 4**

### **ITS COMPARISON RESULTS**

#### **4.1 Methodology for Comparisons**

The primary objective of this research is not just to propose an innovative ITS, but to contextualize its benefits by comparing it with existing models. Given the variety of ITS models available in the domain of programming education, a purposive sampling method was adopted. Models were selected based on their popularity, their reported efficacy in academic literature, and their relevance to the specific objectives our proposed system seeks to address. To achieve a comprehensive comparative analysis, the academic literature was consulted including peer-reviewed articles, conference papers, and case studies that discussed the features, advantages, and challenges of the selected ITS models.

The comparison is structured around three primary parameters: Interactivity, Accessibility, and Reliability. Each of these parameters was broken down into specific, measurable indicators. While every effort was made to ensure an objective and thorough comparison, there are inherent limitations. These include potential biases in assessments, the constant evolution of technology rendering some comparisons time-sensitive, and the challenges in quantifying certain qualitative aspects of user experience.

#### **4.2 Comparison Parameters**

These three comparison parameters will not only allow for a structured comparison but also capture the essential aspects of an ITS for effective programming education:

1. **Interactivity:** Interactivity is pivotal for learning programming. Real-time feedback, dynamic problem-solving sessions, and the ability to query and clarify doubts instantly can greatly enhance the learner's understanding and retention. However, it goes beyond mere user-system interaction, it defines the depth and breadth of the interaction, shaping the learning experience. This definition helps to measure the extent to which the system can engage with users in a dynamic manner, adapting to their inputs and needs in real-time. Features like how long it takes for a user to set up a session, real-time feedback provision, response latency and conversational flow quality will be the indicators of interactivity.
2. **Accessibility:** Accessibility here means the system's availability across platforms, devices, and its inclusivity features catering to learners with diverse needs, including disabilities. An accessible system ensures no learner is left behind due to technological or physical constraints, and this is an important factor when evaluating a tool intended for education. To measure this, Number of platforms and devices supported, and features catering to learners with disabilities will be evaluated.
3. **Reliability:** This will measure the degree to which learners can depend on the system for consistent, quality guidance. The systems will be evaluated based on how consistently they perform as well as the accuracy of feedback. Here, estimates will be made based on assessment of their backend architecture to determine their uptimes, the frequency of technical glitches/error rates, and the precision and relevance of the feedback provided. This is an important metric because inconsistent performance or inaccurate feedback can erode trust and hinder the learning process.

### 4.3 Comparative Analysis with Existing ITS Models

With the parameters established, the next step is to compare the proposed ITS with existing models. Here are the previous ITS selected from the literature and their features:

1. **Dodona:** A popular programming ITS developed at Ghent University in Belgium [17]. The platform is fully open sourced is built to give automatic corrections and meaningful feedback to students.

(a) **Interactivity:** The student onboarding process is lengthy, students have to register to the system and then find the desired course. And while there is a large selection of courses, this presentation can be overwhelming. This contrasts with the ease with which students can start learning on our system, where they are only a few clicks away from starting a new session. Additionally, students can only solve preset questions and view their solutions, and there is no chat feature for the student to explore further.

(b) **Accessibility:** Dodona is a web app and is available on all platforms. Their documentation also explains how to use browser text to speech features in other to aid disabled students. There are also issues users have complained about on github with some students not being able to find or access courses due to teachers not setting them up correctly due to its complexity.

(c) **Reliability:** The system is fairly reliable and is on par with the proposed system.

2. **Korbit:** Korbit is an ITS built in a collaboration between researchers in Canada and the United Kingdom [60]. The system utilizes the GitHub pull request system to analyze students code.

(a) **Interactivity:** Users can make a pull request and then receive feedback from an AI powered tutor. The tutor reviews the changes in the pull request

and the users can also ask questions and chat with the tutor.

(b) **Accessibility:** The tutor is accessible to users with a GitHub account, however this poses an accessibility issue because not everyone knows how to use git and GitHub. There is also no implementation for less abled students.

(c) **Reliability:** The system is reliable and performed consistently throughout testing.

3. **Prutor:** A popular ITS developed at IIT Kanpur [54]. Prutor has been used for teaching in multiple institutes in India and is a web based ITS to help students learn programming.

(a) **Interactivity:** The system allows students to write their code right in the teaching environment and intermittently takes snapshots of the student's attempt to solve the questions. This data can then be visualized and used by the teachers. This system also lacks an interactive chat feature.

(b) **Accessibility:** Prutor is also a web app available on all web-enabled devices with an account. However, there are no specific accessibility features implemented in the platform.

(c) **Reliability:** This system also demonstrates similar reliability to the proposed ITS.

#### 4.4 Discussion of Findings

The proposed ITS was compared with three existing ITS models, namely Dodona, Korbit, and Prutor. The comparison was based on the three parameters previously discussed, Interactivity, Accessibility, and Reliability:

1. **Interactivity:** Teaching programming concepts efficiently is highly dependent on problem-solving and immediate feedback. The analysis carried out in this research indicated a noticeable variability in how interactive the selected systems

were. While platforms like Dodona offer a plethora of courses, they lack the dynamism and immediate feedback loop our system emphasizes. Moreover, our system's reduced setup time and streamlined conversational flow prioritize the learner's experience, ensuring that they can dive right in and remain engaged and also invested throughout the learning process. Systems like Prutor, which do capture intermittent snapshots of the student's coding process, still fall short of providing an interactive environment tailored to the individual learner's pace and style.

2. **Accessibility:** Today's learners use a variety of devices and platforms, and their individual needs vary widely. This means that accessibility in educational platforms is of the utmost importance. The proposed ITS voice tutoring feature not only breaks barriers of textual communication but also bridges the gap for learners with disabilities. Platforms like Dodona and Prutor, while also web-based, do not seem to extend their accessibility features beyond the conventional, leaving a significant segment of the learner population potentially underserved. The challenges with Korbit's dependency on GitHub showcase that while integrating with popular platforms can be beneficial, it can also inadvertently exclude learners unfamiliar with those platforms.
3. **Reliability:** Consistency is a cornerstone of trust in any learning environment, as any disruptions or interruptions in the learning process can be difficult to recover from. The analysis revealed that most systems, including Dodona, Korbit, and Prutor, are fairly reliable. With acceptable feedback precision, uptime, and overall robustness of their backend architectures contributing significantly to the reliability of their system. The proposed ITS, is on par with these systems in these aspects, aims to ensure that the feedback is not only consistent but also contextually relevant, fostering trust and more effective

learning.

The results show that the proposed ITS stands out in its dynamic interactivity, reducing setup time, and enhancing the conversational flow to cater to individual learner needs. Accessibility especially is one of the significant strengths of the ITS. The voice tutoring feature is unmatched in previous programming ITS ensuring inclusivity across devices and platforms, and especially for learners with disabilities. It also exhibits at least equivalent reliability with other systems, which coupled with its other advancements, position it as an advancement in terms of effectiveness.

All of these points further demonstrate the need for specialized ITS for programming education where the requirements differ vastly from other domains. While existing ITS have their strengths, there's an obvious gap that our proposed system addresses with its improved interactivity and accessibility, paving the way for a more immersive and effective programming learning experience.

## Chapter 5

### DISCUSSION AND CONCLUSION

This research centered on exploring AI's recent developments and how the advancements can be applied in the sphere of programming education. Programming is a skill of logic and application and requires nuanced guidance, and this AI powered ITS provides an environment for learners to experiment, fail, and learn at their own pace. First, the foundational importance of ITS in the educational spectrum along with their unique role in programming education were discussed, followed by a comprehensive review of existing ITS models. This revealed diverse approaches to programming education. Many systems were advanced in certain categories, however they lacked usable voice tutoring capabilities, and still exhibited gaps in terms of real-time interactivity, device and platform accessibility, and consistent reliability.

After gaining a foundational understanding of the gaps in current systems, this research then proceeded to conceptualize a unique ITS system. The hallmark features being voice tutoring, in-session code editing and compilation, and enhanced interactivity, accessibility and reliability features. The design propositions focused on providing a cohesive user experience while making sure that the tool was inclusive and dependable. The proposed system was then compared with existing programming ITS, as well as some non-programming chatbot tutors. The comparative assessment suggested the following:

- Interactivity in the proposed system makes the user's learning experience more engaging than previous systems by cutting down on setup times and ensuring

dynamic, real-time feedback.

- The design choices taken to ensure accessibility provide an improvement over existing systems and helps ensure that no one is left behind.
- Improved reliability provided by the system help it provide a consistent, glitch-free experience, further establishing its potential as a widely adopted education tool.

While the research provides significant insights and propositions, it has limitations that must be acknowledged. The proposed ITS system, while theoretically robust, has not been empirically tested against real-world educational environments. Additionally, the evolving nature of technology means that the landscape of ITS and AI-powered education tools is continuously shifting, which might present newer challenges and opportunities. With that being said, these results highlighted our efforts to bridge the identified gaps in existing ITS for programming. Adding voice tutoring in programming education offers a paradigm shift in how learners can interact, explore, and clarify their questions offering a multi-modal learning experience. This integration can significantly shape pedagogical strategies by providing a learning companion alongside other traditional methods. Overall, this research has shown the profound role of AI in personalizing education, setting a benchmark for future ed-tech tools. Further research can also be done on empirical testing of the proposed ITS to measure its efficacy in real-world educational settings as well as exploring solutions to further improve the response time of the AI tutor.

## REFERENCES

- [1] Kam Cheong Li and Billy Tak-Ming Wong. Features and trends of personalised learning: A review of journal publications from 2001 to 2018. *Interactive Learning Environments*, 29(2):182–195, February 2021.
  
- [2] Orly Lipka, Alona Forkosh Baruch, and Yael Meer. Academic support model for post-secondary school students with learning disabilities: Student and instructor perceptions. *International Journal of Inclusive Education*, 23(2):142–157, February 2019.
  
- [3] Aditi Bhutoria. Personalized education and artificial intelligence in United States, China, and India: A systematic Review using a Human-In-The-Loop model. *Computers & Education: Artificial Intelligence*, pages 100068–100068, April 2022.
  
- [4] Vivian Lai, Chacha Chen, Q. Liao, Alison Smith-Renner, and Chenhao Tan. Towards a Science of Human-AI Decision Making: A Survey of Empirical Studies. *ArXiv*, December 2021.
  
- [5] Bong Gyu Lee, Gyeong Uk Jeong, and Won Kyeong Lee. Participatory Action Research on Class for Student Below the Basic Level in Middle School Using AI-based Adaptive Learning System Applying Personalized Instruction Strategy. *Korean Association For Learner-Centered Curriculum And Instruction*, 2022.

- [6] Kersten T. Schroeder, Martha Hubertz, Rachel Van Campenhout, and Benny G. Johnson. Teaching and Learning with AI-Generated Courseware: Lessons from the Classroom. *Online Learning*, 26(3), September 2022.
- [7] Víctor González-Calatayud, Paz Prendes-Espinosa, and Rosabel Roig-Vila. Artificial Intelligence for Student Assessment: A Systematic Review. *Applied Sciences*, 11(12):5467, June 2021.
- [8] Mushtaq Hussain, Wenhao Zhu, Wu Zhang, and Syed Muhammad Raza Abidi. Student Engagement Predictions in an e-Learning System and Their Impact on Student Course Assessment Scores. *Computational Intelligence and Neuroscience*, 2018:1–21, October 2018.
- [9] A. M. Cox. Exploring the impact of Artificial Intelligence and robots on higher education through literature-based design fictions. *International Journal of Educational Technology in Higher Education*, 18(1):3, January 2021.
- [10] Ahtsham Manzoor and Dietmar Jannach. Generation-based vs. Retrieval-based Conversational Recommendation: A User-Centric Comparison. *ACM Conference on Recommender Systems*, pages 515–520, September 2021.
- [11] Oussama Hamal, Nour-eddine El Faddouli, Moulay Hachem Alaoui Harouni, and Joan Lu. Artificial Intelligent in Education. *Sustainability*, 14(5):2862–2862, March 2022.
- [12] Kan Jia, Penghui Wang, Yang Li, Zezhou Chen, Xinyue Jiang, Chien Linag Lin,

and Tachia Chin. Research Landscape of Artificial Intelligence and e-Learning: A Bibliometric Research. *Frontiers in Psychology*, 13, February 2022.

- [13] Fatema AlShaikh and Nabil M. Hewahi. AI and Machine Learning Techniques in the Development of Intelligent Tutoring System: A Review. *2021 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, September 2021.
- [14] Chinedu Wilfred Okonkwo and Abejide Ade-Ibijola. Chatbots applications in education: A systematic review. *Computers and Education: Artificial Intelligence*, 2:100033, 2021.
- [15] Chen Cao. Leveraging Large Language Model and Story-Based Gamification in Intelligent Tutoring System to Scaffold Introductory Programming Courses: A Design-Based Research Study. 2023.
- [16] Santiago Schez-Sobrino, Cristian Gmez-Portes, David Vallejo, Carlos Glez-Morcillo, and Miguel Á. Redondo. An Intelligent Tutoring System to Facilitate the Learning of Programming through the Usage of Dynamic Graphic Visualizations. *Applied Sciences*, 10(4):1518, February 2020.
- [17] Charlotte Van Petegem, Peter Dawyndt, and Bart Mesuere. Dodona: Learn to Code with a Virtual Co-teacher that Supports Active Learning. *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 2*, pages 633–633, June 2023.

- [18] Mohammad Amin Kuhail, Nazik Alturki, Salwa Alramlawi, and Kholood Alhejori. Interacting with educational chatbots: A systematic review. *Education and Information Technologies*, 28(1):973–1018, January 2023.
- [19] William Cai, Josh Grossman, Zhiyuan Jerry Lin, Hao Sheng, Johnny Tian-Zheng Wei, Joseph Jay Williams, and Sharad Goel. Bandit algorithms to personalize educational chatbots. *Machine Learning*, 110(9):2389–2418, September 2021.
- [20] Ninni Singh, Vinit Kumar Gunjan, and Moustafa M. Nasralla. A Parametrized Comparative Analysis of Performance Between Proposed Adaptive and Personalized Tutoring System “Seis Tutor” With Existing Online Tutoring System. *IEEE Access*, 10:39376–39386, 2022.
- [21] Roger Azevedo, François Bouchet, Melissa Duffy, Jason Harley, Michelle Taub, Gregory Trevors, Elizabeth Cloude, Daryn Dever, Megan Wiedbusch, Franz Wortha, and Rebeca Cerezo. Lessons Learned and Future Directions of MetaTutor: Leveraging Multichannel Data to Scaffold Self-Regulated Learning With an Intelligent Tutoring System. *Frontiers in Psychology*, 13:813632, June 2022.
- [22] Gustavo Carreira, Leonardo Silva, Antonio Jose Mendes, and Hugo Goncalo Oliveira. Pyo, a Chatbot Assistant for Introductory Programming Students. *2022 International Symposium on Computers in Education (SIIE)*, pages 1–6, November 2022.
- [23] Priynka Sharma and Mayuri Harkishan. Designing an intelligent tutoring system

for computer programming in the Pacific. *Education and Information Technologies*, 27(5):6197–6209, June 2022.

- [24] N. T. A. Ramaha, H. Abdellatef, and I. R. Karas. AN INTELLIGENT TUTORING SYSTEM TO MAINTAIN THE STUDENTS' MOTIVATION. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVIII-4/W3-2022:143–147, December 2022.
- [25] Tianjia Wang, Daniel Vargas-Diaz, Chris Brown, and Yan Chen. Towards Adapting Computer Science Courses to AI Assistants' Capabilities. 2023.
- [26] Arzu Deveci Topal, Canan Dilek Eren, and Aynur Kolburan Geçer. Chatbot application in a 5th grade science course. *Education and Information Technologies*, 26(5):6241–6265, September 2021.
- [27] Aohua Ni and Alan Cheung. Understanding secondary students' continuance intention to adopt AI-powered intelligent tutoring system for English learning. *Education and Information Technologies*, 28(3):3191–3216, March 2023.
- [28] Kai-Chih Pai, Bor-Chen Kuo, Chen-Huei Liao, and Yin-Mei Liu. An application of Chinese dialogue-based intelligent tutoring system in remedial instruction for mathematics learning. *Educational Psychology*, 41(2):137–152, February 2021.
- [29] Harry Barton Essel, Dimitrios Vlachopoulos, Akosua Tachie-Menson, Esi Eduafua Johnson, and Papa Kwame Baah. The impact of a virtual teaching assistant (chatbot) on students' learning in Ghanaian higher education.

*International Journal of Educational Technology in Higher Education*, 19(1):57,  
November 2022.

- [30] Yunan Chen, Yu Chen, Scott Jensen, Leslie J. Albert, Sambhav Gupta, and Terri Lee. Artificial Intelligence (AI) Student Assistants in the Classroom: Designing Chatbots to Support Student Success. *Information Systems Frontiers*, June 2022.
- [31] Tao Fang, Shu Yang, Kaixin Lan, Derek F. Wong, Jinpeng Hu, Lidia S. Chao, and Yue Zhang. Is ChatGPT a Highly Fluent Grammatical Error Correction System? A Comprehensive Evaluation. *arXiv.org*, 2023.
- [32] Ludovic Le Bigot, Jean-François Rouet, and Eric Jamet. Effects of Speech- and Text-Based Interaction Modes in Natural Language Human-Computer Dialogue. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 49(6):1045–1053, December 2007.
- [33] Franziska Gaiser and Sonja Utz. Is Hearing Really Believing?: The Importance of Modality for Perceived Message Credibility During Information Search With Smart Speakers. *Journal of Media Psychology*, pages 1864–1105/a000384, June 2023.
- [34] Stephan Diederich, University of Göttingen, Germany, Alfred Benedikt Brendel, TU Dresden, Germany, Stefan Morana, Saarland University, Germany, Lutz Kolbe, and University of Göttingen, Germany. On the Design of and Interaction with Conversational Agents: An Organizing and Assessing Review of Human-

Computer Interaction Research. *Journal of the Association for Information Systems*, 23(1):96–138, 2022.

- [35] Sofia Schöbel, Anuschka Schmitt, Dennis Benner, Mohammed Saqr, Andreas Janson, and Jan Marco Leimeister. Charting the Evolution and Future of Conversational Agents: A Research Agenda Along Five Waves and New Frontiers. *Information Systems Frontiers*, April 2023.
- [36] Mateusz Dubiel, Martin Halvey, L. Azzopardi, and Sylvain Daronnat. Investigating how conversational search agents affect user’s behaviour, performance and search experience. July 2018.
- [37] Christine Rzepka. Examining the Use of Voice Assistants: A Value-Focused Thinking Approach. In *Americas Conference on Information Systems*, 2019.
- [38] Nuno M. Guerreiro, Duarte Alves, Jonas Waldendorf, B. Haddow, Alexandra Birch, Pierre Colombo, and André Martins. Hallucinations in Large Multilingual Translation Models. *arXiv.org*, 2023.
- [39] Selin Akgun and Christine Greenhow. Artificial intelligence in education: Addressing ethical challenges in K-12 settings. *AI and Ethics*, 2(3):431–440, August 2022.
- [40] C. Chan and Wenjie Hu. Students’ Voices on Generative AI: Perceptions, Benefits, and Challenges in Higher Education. *arXiv.org*, 2023.

- [41] Wenxiang Jiao, Wenxuan Wang, Jen-tse Huang, Xing Wang, and Zhaopeng Tu. Is ChatGPT A Good Translator? A Preliminary Study. *arXiv.org*, 2023.
- [42] Chokri Kooli. Chatbots in Education and Research: A Critical Examination of Ethical Implications and Solutions. *Sustainability*, 15(7):5614, March 2023.
- [43] Andrea Phillips, John F. Pane, Rebecca Reumann-Moore, and Oluwatosin Shenbanjo. Implementing an adaptive intelligent tutoring system as an instructional supplement. *Educational Technology Research and Development*, 68(3):1409–1437, June 2020.
- [44] Rodrigo Francisco and Flávio Silva. Intelligent Tutoring System for Computer Science Education and the Use of Artificial Intelligence: A Literature Review:. In *Proceedings of the 14th International Conference on Computer Supported Education*, pages 338–345, Online Streaming, — Select a Country —, 2022. SCITEPRESS - Science and Technology Publications.
- [45] Cansu Koyuturk, Mona Yavari, Emily Theophilou, Sathya Bursic, Gregor Donabauer, Alessia Telari, Alessia Testa, Raffaele Boiano, Alessandro Gabbiadini, Davinia Hernandez-Leo, Martin Ruskov, and Dimitri Ognibene. Developing Effective Educational Chatbots with ChatGPT prompts: Insights from Preliminary Tests in a Case Study on Social Media Literacy. 2023.
- [46] Konstantina Chrysafiadi, Maria Virvou, George A. Tsihrintzis, and Ioannis Hatzilygeroudis. Evaluating the user’s experience, adaptivity and learning outcomes of a fuzzy-based intelligent tutoring system for computer programming

for academic students in Greece. *Education and Information Technologies*, 28(6):6453–6483, June 2023.

- [47] Sumit Pandey and Srishti Sharma. A comparative study of retrieval-based and generative-based chatbots using Deep Learning and Machine Learning. *Healthcare Analytics*, page 100198, May 2023.
- [48] Angelo Arguson, Elisa Malasaga, May Florence San Pablo, Rhonnel Paculanan, Shaneth Ambat, and Beau Gray Habal. Data analysis and Constraint-based modeling of novice C programming error logs: An input for developing intelligent tutoring system. *2022 IEEE 14th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, pages 1–6, December 2022.
- [49] Yun Huang, Peter Brusilovsky, Julio Guerra, Kenneth Koedinger, and Christian Schunn. Supporting skill integration in an intelligent tutoring system for code tracing. *Journal of Computer Assisted Learning*, 39(2):477–500, April 2023.
- [50] Rui Zhao, Harvey Siy, Chulwoo Pack, Leen-Kiat Soh, and Myoungkyu Song. An Intelligent Tutoring System for API Misuse Correction by Instant Quality Feedback. In *2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 123–128, June 2022.
- [51] Andre Nickow, Philip Oreopoulos, and Vincent Quan. The Impressive Effects of Tutoring on Prek-12 Learning: A Systematic Review and Meta-Analysis of the Experimental Evidence, July 2020.

- [52] Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C. Schmidt. A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT. 2023.
- [53] Paul Denny, Viraj Kumar, and Nasser Giacaman. Conversing with Copilot: Exploring Prompt Engineering for Solving CS1 Problems Using Natural Language. *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, pages 1136–1142, March 2023.
- [54] Rajdeep Das, Umair Z. Ahmed, Amey Karkare, and Sumit Gulwani. Prutor: A System for Tutoring CS1 and Collecting Student Programs for Analysis. *ArXiv*, August 2016.
- [55] Ijaz Ul Haq, Aamir Anwar, Iqra Basharat, and Kashif Sultan. Intelligent Tutoring Supported Collaborative Learning (ITSCL): A Hybrid Framework. *International Journal of Advanced Computer Science and Applications*, 11(8), 2020.
- [56] Christine Rzepka, Benedikt Berger, and Thomas Hess. Voice Assistant vs. Chatbot – Examining the Fit Between Conversational Agents’ Interaction Modalities and Information Search Tasks. *Information Systems Frontiers*, 24(3):839–856, June 2022.
- [57] Korinn S. Ostrow and N. Heffernan. Testing the Multimedia Principle in the Real World: A Comparison of Video vs. Text Feedback in Authentic Middle School Math Assignments. In *Educational Data Mining*, 2014.

- [58] Emily Kuang, Ehsan Jahangirzadeh Soure, Mingming Fan, Jian Zhao, and Kristen Shinohara. Collaboration with Conversational AI Assistants for UX Evaluation: Questions and How to Ask them (Voice vs. Text). *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–15, April 2023.
- [59] Ali Borji and Mehrdad Mohammadian. Battle of the Wordsmiths: Comparing ChatGPT, GPT-4, Claude, and Bard, June 2023.
- [60] Iulian Vlad Serban, Varun Gupta, Ekaterina Kochmar, Dung D. Vu, Robert Belfer, Joelle Pineau, Aaron Courville, Laurent Charlin, and Yoshua Bengio. A Large-Scale, Open-Domain, Mixed-Interface Dialogue-Based ITS for STEM. *Artificial Intelligence in Education*, 12164:387–392, 2020.