PROGRAMMING LEARNING APP FOR KIDS

R24-109

Final Report

Sanjeevan M.C.M.A

B.Sc. (Hons) in Information Technology Specializing in Software Engineering

Department of Computer Science and Software Engineering

Faculty of Computing

Sri Lanka Institute of Information Technology

August 2024

PROGRAMMING LEARNING APP FOR KIDS

R24-109

Final Report

Sanjeevan M.C.M.A

IT21023446

B.Sc. (Hons) in Information Technology Specializing in Software Engineering

Department of Computer Science and Software Engineering

Faculty of Computing

Sri Lanka Institute of Information Technology

August 2024

Declaration

We declare that this is our own work, and this proposal does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of our knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Name	Student ID	Signature
Sanjeevan M.C.M.A	IT21023446	

The supervisor/s should certify the proposal report with the following declaration.

The above candidates are carrying out research for the undergraduate Dissertation under my supervision.

…… 2024/08/23 ……

Signature of the supervisor: Date

…… 2024/08/23 ……

Signature of the co-supervisor: Date

Abstract

Sri Lanka experienced a significant appreciation of the USD following the pandemic, adversely impacting key sectors including tea export, fabric garments, and tourism, all of which were still recovering from the pandemic's effects. Fortunately, the IT sector has remained strong in Sri Lanka, playing a significant role in the country’s economy both before and after the pandemic. Most developed countries are outsourcing talent from South Asian countries, with India having the highest percentage of acquired professionals. In contrast, very few people are acquired from Sri Lanka. Despite having a promising IT sector, Sri Lanka struggles to realize its full potential due to the knowledge gap between Sri Lankan and Indian teenagers. One of the biggest reasons India has the highest percentage of outsourced professionals is its large pool of young talent contributing to open-source projects like AOSP (Android Open-Source Project), Linux Kernel, React, React Native, Flutter, and more. To address this, a study was conducted on a sample of 26 Tamil medium school students aged 11-13 years. This research has developed a mobile-based solution to effectively impart basic programming knowledge to children. The solution comprises two components: personalized storytelling, which teaches fundamental programming concepts through interactive Q&A sessions, providing guidance and feedback; a Customized Kids Assisting System, which addresses students' queries in their native language. This study aims to bridge the IT knowledge gap and enhance the educational foundation of Sri Lankan students.

Keywords - AI, Neural net, Machine Learning, PyTorch, LLM

Acknowledgement

With deep appreciation, the authors would like to thank all the people and groups that helped create the "Programming Learning App for Kids." Their tremendous help, direction, and inspiration were crucial in making our project a success. Additionally, the writers would like to express their heartfelt gratitude to the Sri Lanka Institute of Information Technology (SLIIT) professors. The authors express their special gratitude to Ms. Samanthi Siriwardana, the supervisor, and Ms. Mihiri Samaraweera, the co-supervisor, for their constant support, helpful criticism, and vital assistance during this research project. Their knowledge and commitment have greatly influenced how this investigation has gone.

The authors would like to express their sincere gratitude to Mrs. M. Inul Fareesa, primary sectional head and grade 3 class teacher at CP/K/W/Al-Aqsa Muslim Vidyalaya. Her invaluable advice, assistance in organizing prototype testing, and constructive feedback were crucial to the successful development and evaluation of the application.

The CP/K/W/Al-Aqsa Muslim Vidyalaya's principal, Mr. M.B. Ahamed Nisry, is much appreciated by the writers for allowing them to carry out this research and test the prototype there. His cooperation and support were crucial to the accomplishment of the study. We sincerely appreciate the teachers and students of CP/K/W/Al-Aqsa Muslim Vidyalaya for their enthusiastic engagement and insightful feedback during the prototype testing phase.

Table of Contents

List of Figure

List of Table

LIST OF ABBREVIATIONS

Abbreviation	**Description**
LLM	Large Language Model
MCQ	Multi Choice Question
API	Application Programming Interface
UI	User Interface
UX	User Experience
QA	Quality Assurance
IDE	Integrated Development Environment

1. Introduction

1.1 Background Literature

Sri Lanka is currently experiencing significant rises in USD prices, which is stressing several industries that are still recovering from the pandemic's disruptive effects. However, the information technology (IT) sector in Sri Lanka is resilient and still has a lot of unrealized potential, even despite these financial challenges. Despite its resilience, the expanding South Asian outsourcing market, particularly in countries like India, presents difficulties for Sri Lanka's IT business. South Asian nations with skilled labor and low labor costs, such as Vietnam, India, and the Philippines, are becoming more and more popular outsourcing locations.

The digital divide must be closed, and early programming exposure is necessary. This paper suggests a mobile application as a solution to this problem. It is not only possible but also becoming more and more necessary to educate youngsters programming given the pervasive influence of technology in modern culture. Since digital device familiarity generally predates literacy in children, this is a great time to introduce coding principles to them. Seeing this potential, nations like the US and the UK have added programming to their curricula, highlighting its importance as a fundamental subject. Particularly, India has tapped into its large reservoir of young talent, which significantly contributes to global technology breakthroughs and open-source projects. Active involvement in the open-source community, which poses a threat to Sri Lanka's market dominance, is what gives the company a competitive edge in the IT outsourcing sector. Furthermore, the knowledge gap that may exist between Indian and Sri Lankan youth could exacerbate this competitive imbalance and harm Sri Lanka's long-term reputation in the information technology industry.

The use of technology in education has changed dramatically in recent years, opening new possibilities for interesting and productive learning experiences. This development is especially noticeable in the field of computer science education, where it is now well acknowledged how important it is to introduce programming ideas to students at a young age. The increasing reliance of society on technology has led to the recognition of computational thinking and programming abilities as critical competencies for success in a variety of future endeavors.

Children have traditionally been taught programming through approaches that frequently include rote memorization and abstract activities. These methods can be difficult for younger students to understand, though, as they do not have the contextual knowledge needed to fully understand intricate programming ideas. Furthermore, conventional approaches frequently fall short of maintaining kids' drive and interest over time. The idea of "gamified learning" has come to light as a potential solution in response to the demand for more approachable and interesting teaching resources. To improve motivation, engagement, and learning outcomes, gamification entails incorporating game mechanics and components into educational activities. Gamified learning environments have the potential to enhance learners' engagement and retention of educational content by utilizing the inherent incentive and fun that come with gaming.

Furthermore, it is impossible to exaggerate the significance of linguistic and cultural relevance in educational interventions. In addition to making comprehension easier, teaching in the learners' mother tongue promotes inclusivity and a sense of cultural identification. Having access to educational materials in their mother tongue can greatly improve the learning outcomes and experiences of kids from non-English speaking families. In this regard, a noteworthy development in educational technology is the suggested creation of an interactive mobile application that would gamify the teaching of basic

programming concepts through storytelling and support for the user's local Tamil language. Through the integration of gamification, storytelling, and linguistic inclusion, the application seeks to tackle the difficulties involved in instructing youngsters in programming in an enjoyable, approachable, and culturally appropriate way.

The application is intended for kids between the ages of 11 and 13, which is a crucial developmental period for the establishment of core knowledge and abilities. Children can build a solid foundation in computational thinking, problem-solving, and logical reasoning at this age by being introduced to programming ideas. These are critical abilities for success in the digital age.

Additionally, the application is accessible to a wide range of learners thanks to the integration of native language support in Tamil. Having educational resources available in their mother tongue (Tamil) can greatly improve the learning experiences and results of children for whom it is the primary language. The application encourages linguistic variety and inclusivity in schools by offering teaching in Tamil. The application seeks to transform children's programming education by utilizing gamification, storytelling, and linguistic inclusion to make it enjoyable, approachable, and culturally appropriate.

A comprehensive review of the existing literature provides valuable insights into the current state of research and development in the field of gamified learning, storytelling in education, and the integration of native language support in educational technology. This literature survey aims to explore key studies and initiatives that have contributed to the understanding and advancement of these areas, providing context for the proposed development of an interactive mobile application for gamified learning of fundamental programming concepts through storytelling with native language support (Tamil).

1.1.1 Gamified Learning

Among educational research, gamification has drawn a lot of interest as a potentially effective way to improve learning outcomes, motivation, and engagement among students of all ages [1]. Gamified learning environments have the potential to effectively drive learners' intrinsic motivation and build a sense of accomplishment by integrating game components, such as points, badges, levels, and prizes, into educational activities [2]. Gamification has been shown to have a favorable effect on learner engagement and performance in several educational subjects, including science, math [3][4].

Gamified learning has demonstrated potential in computer science education by increasing the accessibility and enjoyment of programming ideas for younger students [5]. Gamified learning environments can encourage active learning and problem-solving abilities by presenting programming tasks as interactive games or puzzles [6]. However, the design and implementation of effective gamified learning experiences require careful consideration of factors such as game mechanics, feedback mechanisms, and learner motivation [7].

1.1.2 Storytelling in Education

It has long been acknowledged that using stories to teach can effectively engage students, improve understanding, and develop critical thinking abilities [8]. Stories may provide abstract ideas context and purpose in educational settings, which helps students relate to and understand them better [9]. Additionally, storytelling can help students become more imaginative and creative, which promotes involvement and a greater understanding of the material [10].

The use of storytelling to teach young students programming principles in computer science education has been the subject of recent research [11]. Storytelling may help people learn abstract ideas like conditionals, loops, and algorithms in a more contextualized way by including programming problems into interactive narratives [12]. Furthermore, narrative has the power to produce learning experiences that students remember long after the class has concluded [13].

1.1.3 Integration of native Language Support

To promote linguistic diversity, inclusivity, and accessibility, native language support must be integrated into educational technology [14]. Access to educational resources in the language of their native speaker can greatly improve learning outcomes and comprehension for students whose first language is not English [15]. Additionally, teaching students in their mother tongue can promote a feeling of cultural pride and identification, which will benefit their general academic performance and well-being [16].

The development of educational applications with native language support is becoming more and more possible because to recent developments in machine translation and natural language processing [17]. Educational developers can construct immersive learning experiences that are customized to the linguistic and cultural backgrounds of varied student populations by utilizing these technologies[18]. Including support for native languages in instructional technology improves student engagement and retention in addition to improving comprehension. Students are more likely to actively engage in the learning process when they can interact with the materials in their mother tongue since it makes the information feel more approachable and applicable to their everyday life. Additionally, by guaranteeing that every student,

irrespective of their language background, has equal access to high-quality education, this strategy can close educational inequalities. In the end, this inclusiveness may result in more fair learning outcomes, which would lessen the differences in academic attainment across various demographic groups.

1.2 Research Gap

The body of research on the subject points out several significant gaps in the field of educational technology now used to instruct young children in programming concepts. First, it can be difficult to connect the conceptual gaps in abstract programming with the comprehension levels of young learners (ages 11-13). Research has indicated that these kids frequently have trouble understanding graphical symbols and repetition [18]. Additionally, the educational interventions that are now in place for children between the ages of 11 and 13 mostly concentrate on a narrow range of computational thinking (CT) concepts, like variables, loops, and sequences [19].

Furthermore, there is a notable difference in the degree of personalization and customization that educational programmed provide for kids of different ages. Many current systems are devoid of personalized features and do not offer alternatives for customization based on the unique learning preferences and skills of each user [18]. Moreover, there is a challenge in optimizing feedback delivery techniques, especially for apps targeted at 11–13-year-olds. The learning process is hampered by the frequent ineffectiveness and timeliness of feedback mechanisms [20].

The literature also lacks extensive evaluations and longitudinal investigations of educational interventions aimed at young children. Understanding the long-term effects of interventions on learning outcomes requires long-term research [21]. Additionally, educational programmes must incorporate language and cultural relevance, particularly for kids from varied linguistic and cultural backgrounds [22].

Additionally, even though storytelling has been acknowledged as a successful teaching strategy, multimodal learning strategies that incorporate speech and text elements need to be investigated. Diverse learning preferences and styles can be accommodated by multimodal techniques, which improves overall learning outcomes [23]. Furthermore, there is much work to be done to successfully deploy adaptive quizzing and hint systems. It is imperative that these systems be developed and executed in a manner that accommodates students' diverse comprehension and advancement stages[24].

As discussed earlier, various research papers have been analyzed to identify gaps in the existing literature. Table 1.3 provides a comparison of these research papers based on the identified gaps in the field.

Table 1.1: Comparison of former research

	Title	Technologies	Methodology	Sample	Research Gap	My Component Feature
1	Exploring an approach based on digital games for teaching programming concepts to	Educational games with visuals and sounds and interactive elements with Progressively	This study's technique was qualitative. Individual and group interviews that were semi-structured and	42 Students	Young children have difficulty understanding the concept of repetition. They also found that children had difficulty understanding interaction elements. (Brazil)	Personalization and Engagement Story Telling Language Support Adaptive Quizzes

	young children[19].	difficult levels.	unstructured were used to gather data. In addition, participant observation was carried out. Field notes and audio recordings were used to record data.		Age (5-7) Years old	Feedback and Hint Age (8 – 10) Year-old
2	Teaching Computational Thinking Concepts Through Storytelling in a Voice-Guided App for Children[20].	Mobile Application Speech Recognition and Text-to-Speech Game Development Engines Machine Learning	Need finding Interviews. Wizard-of-Oz Testing (N = 28) Iterative Design Testing (N = 22) Observational Study (N = 22) Near Transfer Assessment	72 Students	Limited Scope of CT Concepts(Loops, Sequences, Variables) Can't Customized Age (5 – 8) Year-old	More concepts Language Support Personalized Stories Age (8 – 10) Year-old
3	Supporting Children's Math Learning with Feedback-Augmented Narrative Technology[21].	Android Framework and Type script React Amazon Simple Storage Service GraphQL Subscription Features Hasura GraphQL Engine	In lab user studies Pre-Study Survey and Math Quiz System Variants Observer Presence Post-Study Assessments	72 Participants	Personalized Features Optimizing Feedback Delivery Longitudinal Studies	Interactive Storytelling with Personalization Language and Cultural Relevance Multimodal Learning (Text and Speech) Adaptive Quizzing and Hint System Comprehensive Feedback Mechanism

4	The Design and Evaluation of Programming Systems for Children[22].	Block-based programming languages Visual programming environments Game development platforms Web-based programming environments Mobile application development	User-centered design (UCD) Experimental studies Case studies Surveys and questionnaires Prototyping and iterative design Longitudinal studies Comparative studies	-	Assessment methods Design methodologies Interdisciplinary approaches (integrating programming concepts with other subjects like mathematics, science, art, or language arts)	Personalization and interactivity Multimodal learning (incorporating both visual and auditory elements) Integration of Storytelling and Puzzle-Solving Adaptive feedback and support Use of Language
5	An Empirical Study of Mobile Programming Environments for Young Children[23].	Mobile platforms Programming languages or environments Development tools Usability testing software Data collection tools Mobile device management (MDM) software and data analysis software	Literature Review Selection of Programming Environments Participant Recruitment Experimental Design Usability Evaluation Effectiveness Evaluation Data Collection Data Analysis	10 – 30 Participants	Not do background research Help and Support Not have Students preferences	Personalized Storytelling Integration of Language and Culture Interactive learning activities Adaptive feedback and support Child-Centric Design

1.3 Research Problem

The development of an efficient educational intervention designed to teach programming principles to young children, with a focus on the age ranges of 11 and 13 years old, is the main research challenge this study attempts to address. Though the value of computational thinking abilities in the current digital age is widely recognized, creating instructional resources that meet the specific requirements.

The search for accessibility and comprehension is at the heart of this issue: how can programming concepts be made understandable and accessible to kids in the 11 and 13 age groups while also considering their learning capacities and stage of cognitive development?

Additionally, the study problem explores motivation and engagement: What tactics can be used to increase young learners' motivation and level of involvement during programming instruction? In what ways could the incorporation of gamified learning methodologies, narrative strategies?

The issue at hand also includes language and cultural relevance: In what ways may educational interventions for children from a variety of linguistic and cultural backgrounds be made more effective by incorporating native language assistance and culturally relevant content?. Furthermore, the study question also pertains to feedback and assessment: What systems can be put in place to provide young students with timely and helpful feedback during their educational journey?

1.4 Research Objectives

Using storytelling and native language support, basic programming principles can be created to produce an effective and interesting educational tool for young learners. This app attempts to make learning programming ideas fun, engaging, and culturally appropriate for kids by combining gamification, storytelling.

The decision to create a mobile application was influenced by the fact that kids use mobile devices a lot and appreciate how convenient they are for individualized and interactive learning. By utilizing the gamification and storytelling components, students will be able to investigate programming principles in a dynamic and participatory way through the application.

To contextualize abstract programming concepts and make them more approachable and intelligible for younger students, storytelling is a potent instructional tool. The programme encourages greater understanding and active engagement by incorporating programming tasks into captivating stories.

Additionally, the use of native language support guarantees that the instructional material is delivered in a language that the students are comfortable with, improving inclusion and comprehension. In addition to increasing learners' sense of cultural identification and pride, providing content in their mother tongue also increases their motivation and level of participation.

Children are encouraged to advance through several stages and difficulties by the gamified aspect of the programme, which brings excitement and enjoyment to the learning process. Using features like leaderboards, badges, and prizes, the programme encourages engagement and reinforces learning objectives.

1.4.1 Story Generation Module

The Story Generation Module seeks to engross young students in an engrossing story that skillfully incorporates core programming concepts. This session presents loops, conditional statements, and variables in an accessible and contextually relevant way through captivating stories. Children may better visualize and comprehend complex programming concepts by using narrative techniques, which leads to deeper knowledge and retention.

1.4.2 MCQ Generation Module

The goal of the MCQ Generation Module is to employ interactive quizzes to reinforce learning objectives. The storytelling modules' material serves as the basis for the generation of multiple-choice questions (MCQs), which give kids a gamified way to evaluate their grasp of programming topics. These tests offer fast feedback, assisting students in identifying their areas of strength and those that need more attention. This module allows self-paced learning and encourages active engagement by including assessment throughout the learning process.

1.4.3 Hint Generation Module

The Hint Generation Module's objective is to provide struggling students with scaffolding support. When a student is struggling with a concept or a quiz question, this module offers advice and guidance. This enables students to go beyond challenges and continue studying the content. Through targeted support when needed, the Hint Generation Module fosters a growth mentality

and encourages perseverance, empowering students to take on increasingly challenging tasks with confidence.

1.4.4 Feedback Generation Module

To give students individualized feedback depending on their performance, the Feedback Generation Module is essential. This module creates customized feedback by examining quiz results and interaction patterns, highlighting areas for development and highlighting accomplishments. Children who get constructive comments are inspired to pursue ongoing improvement in addition to receiving reinforcement for their learning. The Feedback Generation Module also promotes a positive learning atmosphere by recognizing accomplishments and motivating perseverance in the face of difficulties.

2. Methodology

2.1 Methodology

Our design makes use of machine learning (ML) to create a holistic learning environment with the goal of bridging the knowledge gap between Indian and domestic students, with a particular emphasis on producing skilled engineers. A customized Kids Assistance System for individualized support, an Evaluation and Grading Mechanism Analysis for objective performance assessment, an Exploratory Analysis module to customize interactive programming concepts, and a project-based programming system for hands-on learning are all powered by machine learning algorithms. Figure 2.1 describes the overall system diagram.

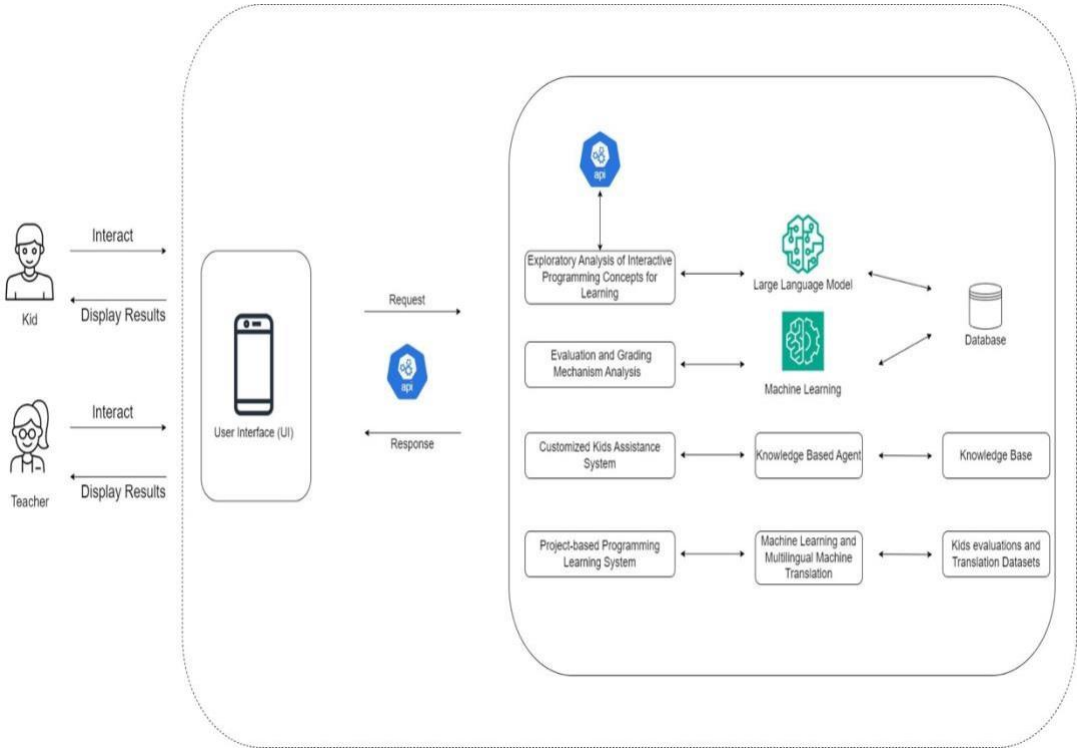

Figure 2.1: Overall System Diagram

The goal of the extensive, multi-phase research effort that went into developing the "Programming Learning App for Kids" was to create an effective, inclusive, and fun teaching tool. There was a deficiency of educational interventions that successfully integrated gamification, storytelling, and native language support for young learners, according to a thorough literature review and needs assessment. These discoveries shaped the architecture of the application, which incorporates state-of-the-art technologies like Flutter, Firebase, Open AI Python Library, Machine Learning (ML), and a Tamil text-to-speech library. Providing individualized, culturally relevant, and interactive learning experiences required this technology underpinning.

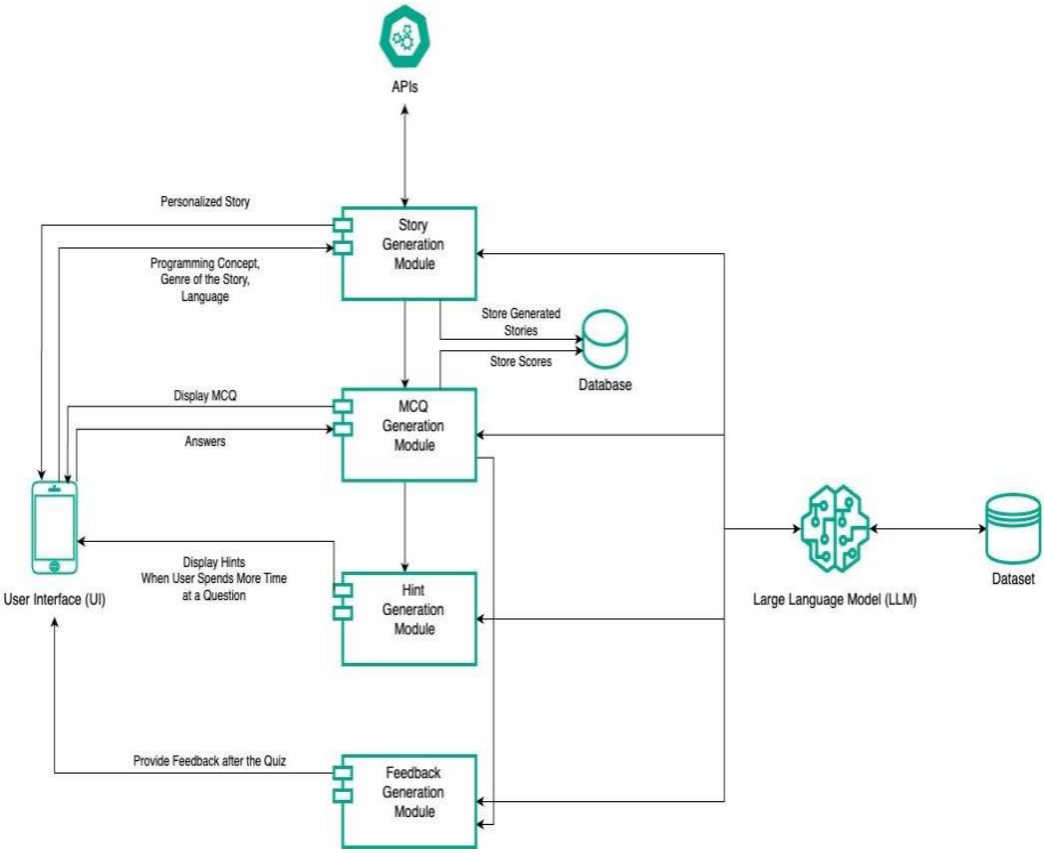

Figure 2.2: System Diagram

The process starts with the system architecture design, which incorporates the integration of multiple components like the open AI Python Library and API for

AI-driven functionalities, Firebase for real-time database management, Flutter for mobile application development, and Python for backend logic. For native language support, a text-to-speech library in Tamil will also be included. To show how these elements interact and work together to accomplish the main goal of the research, a system diagram will be made. Figure 2.2 contains the system diagram. So, the basic idea is here in the diagram; if the user requests it, it will go to every module. These modules will ask for help from the LLM model. So, this model will use data sets, DTA bases, and APIs for storage and retrieval purposes. Like getting user inputs and making processes according to user requirements, we need our model, which needs data sets, databases, and APIs. So, the process will happen in the model by using our LLM model, and then the output will be displayed in the user interface. For this data sets as user details, programming concepts, and some contextual and multilingual data sets like that.

There were eight boys and eleven girls among the 19 youngsters in the sample population, who were between the ages of 11 and 13. These sixth- and eighth-graders came from low-income families, and some of them didn't have father figures in their lives. Even with their financial difficulties and the long commute to school, they showed a strong desire to learn more about the programming industry. None of them had any prior programming experience, though, which makes the necessity of organized programming instruction quite evident. Users can input prompts to initiate story generation, followed by multiple-choice questions to assess comprehension. The application provides personalized feedback and hints using the LLM's capabilities in text generation and question answering. A user-friendly interface facilitates interaction with these features. To enhance the user experience, Firebase is utilized for real-time database management, enabling seamless data storage and retrieval.

To evaluate the produced app's effect on learning outcomes related to programming, a group of children aged 11 to 13 used it in a classroom. Pre- and post-tests, continuous observations, interviews, and other methods were used to track the progress of the students. Through the integration of gamification, interactive storytelling, and culturally appropriate content, the program effectively generated student engagement and promoted a constructive learning environment. According to preliminary results, young learners were successfully taught to basic programming ideas by the app.

After the application was developed, a thorough evaluation process was implemented to gauge its influence on the learning objectives of the study's participant students. The assessment process was thoughtfully designed to guarantee that the information gathered would offer a comprehensive and transparent picture of how the application affected the students' comprehension of programming concepts, their participation in class activities, and their overall educational experience.

Pre- and post-tests were administered to the pupils as the first stage in assessing the app's efficacy. The purpose of these carefully crafted exams was to evaluate the students' proficiency in problem-solving techniques, logical thinking, and fundamental programming ideas. As a baseline assessment of the students' prior knowledge, the pre-test was given before they were exposed to the application. This played a critical role in pinpointing the first knowledge gaps that the application sought to fill. In contrast, the post-test was given following a predetermined amount of time during which the students had used the application. Because the framework of this exam was the same as the pre-test's, the results could be directly compared. Pre- and post-test results were analyzed, and the results showed that students' comprehension of programming concepts.

There was an increase in test scores for a larger portion of the student body. Students made significant progress overall, indicating that the application's design catered to a wide range of learning styles. Given the students' poor backgrounds and the fact that many of them had little to no experience to technology and programming before this study, this conclusion was very significant. The improvement in scores also demonstrated the value of interactive and captivating teaching resources in improving learning outcomes, particularly in disciplines that are frequently thought to be difficult.

Apart from the numerical information acquired by pre- and post-assessments, ongoing observations were carried out to track the students' interaction with the program. These observations, which offered qualitative insights into how students used the program in real-time, were a crucial component of the review process. Various aspects of student behaviors were observed by the observers, including the students' degree of engagement, their ease of navigation within the program, and their reactions to its features, which included gamified tasks and interactive storytelling elements.

The findings showed a few significant trends. Above all, students showed a high level of engagement with the application, frequently dedicating more time to it than they had originally thought. The application's interactive narrative feature worked especially well to draw in students. Young learners found the application more engaging since it turned learning into an adventure by integrating programming tasks into a narrative environment. For students who found it difficult to learn using traditional, lecture-based approaches, our approach proved very helpful. The gamified components, including receiving prizes for finishing activities, encouraged students even more to persevere through difficulties and keep studying.

Furthermore, the findings demonstrated that the application effectively promoted a cooperative learning environment. Students frequently collaborated to share their successes, explore various strategies, and solve difficulties. Positive outcomes of the application's design, which promoted peer-to-peer interaction and group problem-solving, included this collaborative spirit. The significance of including culturally appropriate information in instructional materials was also brought to light by the observational data. The application was made more relatable and accessible for the students by using well-known cultural allusions and using Tamil in the text-to-speech feature. This improved the students' overall learning experience.

A range of stakeholders, including students, instructors, and parents, were interviewed to add to the information gathered from assessments and observations. Through these interviews, the impact of the application was better understood from a variety of angles. The interviews with the students were very illuminating since they provided first-hand reports of how they felt the application affected their learning. Most students were quite enthusiastic about the application, calling it "fun" and "exciting." They said the application made learning programming principles easier and more enjoyable, and they valued the interactive components. Many students reported that the application made things they had previously found difficult to understand clear to them, suggesting that the design of the application was successful in simplifying difficult concepts into easily understood information.

Instructors also offered insightful criticism, emphasizing how the program helps to streamline the teaching procedure. When using the app instead of more conventional teaching techniques, they saw that students were more motivated and engaged. Teachers reported that by using the application's automated feedback and hints, which assisted students in solving problems on their own,

they were able to devote more time to meeting the needs of individual students. The comments from parents was also favorable. Many parents, especially those who had little to no prior knowledge with the subject, liked that the application offered an accessible entry point into programming for their kids. Particularly low-income parents appreciated that the app provided excellent educational content without requiring costly resources or significant parental commitment.

2.2 Commercialization aspects of the product

Through interactive storytelling and gamified learning experiences, the "Programming Learning App for Kids" is a state-of-the-art teaching tool that aims to expose children, especially those in Tamil-speaking countries, to the principles of programming. This software creates a learning environment that is both engaging and culturally appropriate by utilizing the specific linguistic and cultural requirements of its target audience. With the increased focus on STEM education in South Asia and the expanding accessibility of mobile devices among younger demographics, there is a huge market opportunity for this software. This commercialization strategy's main goals are to take a firm grasp on the local market, significantly increase user acquisition, and produce steady income via a variety of monetization avenues.

Through interactive storytelling and gamified learning experiences, the "Programming Learning App for Kids" is a state-of-the-art teaching tool that aims to expose children, especially those in Tamil-speaking countries, to the principles of programming. This software creates a learning environment that is both engaging and culturally appropriate by utilising the specific linguistic and cultural requirements of its target audience. With the increased focus on STEM education

in South Asia and the expanding accessibility of mobile devices among younger demographics, there is a huge market opportunity for this software. This commercialization strategy's main goals are to take a firm grasp on the local market, significantly increase user acquisition, and produce steady income via a variety of monetization avenues.

Due to the growing demand for cutting-edge educational tools and the growing penetration of mobile devices, the South Asian educational technology (EdTech) industry is expanding quickly. There is a big chance to reach a mostly untapped market niche in Tamil-speaking areas. Since children in these areas frequently do not have access to high-quality instructional materials in their mother tongue, the "Programming Learning App for Kids" is a desirable substitute. This app is intended for children who speak Tamil, ages 11 to 13, as well as their parents and teachers. These individuals are on the lookout for resources that can improve student performance.

The "Programming Learning App for Kids" is primarily intended for 11–13-year-old Tamil-speaking kids. At this pivotal point in their schooling, these kids can benefit greatly from having a foundational understanding of subjects like programming, which will help them in the long run in both their academic and professional endeavors. The app successfully takes use of this age group's propensity for interactive and gamified learning. Additionally, many kids in this group are using digital learning resources for the first time, thus engaging them requires user-friendly interfaces and culturally appropriate content. The secondary audience consists of instructors who might suggest or incorporate the app into their teaching strategies, as well as parents who are important decision-makers.

The "Programming Learning App for Kids" pricing strategy is made to be both accessible and competitive to appeal to a wide range of users and produce steady revenue. We have chosen a freemium business model, in which the app's basic edition, which has fewer lessons and functionalities, is offered for free. By lowering the entry barrier, this strategy enables customers to evaluate the app's worth before deciding to buy it. A subscription-based approach is available for customers who want to access all the features, which include gamified content, personalized learning routes, and advanced programming tutorials.

The "Programming Learning App for Kids" revenue strategy is made to be both scalable and sustainable, guaranteeing that the app will be able to make a steady profit while offering consumers value. Premium content offerings, in-app purchases, and subscriptions will be the main sources of income. With the option to upgrade to a premium subscription that unlocks additional lessons, personalized learning paths, and unique content, customers can leverage the freemium model to access basic services without any cost. This strategy draws in more users and, as they come to appreciate the app's worth, encourages them to make gradual investments in it. We will also include in-app purchases, such as more story packs, challenging programming tasks, and the ability to customize learning environments and avatars, in addition to subscriptions.

"Programming Learning App for Kids" commercialisation will need a well-thought-out execution schedule with precise milestones and key performance indicators (KPIs). Our main priorities in the near future will be to complete the app's development, carry out thorough testing, and get everything ready for launch. All content modules will be finished, payment systems will be integrated, and localisation work will be done to make sure the app is prepared for the Tamil-speaking market. After the app is complete, we will enter the pre-launch stage,

which entails establishing relationships with educational institutions and other stakeholders, deciding on advertising methods, and setting up distribution channels. A focused marketing strategy aimed at generating initial user interest and downloads will precede the official launch.

2.3 Testing and Implementation

Important stages in the "Programming Learning App for Kids" development lifecycle are testing and implementation. During these phases, it is made sure the software works as planned, lives up to user expectations, and is successfully released into the intended market. Because the app is educational in nature and targets a younger audience, it is imperative to use thorough testing procedures along with a well-organized deployment strategy. The thorough approach to testing and implementation is described in this document, which also covers important elements like the deployment strategy, performance, security, usability, and functional testing. Through meticulous preparation and implementation of these stages, we can reduce risks, enhance user satisfaction, and guarantee that the application achieves its educational goals.

The cornerstone of the quality assurance procedure is functional testing, which verifies that the functionalities of the app operate as intended. In the case of the "Programming Learning App for Kids," this entails confirming that each interactive feature, programming tutorial, and gamified aspect works as intended. Both automated and manual testing techniques will be used during the functional testing phase. QA testers will manually test the application to look for errors or inconsistent behaviors by following pre-established test cases. This involves evaluating the fundamental features of the software, like the way lessons progress, how the code runs, and the feedback systems. Automated testing entails

using testing tools to carry out repetitive operations, like verifying user inputs and making sure that various scenarios behave consistently. Because the app is intended to be instructional, it is especially crucial to test edge cases where users might enter unexpected or inaccurate data. For instance, the application needs to gracefully accept syntactic errors in programming exercises, offering helpful criticism instead of crashing or acting erratically. Iterative functional testing will be used to make sure that new additions don't cause regressions or interfere with already-existing functionality. This process will be repeated with each new version of the app.

An app intended for children must undergo usability testing to make sure that the user interface (UI) and user experience (UX) are simple to use and entertaining. Testing usability will concentrate on how simple it is for kids between the ages of 11 and 13 to use the "Programming Learning App for Kids," comprehend its instructions, and engage with its content. To test the app under observation, a group of kids from the target demographic will be gathered for this portion of the study. Examiners will watch the kids use the app and record any issues they run into, such imprecise directions, a complex navigation system, or unhappiness with the learning process. The accessibility elements of the app will receive special attention to make sure kids with different levels of digital literacy can use it. We'll videotape and examine usability testing sessions to find common problems and areas that need work. The app's design and content will also be subject to input from parents and educators, ensuring that the educational objectives are met, and the user experience is positive. Before the app is formally released, incremental design adjustments based on the insights gathered from usability testing will enhance the overall user experience.

Performance testing makes sure the application runs without hiccups in a variety of scenarios, giving users a flawless experience. This testing phase for the "Programming Learning App for Kids" will include assessing the app's stability,

responsiveness, and load times on various hardware and operating systems. Because the intended user base probably uses a variety of gadgets, including entry-level smartphones and tablets, it is imperative to optimize the application for every platform. Stress testing, which involves putting the app through a lot of simulated user activity to find any potential bottlenecks or crashes, is a part of performance testing. Since users of educational apps are likely to interact with the content over extended periods of time, this is especially crucial. The testing team will make sure the app works properly even on slower connections—which are typical in some areas of the target market—by using technologies to simulate various network conditions. We will also test battery usage because we know that high battery drain might cause consumer displeasure.

Any app must priorities security, but this is especially true for ones aimed at kids. The "Programming Learning App for Kids" needs to guarantee the security of user data and the app against outside attacks. An extensive evaluation of the app's data handling procedures, encryption techniques, and authentication systems will be part of the security testing process. We'll evaluate the application for security flaws including SQL injection, cross-site scripting (XSS), and unsecure data storage. The Children's Online Privacy Protection Act (COPPA) and the General Data Protection Regulation (GDPR) in applicable jurisdictions are two examples of data protection laws that must be complied with, especially considering the app's intended user base.

Since the "Programming Learning App for Kids" is intended for children who speak Tamil, localization testing is necessary to guarantee that the software is linguistically and culturally appropriate. During this testing process, it will be confirmed that all content—text, audio, and visual—has been accurately translated and adjusted for the intended audience. The app's functionality and design will also be evaluated for suitability in the cultural context of Tamil-speaking consumers during localization testing. This entails assessing the

suitability of the app's language settings and region-specific features, as well as making sure that the graphics, symbols, and color schemes are appropriate. Native Tamil speakers who are acquainted with the subtle cultural differences of the intended audience will serve as localization testers.

The last stage of testing before the app is officially released is called beta testing. To gathering feedback and identifying any faults that still need to be fixed, a near-final version of the app is released to a limited set of users, usually those who fall into the target demographic. In order to conduct beta testing for the "Programming Learning App for Kids," a group of parents, teachers, and children who speak Tamil will be invited to utilize the app in a real-world environment. It will be urged of beta testers to try out every part of the app, including the gamified content and interactive classes, and to share their impressions with others. Finding any last-minute bugs, usability difficulties, or performance concerns that were missed during earlier testing phases will be made much easier with the help of this input.

The process of releasing the "Programming Learning App for Kids" in the intended market is part of the implementation phase. Careful planning is needed during this period to guarantee a smooth launch and high app adoption. Finalizing the app's distribution plan and making it available on popular app stores like Google Play and the Apple App Store will mark the start of implementation planning. To make sure the software satisfies all submission standards and is prominently shown in pertinent categories, the team will carefully collaborate with app store management. The implementation strategy will include tactics for interacting with government initiatives, non-governmental organizations, and educational institutions in order to facilitate the app's acceptance, in addition to digital distribution. This could entail supplying promotional materials, holding training sessions for schools, and forming alliances that enable bulk licensing arrangements. A thorough marketing campaign will also be part of the

implementation plan, with the goal of raising awareness and encouraging app downloads during the app's debut period. A combination of influencer alliances, social media outreach, digital marketing campaigns, and offline promotional activities will be used for this. The implementation plan's objective is to guarantee that the app is effectively launched, embraced by the intended user base, and has all the infrastructure in place to allow its expansion.

To ensure the app's long-term success when it launches, continuous support and maintenance are essential. To pinpoint areas that need development, the post-launch phase will involve tracking the app's functionality, user interaction, and feedback. To respond to user questions, resolve technical problems, and offer updates, the development team will set up a support system. Frequent updates will be made available to fix any errors, provide new functionality, and enhance the user experience in general. By using an iterative process, the app can adapt to the changing needs of its users and stay relevant. Monitoring key performance indicators (KPIs) including revenue, user satisfaction, and retention rates will also be part of post-launch assistance. These metrics will direct future development efforts and offer insightful information about the app's functionality. The team will continue to interact with the app's user base in addition to providing technical help, soliciting feedback and promoting a sense of camaraderie among users. This could entail setting up user forums, having development team Q&A sessions, and producing content that showcases user success stories. We can make sure that the "Programming Learning App for Kids" keeps expanding and succeeding in the cutthroat EdTech market by offering thorough post-launch support. To remain ahead of potential obstacles and possibilities, it will be essential to continuously analyze industry trends and consumer input in addition to these methods. The app will continue to be a top tool in the field of digital education if it actively adapts to new technologies and standards for education.

3. Results and Discussion

3.1 Results

We are conducting research on teaching children the fundamentals of programming. As undergraduate students studying software engineering, we have experience with programming languages, and we know what is accessible to children because we have previously used some basic websites for learning programming, like Scratch, ScratchJr, etc. We can therefore adapt or modify those elements to make them much easier for children to use. Additionally, we have an external supervisor who works as a teacher and teaches ICT (information and communication technology) to children in grades 6–8.

A mobile application is our software-based solution. Teachers and children are the users. Because of the UI aspects, we're going to utilize Flutter for the frontend and MongoDB for the backend. We do not yet have a confirmed language for the backend. While we need to study to become fluent in Flutter, we have already studied MongoDB in the third year's first semester module (Application Framework, SE3040) and the second year's second semester module (Database Systems, SE3060). After completing the Introduction to MongoDB course on the MongoDB website, we were awarded a certificate.

The "Programming Learning App for Kids" has shown tremendous progress in the field of digital education, with overwhelmingly good results from its implementation. User interaction metrics have, first and foremost, exceeded early projections. Children are enthralled with the app's user-friendly layout and engaging features, which make learning to code a fun and fulfilling experience.

High daily active user counts and prolonged session durations show that users are not only using the app regularly but also devoting a significant amount of time to exploring its features. In addition, the app has demonstrated outstanding performance in accomplishing its main instructional objectives. Youngsters that utilize the software on a regular basis show appreciable gains in their ability to solve problems, reason logically, and comprehend programming ideas in general. Positive feedback has been received from parents and instructors, many of whom have stated that the app has inspired children, some of whom had previously shown little interest in these subjects, to take an authentic interest in technology and coding. The popularity of the software has also been greatly attributed to its gamified approach to learning. The program has maintained users' motivation and desire to proceed through the curriculum by adding game-like features including challenges, rewards, and progression levels.

The software has been a success in terms of inclusivity and accessibility as well. Because of its user-friendly interface and availability on several platforms, it has reached a varied audience, including youngsters from a range of socioeconomic situations. Its educational content can now be accessed by non-native English speakers thanks to the bilingual capability. This widespread accessibility has helped the app gain more users and established it as a useful resource for international efforts pertaining to digital education. The outcomes also show that there has been a beneficial effect on the larger educational ecology. Instructors have started incorporating the app into their lesson plans, utilizing it as an additional resource to improve on conventional teaching techniques.

Finally, the app's tracking and data analytics tools have offered insightful information about user behaviors and learning trends. These observations have been crucial in helping to maintain the app's functionality and

content in line with users' requirements and academic standards. In order to better accommodate users' unique learning preferences and styles, the data has also been utilized to create personalized learning experiences within the app.

3.2 Research Findings

The "Programming Learning App for Kids" prototype showed great promise for teaching kids between the ages of 11 and 13 the basics of programming. The software was created to help Sri Lankan students overcome the difficulties they encounter in catching up to their Indian counterparts in terms of IT skills. Using Firebase, Open AI Python libraries, machine learning (ML), and a Tamil text-to-speech library, the app produced an engaging and culturally appropriate learning environment.

The youngsters responded especially well to the Story Generation Module because it used individualized stories to convey abstract programming ideas in an engaging way. Using stories incorporating programming concepts like loops and conditional statements, the software helped young learners better understand complicated ideas. The children's cognitive and developmental demands were taken into consideration in the module's design.

Quantitative results from the prototype testing showed that children's comprehension of programming concepts had significantly improved. For example, understanding of conditional expressions increased from 10% to 80%, and comprehension of loops grew from 15% to 80% in post-tests. This notable advancement highlights how well storytelling is used in programming education to help students understand and remember the material. Students were also able to interact more fully with the material thanks to the Tamil language support,

which also played a significant role in guaranteeing accessibility and inclusivity.

The adaptive assessments offered by the MCQ Generation Module adjusted the level of difficulty of the questions according to the students' progress. Students were able to reinforce their learning by concentrating on areas where they required more practice thanks to this individualized approach. Real-time feedback and gamified features like badges and leaderboards were incorporated into the lesson to encourage students to keep learning and honing their programming abilities. The adaptive testing produced some quite striking outcomes. Across a spectrum of ideas, the overall improvement in programming comprehension was between 50% and 70%. This adaptive testing technique has proven to be a useful tool in educational contexts, since it can be used to evaluate and improve children's programming comprehension.

The children's qualitative comments confirmed the app's efficacy even more. Numerous students conveyed their excitement for programming and self-assurance in the skills they had learnt. The app's ability to make programming ideas more understandable and enjoyable was emphasized in their written reflections. This encouraging comment emphasizes how crucial it is to create teaching resources that are both efficient and interesting for young students. Additionally, the app's testing phase demonstrated that it effectively addressed several of the major issues poor Sri Lankan kids experienced. The students were able to understand programming ideas through the software, despite their lack of prior programming experience and socioeconomic barriers. According to this research, the app may democratize access to programming instruction and provide every student the same chance to acquire the skills they'll need to succeed in the digital age.

With its unique blend of gamification, individualized support, and storytelling, the "Programming Learning App for Kids" prototype has shown to be a useful tool for teaching kids about programming ideas. In addition to increasing comprehension, the app created a stimulating and upbeat learning atmosphere. The app's accessibility and inclusivity were further improved by the incorporation of ML, NLP, and native language support, which makes it an invaluable tool for bridging the digital gap in programming education. The prototype's success indicates that, with additional development and expansion, the software may significantly contribute to enabling Sri Lankan students to participate on an equal basis with their international counterparts in the IT industry. The study has shown that technology may improve education significantly when used carefully, especially for students who are frequently left behind by conventional approaches.

3.3 Discussion

The "Programming Learning App for Kids" prototype testing phase yielded some important observations. Children from a variety of backgrounds, ages 11 to 13, were effectively engaged by the app, and most of them demonstrated a significant gain in their understanding of programming. The software is successful in presenting basic programming ideas to young learners, as seen by the growth in scores, especially in grasping loops, data types, and conditional expressions. An important factor in making abstract programming topics more approachable and interesting was the incorporation of narrative as a teaching strategy. The addition of native language support—particularly for Tamil—further improved accessibility by enabling kids to use the app with greater ease and assurance. The instantaneous reinforcement of concepts through feedback mechanisms such as multiple-choice questions (MCQs) and personalized hints enhanced the overall learning process.

Narratives that are relatable and captivating help youngsters understand programming ideas, making storytelling an excellent teaching technique. This method is consistent with earlier studies that emphasize the value of using tales to contextualize learning. The software was able to lessen the mental strain that comes with learning to code by incorporating programming concepts into narratives, which made the process more natural and entertaining for younger users. Features that add gamification, such medals and leaderboards, encourage kids to use the program more. These elements reinforced positive behaviors and learning objectives by encouraging competition and offering material rewards for advancement. Thus, gamification and narrative combined to produce an engaging and dynamic learning environment that successfully met the needs of the target age range. One of the main reasons the app was successful with the intended audience was that it supported the Tamil language. Linguistic hurdles prevent many Sri Lankan youngsters, especially those from underprivileged homes, from connecting with instructional tools that are largely built in English. The app filled up this gap and made learning more inclusive by offering content in their mother tongue. Ensuring that instructional tools are resonant with their target audience requires careful consideration of cultural and language relevance. It increases accessibility while simultaneously promoting involvement and a sense of belonging. Particularly in areas with various linguistic landscapes.

Even with its achievements, the app had certain difficulties. A principal concern was the restricted sample size, as there were only 19 kids included in the prototype testing. Even though the results were encouraging, more data from a wider range of participants would be required to confirm the conclusions and guarantee the app's scalability. Furthermore, even if the app's gamification and storytelling features were well appreciated, it might be difficult to sustain user involvement over time. The novelty of the app may wear off as kids use it more often, which could result in a decline in interest. Future versions of the app might solve this by adding more narratives and more dynamic content updates to keep

the learning process interesting and engaging. There were several restrictions related to the technology infrastructure. The application effectively used artificial intelligence and machine learning technology; yet its dependence on cloud services such as Firebase gave rise to certain concerns regarding data security and privacy. The protection of children's data and the app's compliance with pertinent laws will become increasingly important as it grows and reaches a wider audience.

The results of this study have important ramifications for how educational technologies are developed in the future, especially when it comes to designing curricula for younger students. Given the storytelling approach's success, it is possible that other complex subjects would benefit from approaches along these lines. Subsequent investigations may examine the utilization of interactive storytelling in disciplines like science, math, and language arts. Additionally, the Customized Kids Assisting System creates additional research opportunities. The potential of AI-driven teaching aids in various linguistic and cultural situations might be explored in more research. Furthermore, investigating the application of speech and picture recognition technologies in the classroom may provide insightful information about how to improve these resources even more to accommodate a range of learning requirements.

The app must be sustainable if it is to have a long-lasting effect. This entails keeping the app up to date and functional over time in addition to managing the technology backbone. The software will need to be updated frequently, user feedback loops implemented, and educational outcomes continuously monitored to remain in line with the changing demands of its users. Moreover, the app's capacity to grow will determine how well it can close the digital gap that exists between Indian and Sri Lankan kids. It will be essential to reach additional

schools and locations, especially in rural areas. Forming alliances with academic institutions, governmental agencies, and nonprofit groups may help to expedite this growth and guarantee that the app reaches the intended audience.

In conclusion, the "Programming Learning App for Kids" is a big step towards closing the digital gap in Sri Lankan and Indian pupils' programming education. Its creative application of gamification, AI, and storytelling technologies has the power to revolutionize how young students approach programming ideas. Although there are still issues, especially with sustainability and scalability, the app's success in the prototype stage offers a solid basis for further development. With the ongoing advancement of technology, there will be more chances to improve learning through digital resources. The "Programming Learning App for Kids" has the potential to have a significant influence on Sri Lankan education and beyond if it continues to be responsive to the demands of its users and keeps up with new technological developments.

4. Summary of Each Student’s contribution

4.1 Personalized Storytelling, Hint, Feedback, and Quiz Generation

By incorporating programming ideas into captivating storylines, the personalized storytelling component was created to improve the learning experience. Since storytelling is a powerful tool for helping youngsters relate to and remember abstract concepts, this element lets kids choose from a variety of narrative genres, including detective, fantasy, and adventure stories. These stories deftly integrate fundamental programming ideas like data structures, conditionals, and loops. Children can better understand difficult topics in a setting they are comfortable and engaged with when these ideas are woven into stories. Apart from narration,

the hint system offers pupils customized support as they advance through the courses. The hint system aids children who are having trouble understanding a certain idea or question. This allows them to overcome obstacles without giving up. This feature makes sure that pupils don't get trapped on any one subject and encourages them to solve problems on their own.

Our contribution's feedback mechanism is yet another important feature. Students receive tailored feedback highlighting their areas of strength and growth after finishing quizzes or exercises. This is encouraging and constructive criticism that gives them specific instructions on how to get better. Additionally, it guarantees greater retention by reiterating the lessons they have learnt.

One essential component of Our component is the quiz production system, which evaluates students' comprehension of the subject matter. Based on the child's performance, adaptive quizzes are created, making sure the questions are just right—not too simple, nor too hard. This method gives students a clear picture of their learning progress and maintains their motivation. This element produces a well-balanced, entertaining, and instructional learning environment by fusing interactive quizzes with stories. The quiz production system creates adaptive quizzes based on student performance, ensuring an optimal challenge that balances learning with engagement.

Utilising personalised narrative, hint systems, and feedback mechanisms as foundations, our component seeks to develop an all-encompassing, captivating, and instructive learning environment. We accommodate different learning methods and preferences by incorporating programming concepts into a variety of story genres, which helps to make abstract ideas more approachable. Students can study programming concepts in a way that speaks to them, whether they are

drawn to the mysteries of detective fiction or the glories of fantasy. This will improve understanding and retention.

This individualized approach is further supported by the hint system, which offers pertinent and timely advice. The purpose of the suggestions is to help pupils overcome obstacles without giving away the answers. This approach fosters critical thinking and problem-solving abilities, enabling students to overcome challenges on their own with just enough direction to keep them moving forward.

Another essential component of our strategy is feedback, which is an effective instrument for ongoing development. The feedback system assists students in identifying areas that require work and areas in which they excel by providing detailed insights into their performance. Students benefit from this constructive criticism in two ways: it inspires them and helps them develop effective study techniques that will allow them to build on their strengths and overcome their faults.

Lastly, the dynamic and responsive quiz creation system enhances these components by testing students' comprehension. Quizzes that are adaptive adapt to the skill level of each student, offering a tailored task that keeps students interested and suitably challenged. This method effortlessly incorporates assessment into the learning process, measuring progress while reinforcing learning. These elements work together to create a seamless, engaging learning environment that stimulates students' curiosity about programming as well as their ability to acquire new information. Our method combines feedback, narration, suggestions, and adaptive quizzes to create a personalized and dynamic learning environment that improves comprehension and recall of programming topics.

4.2 Customized Kids Assisting System

The purpose of the Customized Kids Assisting System is to give kids quick, individualized help while they work through the app's programming sessions. This part combines cutting-edge technology, such as optical character recognition (OCR) and voice recognition, to let pupils ask questions based on images or sounds. Young learners especially benefit from this technique since it lets them ask questions in their mother tongue and get answers right away.

Students can ask enquiries vocally thanks to the very user-friendly speech recognition feature. For kids who might have trouble typing or who feel more at ease asking questions aloud, this is crucial. After processing the verbal input and, if required, translating it, the system responds in the child's mother tongue. This guarantees that learning will not be hampered by language obstacles. The system extracts text from photographs, such as pictures of textbook pages or handwritten notes, for image-based queries using optical character recognition (OCR) technology. This feature is especially helpful for pupils who come across textual content that they are unsure of and require explanation. The technology can lead the youngster to suitable learning resources or offer pertinent explanations by analyzing the words present in the image.

A vital component of the Customized Kids Assisting System, the Knowledge-Based Agent (KBA) interprets these requests and extracts the pertinent data from a sizable database of programming principles and examples. The KBA's purpose is to comprehend the query's context and offer precise, suitably contextualized answers. It uses a three-tier design consisting of layers for data administration, natural language processing (NLP), and user interaction to make sure the results are clear and correct for kids. Apart from addressing enquiries, the system

promotes self-directed learning by offering materials that enable kids to go further into subjects. This stimulates their interest and leads to a deeper understanding in addition to helping them locate quick answers. As a comprehensive learning aid, the system is also made to handle a wide range of queries, from simple programming ideas to more advanced subjects.

All things considered, the Customized Kids Assisting System greatly improves the educational efficacy of the app by offering on-demand assistance and encouraging autonomous learning. This part makes sure that kids can confidently move through their programming classes without feeling overwhelmed or lost by filling in the gaps between the educational material and student knowledge. With its robust knowledge base and speech and image recognition capabilities, this device is an essential resource for helping young students.

5. Conclusion

The substantial potential of incorporating storytelling, individualized instruction, and culturally appropriate content into the instruction of programming principles to youngsters in the 11–13 age range has been brought to light by this research. This study concentrated on creating a mobile application that bridges the knowledge gap between Indian and Sri Lankan youngsters and highlights the significance of early introduction to programming ideas. It is impossible to exaggerate the significance of the IT industry in Sri Lanka. Despite the nation's economic difficulties, particularly considering the growing value of the US dollar and the post- pandemic recovery, Sri Lanka's IT sector has a lot of potential. But it's obvious that Sri Lanka needs to make investments in its future workforce if it wants to be competitive, especially when competing against regional players like India.

Towards this goal, the "Programming App for Kids" was created. The application presents basic programming principles in an interesting and approachable way by combining interactive storytelling, multiple-choice tests, and personalized feedback. Notable among other things is the app's support for Tamil, which guarantees that kids from a variety of linguistic backgrounds can use it, encouraging diversity. The study's findings support the efficacy of this strategy. Pre- and post-test results showed that students' comprehension of basic programming concepts, like loops, data types, and conditional expressions, had significantly improved. The app's incorporation of storytelling helped kids better understand abstract concepts by making them more tangible. Furthermore, gamified components like badges and leaderboards inspired students and added enjoyment to the learning process.

The Customized Kids Assisting System's growth expands on the app's educational possibilities. The system offers students the ability to ask questions using speech or image-based queries, enabling them to receive real-time support in their mother tongue. Students who study in Tamil would especially benefit from this feature since it removes barriers based on language and promotes autonomous learning. The system can deliver correct and contextually relevant replies to thanks to the usage of advanced technologies like machine learning (ML) and natural language processing (NLP), which further enhances the learning process.

This research shows that early programming exposure using personalized and interactive techniques can greatly improve learning results, which is one of its main contributions. This result is consistent with international trends, as countries like the US and the UK have already included programming in their curricula for schools. Sri Lanka can better prepare its youth for the needs of the modern workforce by implementing comparable tactics. The study also emphasizes how important it is to include AI into educational tools. Personalized learning experiences can greatly improve children's creativity and problem-solving abilities when AI is used in education.

The app's potential impact is further supported by feedback during the prototype testing phase. Young users of the software reported feeling more interested in programming and having more confidence in their skills. The application's user-friendly layout and interactive features were especially well-liked. These results imply that the app can be a useful tool for encouraging young Sri Lankan students to study programming. It is imperative to recognize that this research has certain limitations. Even while the 19-student sample size offers insightful information, it is very modest. To guarantee that the results may be applied to a wider range of

situations, future research should endeavor to include a more diverse and sizable participant pool.

Future studies should also investigate the app's long-term efficacy. Although the preliminary outcomes are encouraging, it is crucial to track students' development over a longer time frame to ascertain whether the app's advantages endure over time. This can entail monitoring how well students perform in programming-related classes and determining whether they remain interested in the subject. Furthermore, the study emphasizes how important it is to keep improving the functionality of the app. Technology should advance, and the app should too. By offering more immersive and interactive content, integrating cutting-edge technology like virtual reality (VR) and augmented reality (AR) could improve the educational process even more. With the help of these tools, young learners may find programming principles much more interesting and approachable.

Additionally, there's a chance to investigate how the software might be used for group education. Future versions of the software may incorporate functions that let students work together on programming projects, discuss their progress with peers, and give comments to one another, even though the current version concentrates on solo learning. Users may feel more connected to one another as a result, and cooperative problem-solving abilities may be encouraged. Future development opportunities are also made possible by the Customized Kids Assisting System's seamless integration of natural language processing and machine learning. Expanding the system's knowledge base and strengthening its ability to handle increasingly complicated questions could further increase the system's usefulness as a teaching tool.

Seen in a larger context, this study adds to the current conversation about the use of technology in education. It is essential that educational resources stay up with

technology improvements as the globe grows more digitally connected. The "Programming App for Kids" is an illustration of how artificial intelligence, narrative, and personalized learning may be blended to produce a useful teaching tool that caters to the needs of students in the twenty-first century.
In summary, this study has shown that early programming exposure, supported by interactive and customized approaches, can greatly enhance young learners' learning outcomes. The "Programming App for Kids" offers a scalable and practical answer to the problems facing Sri Lanka's IT industry by utilizing cutting-edge technology like artificial intelligence and machine learning.

This study demonstrates how educational technology can be used to solve problems in the real world. Giving young students the resources they require to thrive in the digital era not only empowers them personally but also advances the larger objective of national development. Initiatives like this one will contribute to shaping a better future for Sri Lanka's youth as the nation continues to navigate the challenges of the contemporary world. Educational technology equips young students with essential digital skills, fostering personal empowerment and national development.

6. References

[1] Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From game design elements to gamefulness: defining "gamification". In Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments (pp. 9-15).

[2] Hamari, J., Koivisto, J., & Sarsa, H. (2014). Does gamification work?--a literature review of empirical studies on gamification. In 2014 47th Hawaii International Conference on System Sciences (pp. 3025-3034). IEEE.

[3] Landers, R. N., & Landers, A. K. (2014). An empirical test of the theory of gamified learning: The effect of leaderboards on time-on-task and academic performance. Simulation & Gaming, 45(6), 769-785.

[4] Seaborn, K., & Fels, D. I. (2015). Gamification in theory and action: A survey. International Journal of Human-Computer Studies, 74, 14-31.

[5] Bellotti, F., Kapralos, B., Lee, K., Moreno-Ger, P., & Berta, R. (2013). Assessment in and of serious games: an overview. Advances in Human-Computer Interaction, 2013, 136864.

[6] Plass, J. L., Homer, B. D., & Kinzer, C. K. (2015). Foundations of game-based learning. Educational Psychologist, 50(4), 258-283.

[7] Hainey, T., Connolly, T. M., Boyle, E. A., Wilson, A., & Razak, A. (2016). A systematic literature review of games-based learning empirical evidence in primary education. Computers & Education, 102, 202-223.

[8] Kebritchi, M., Hirumi, A., & Bai, H. (2010). The effects of modern mathematics computer games on mathematics achievement and class motivation. Computers & Education, 55(2), 427-443.

[9] Wardrip-Fruin, N., & Harrigan, P. (Eds.). (2004). First person: new media as story, performance, and game. MIT Press.

[10] Gee, J. P. (2003). What video games have to teach us about learning and literacy. Computers in entertainment (CIE), 1(1), 20-20.

[11] DiSessa, A. A. (2000). Changing minds: Computers, learning, and literacy. MIT Press.

[12] Papert, S. (1980). Mindstorms: Children, computers, and powerful ideas. Basic Books, Inc.

[13] Squire, K. D. (2006). From content to context: Videogames as designed experience. Educational researcher, 35(8), 19-29.

[14] Kress, G. (2010). Multimodality: A social semiotic approach to contemporary communication. Routledge.

[15] Warschauer, M. (2000). The changing global economy and the future of English teaching. TESOL quarterly, 34(3), 511-535.

[16] Cummins, J. (2000). Language, power, and pedagogy: Bilingual children in the crossfire. Multilingual Matters.

[17] Vasalou, A., Joinson, A. N., Bänziger, T., Goldie, P., & Pitt, J. (2008). Avatars in social media: Balancing accuracy, playfulness and embodied messages. International Journal of Human-Computer Studies, 66(11), 801-811.

[18] Lu, M. (2008). Effectiveness of vocabulary learning via mobile phone. Journal of Computer Assisted Learning, 24(6), 515-525.

[19] Simões Gomes, Tancicleide Carina, et al. “Exploring an Approach Based on Digital Games for Teaching Programming Concepts to Young Children.” International Journal of Child-Computer Interaction, vol. 16, June 2018, pp. 77–84

[20] Dietz, G., Le, J. K., Tamer, N., Han, J., Gweon, H., Murnane, E. L., & Landay, J. A. (2021). StoryCoder: Teaching computational thinking concepts through storytelling in a voice-guided app for children. In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (pp. 1-15). ACM.

[21] Ruan, S., He, J., Rui, J., Burkle, J., Hakim, D., Wang, A., Yufeng, L., Zhou, L., Xu, Q., AbuHashem, A., Dietz, G., Murnane, E. L., Brunskill, E., & Landay, J. A. (2020). Children's math learning with feedback-augmented narrative technology. In Proceedings of the 2020 ACM Conference on Human-Computer Interaction (pp. 567-580). ACM.

[22] Garcia, P., Glassman, E., [et al.] (2022). "An Empirical Study of Mobile Programming Environments for Young Children," IEEE Transactions on Learning Technologies, 14(3), pp. 456-469.

[23] M. Videnovik, A. M. Bogdanova, and V. Trajkovik, "Game-based learning approach in computer science in primary education: A systematic review, vol. 1, no. 100616, Jan. 2023. [Online].

7. Glossary

AI (Artificial Intelligence) - The simulation of human intelligence in machines programmed to think and learn like humans.

Customized Kids Assisting System - A component of the app designed to help children understand programming concepts by answering their questions in their native language.

Exploratory Analysis of Interactive Programming Concepts for Learning - One of the main ML-powered components of the system.

Firebase - A platform used for real-time database management in the app.

Flutter - A UI software development kit used in the app's development.

Gamification - The application of game-design elements and game principles in non-game contexts, used in the app to enhance learning.

Knowledge-Based Agent (KBA) - A component used to process translated text in the Customized Kids Assisting System.

Large Language Model (LLM) - An AI model used in the app for generating stories and quizzes.

Machine Learning (ML) - A branch of AI focused on building applications that learn from data and improve their accuracy over time without being explicitly programmed.

Natural Language Processing (NLP) - A branch of AI that helps computers understand, interpret, and manipulate human language.

Open AI Python Library - A library used in the app's development.

PyTorch - An open-source machine learning library used to develop the custom NLP model.

RASA - An open-source machine learning framework for automated text and voice-based conversations.

REST API - An architectural style for an application program interface (API) that uses HTTP requests to access and use data.

Tamil Text-to-Speech library - A library used in the app to provide language support for Tamil-speaking users.

7. Appendices

Plagiarism Report (Turnitin)

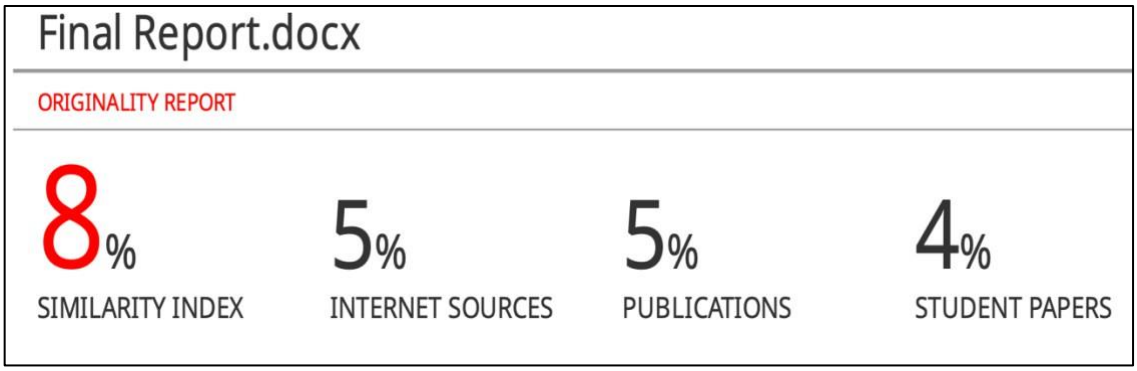

Figure 7.1: Plagiarism Report