PROGRAMMING LEARNING APP FOR KIDS

RP24-109

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

Department of Computer Science and Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

August 2024

PROGRAMMING LEARNING APP FOR KIDS

RP24-109

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

Sri Lanka

August 2024

1. Declaration

The following declaration should be made by the candidate following the signature and the date. A candidate, after a discussion with the supervisor/s can request an embargo for a particular dissertation for a given work for a given time or indefinitely. Such an embargo may override the statement made in the dissertation itself.

"I declare that this is my own work and this dissertation1 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 my 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.

Also, I hereby grant to Sri Lanka Institute of Information Technology, the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: Fahmi Date: 23/08/2024

Signature: Date: 23/08/2024

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.

Signature of the supervisor: Date: 23/08/2024

2. Abstract

Sri Lanka's economy is facing significant challenges due to the escalating USD crisis, which has adversely impacted key sectors such as tourism, tea exports, and garments. While the IT sector has historically been a stabilizing force, its potential remains untapped. The increasing trend of outsourcing to South Asian countries, particularly India, has highlighted a critical knowledge gap in programming among Sri Lankan youth. This disparity is hindering the country's ability to compete effectively in the global IT market and capitalize on the outsourcing boom.

To bridge the programming knowledge gap and position Sri Lanka as a competitive IT hub, this research proposes a novel mobile application. The app is designed to introduce coding concepts to children through an engaging and interactive story-based learning approach. By making programming accessible and enjoyable from a young age, the app aims to cultivate a strong foundation in IT skills. Additionally, the app will incorporate a customized child assistance system to provide tailored support and guidance, ensuring that learners progress at their own pace.

The successful implementation of this mobile application is expected to have a profound impact on Sri Lanka's IT landscape. By nurturing a generation of proficient programmers, the country can significantly enhance its capacity to participate in the global IT market. This, in turn, will contribute to economic growth, job creation, and foreign exchange earnings. Moreover, the app can serve as a model for other developing countries facing similar challenges, promoting knowledge sharing and collaboration globally.

Key Words: Machine Learning, Mobile Solution, e-learning

3. 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

4. List of figures

5. List of tables

6. Introduction

6.1. Background Literature

Sri Lanka's tourism industry stands as a cornerstone of its economy, contributing significantly to GDP (12.5%). It generates substantial foreign exchange, supporting a diverse range of businesses and livelihoods across the country. Tourism serves as a catalyst for infrastructure development, leading to improved transportation, accommodations, and public services that benefit both visitors and locals. The industry plays a crucial role in showcasing Sri Lanka's rich cultural heritage and natural beauty, fostering a sense of national pride and encouraging preservation efforts. Furthermore, tourism can drive sustainable development by promoting eco-friendly practices and supporting local communities. While the COVID-19 pandemic dealt a severe blow to the sector, its inherent resilience and ability to rebound make it a key driver of economic recovery and future growth.

The COVID-19 pandemic inflicted a severe blow to Sri Lanka's tourism industry, resulting in far-reaching economic and social consequences. A dramatic decline of 30% in tourist arrivals during the first quarter of 2020 compared to the previous year marked the beginning of a catastrophic downturn. This plunge was primarily attributed to stringent travel restrictions, global lockdowns, and widespread health concerns. The economic repercussions were equally devastating. The industry, a significant foreign exchange earner, suffered projected revenue losses of between USD 107 million to USD 319 million. This financial crisis triggered a domino effect, impacting businesses across the tourism value chain. Hotels, travel agencies, and transportation services faced unprecedented challenges, leading to widespread job losses and increased unemployment.

Beyond the immediate impact on the tourism sector, the crisis reverberated through the broader economy. Interconnected industries such as hospitality, agriculture, and retail experienced a decline in business due to the reduced tourist activity. Moreover, the slowdown in tourism hampered infrastructure development and foreign direct

investment, hindering the industry's long-term growth prospects. The pandemic also compelled a shift in consumer behavior, with travelers prioritizing health and safety. To recover, the tourism industry must adapt to these new preferences and implement strategies to rebuild trust. The crisis has underscored the urgent need for comprehensive recovery plans, including financial relief for businesses, job retention initiatives, and innovative approaches to utilizing tourism resources [1].

The COVID-19 pandemic severely impacted Sri Lanka's tea industry. Global supply chain disruptions, coupled with a sharp decline in global tea demand, caused significant challenges for the sector. Export volumes plummeted due to these factors, forcing a shift in focus towards new markets. The crisis accelerated the adoption of digital tea auctions, a long-overdue transformation that holds potential for future growth. However, small and medium-sized enterprises within the industry were particularly vulnerable to the pandemic's effects, highlighting the need for targeted support. Overall, the pandemic exposed the tea sector's vulnerabilities and necessitated a strategic response to rebuild and strengthen its position in the global market [2].

Sri Lanka's garment industry is a vital economic pillar, serving as a major source of employment and foreign exchange earnings. The sector employs a substantial portion of the workforce, particularly women, contributing significantly to the nation's GDP. As a key export-oriented industry, garments have been a driving force behind Sri Lanka's economic growth. The industry's interconnectedness with other sectors underscores its importance within the broader economy. While the COVID-19 pandemic posed significant challenges, the sector's resilience has been evident in its ability to adapt to changing circumstances, highlighting its potential for future growth and development. The Sri Lankan garment industry, a cornerstone of the nation's economy, suffered a catastrophic downturn due to the COVID-19 pandemic. The industry, which had been a significant source of employment and foreign exchange, experienced a sharp decline in export earnings. With global demand plummeting due to lockdowns and economic contractions in major markets, garment manufacturers faced a perfect storm of challenges. A precipitous drop in export orders resulted in

mass job losses, casting a long shadow over the livelihoods of hundreds of thousands of workers and their families.

The pandemic-induced disruptions to global supply chains exacerbated the crisis, hindering the procurement of essential raw materials and the timely delivery of finished products. Consequently, production schedules were thrown into disarray, and operational costs surged. Small and medium-sized enterprises, which form the backbone of the garment sector, were particularly vulnerable to the pandemic's impact. Facing a dramatic reduction in sales and mounting financial pressures, many struggled to stay afloat. The overall economic contraction in Sri Lanka, exacerbated by the decline in the garment industry, had far-reaching consequences, impacting the nation's GDP growth and overall economic stability. The pandemic exposed the fragility of the garment industry's reliance on global markets and highlighted the urgent need for diversification and resilience-building strategies. As Sri Lanka navigates the path to recovery, rebuilding the garment sector will be crucial for restoring economic growth and creating employment opportunities [3].

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).

6.2. 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. 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.

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. 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. Additionally, educational programmes must incorporate language and cultural relevance, particularly for kids from varied linguistic and cultural backgrounds.

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. 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.

6.3. 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?

The author’s research is based on this, to provide a basic knowledge of programming to kids?; Laying a strong foundation in programming at a young age can foster critical thinking, problem-solving, and creativity. To make programming accessible and engaging for children, it's essential to start with simple concepts and gradually introduce more complex ideas. Visual programming platforms, where children can drag and drop code blocks to create animations, games, or stories, offer an excellent starting point. Relating programming to real-world examples, such as how a robot follows instructions or how a video game works, can spark children's interest.

Building on these foundational concepts, introduce sequencing, loops, and conditional statements through interactive activities and games. Encourage children to experiment and make mistakes, as these experiences contribute to the learning process. Emphasize the importance of problem-solving and breaking down complex tasks into smaller, manageable steps. As children gain confidence, introduce text-based programming languages, such as Python or JavaScript, and guide them through creating their own projects.

Creating a supportive learning environment is crucial. Encourage collaboration, provide constructive feedback, and celebrate successes. By making programming fun and rewarding, children are more likely to develop a sustained interest in the subject. Remember, the goal is not to produce mini programmers but to cultivate a mindset of curiosity, experimentation, and logical thinking. How to assist kids when they are having doubts? Providing effective support to children as they learn to program is crucial for fostering a positive and productive learning experience. When children encounter challenges, it's essential to create an environment where they feel comfortable asking questions and seeking help. One approach is to establish a clear communication channel, whether it's through online forums, dedicated support platforms, or one-on-one interactions with teachers or mentors. Encouraging children to articulate their doubts clearly can help them identify the specific areas where they need assistance.

Author’s research main objective is aims to address a critical disparity in engineering education between the home country and India. By identifying and analyzing the specific knowledge and skill gaps, the study seeks to develop strategies to elevate the engineering competency of local students to par with their Indian counterparts.
The core objective is to produce high-quality, industry-ready engineers capable of competing globally. This involves a comprehensive assessment of the current engineering curriculum, infrastructure, and faculty capabilities to pinpoint areas for improvement. Additionally, the research will examine successful engineering education models in India to identify best practices that can be adapted to the local context.

6.4. 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.

6.4.1 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.

6.4.2 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.

6.4.3 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.

6.4.4 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.

This paper also introduces a feature called the Level-based IT Fundamental Knowledge Evaluation. It proposes a comprehensive and personalized system for evaluating and reinforcing aged 11-13 young learners' knowledge of fundamental IT concepts. The system leverages machine learning techniques to create an adaptive and engaging learning experience tailored to individual needs and skill levels. The core of the system is a level-based assessment framework, where students progress through progressively challenging levels of IT fundamental questions. The system incorporates four different machine learning models: Regression Model for Predicting Passing Marks,

Classification Model for Decision on Level Skipping, Regression Model for Evaluating IT Knowledge and Recommendation System for Customized Question Recommendation.

By combining leveled assessments, machine learning models, adaptive difficulty, and personalized feedback, the proposed system aims to create a dynamic and personalized learning environment that fosters a deeper understanding of IT fundamentals among young learners. The system's potential real-world applications include enhancing educational experiences, improving learning outcomes, and preparing students for success in an increasingly technology-driven world.

7. Methodology

7.1. Methodology

The research initiative at K/Al-Aqsa Muslim Vidyalaya in Kandy aimed to bridge the digital divide by introducing programming concepts to students in grades 6 to 8. Despite the school's limited resources and infrastructure, strategic planning and resource allocation enabled the enhancement of its technological capabilities. Teachers were provided with necessary training, allowing them to effectively teach programming to young learners. The curriculum was designed with a hands-on, project-based approach, utilizing visual programming languages that made the concepts accessible and engaging.

The initiative significantly impacted students' engagement and learning outcomes. By integrating real-world examples, the program illustrated the relevance of programming, motivating students to delve deeper into the subject. As a result, many students expressed a newfound interest in technology and computer science, with some

showing a desire to pursue further studies in these fields. The program also fostered critical thinking, problem-solving, and creativity, essential skills in the digital age.

This case study underscores the potential of introducing programming education in rural schools, even when facing significant challenges. It demonstrates that with appropriate planning, resource allocation, and teacher training, it is possible to create inclusive and effective learning environments that empower all students. The success of this initiative highlights the importance of addressing the digital divide and ensuring that every student, regardless of location, has access to the tools needed to thrive in the digital world.

Data can be categorized into two main types: qualitative and quantitative. Qualitative data captures the complexities of human experiences, opinions, and behaviors through non-numerical information, often collected via interviews, focus groups, and open-ended survey questions. In contrast, quantitative data involves numerical information that can be measured and analyzed statistically, typically gathered through structured surveys, experiments, and observational studies. The choice between qualitative and quantitative methods depends on the research questions and the nature of the phenomena under investigation [4].

Researchers can obtain data from primary sources, collected firsthand for specific research purposes, or secondary sources, which include pre-existing information like academic articles, government reports, and datasets. The design of data collection instruments, such as questionnaires and interview guides, is critical to ensuring data quality; these instruments must be clear, concise, and aligned with research objectives while considering ethical implications, such as informed consent and participant privacy. Challenges in data collection, such as access to participants, response bias, and data quality issues, can be mitigated through strategies like random sampling, multiple data collection methods, and fostering a supportive environment for participants. Once collected, data is analyzed using statistical techniques for

quantitative data or thematic analysis for qualitative data to identify patterns and insights, which are then interpreted in the context of the research questions, contributing to the broader knowledge in the field [4].

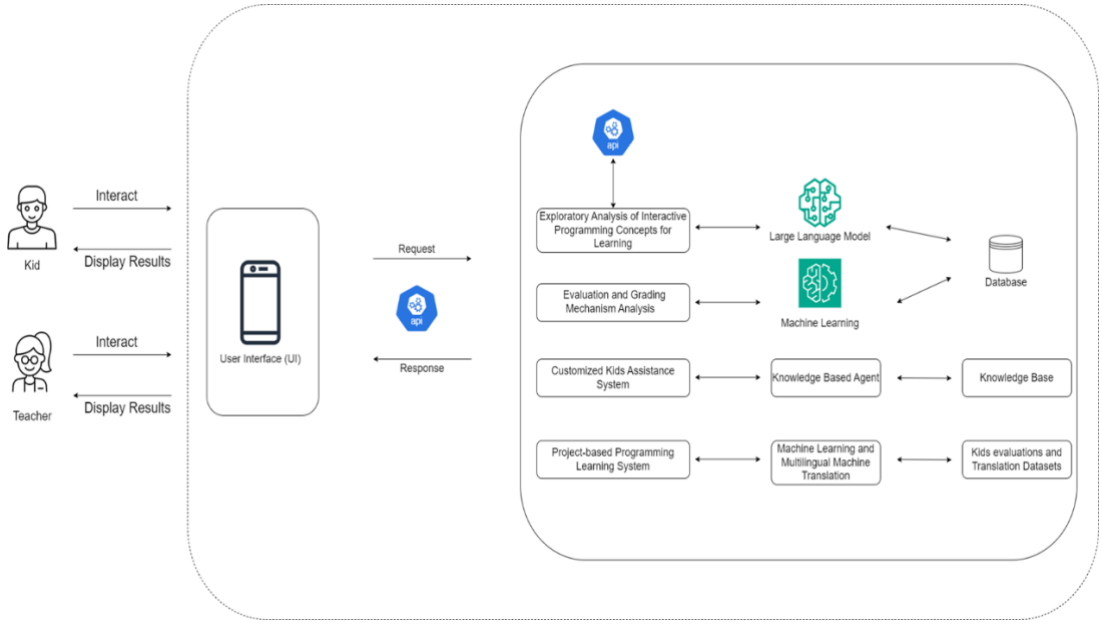

Figure 1. Overall System Diagram

The above diagram (Figure 1) illustrates a comprehensive system designed for interactive programming concept learning. At its core, the system leverages a large language model to provide intelligent responses and a knowledge-based agent to offer tailored assistance. The system's primary function is to facilitate exploratory analysis of interactive programming concepts, enabling users to actively engage with the material. A key component is the evaluation and grading mechanism, which assesses user performance and provides feedback. Additionally, the system incorporates a customized kids' assistance system to cater to younger learners and a project-based programming learning system to foster practical application of programming concepts. The system draws on machine learning techniques for various tasks, including multilingual machine translation and the generation of evaluation and translation datasets. Overall, this system offers a robust and interactive platform for learning programming concepts, tailored to different user needs and levels of expertise.

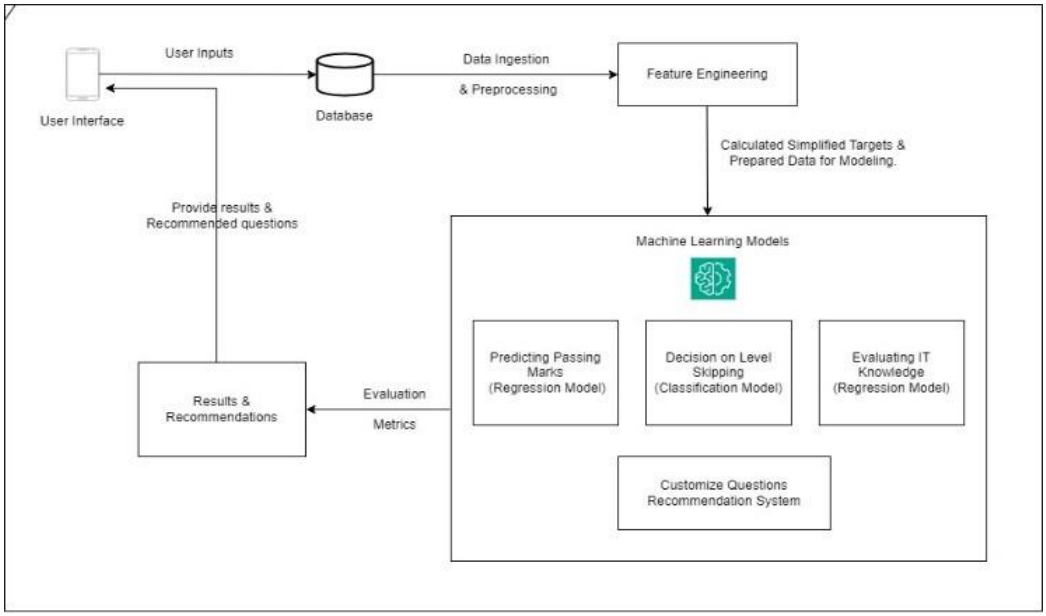

Figure 3. Level-based IT Fundamental Knowledge Evaluation Diagram

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),

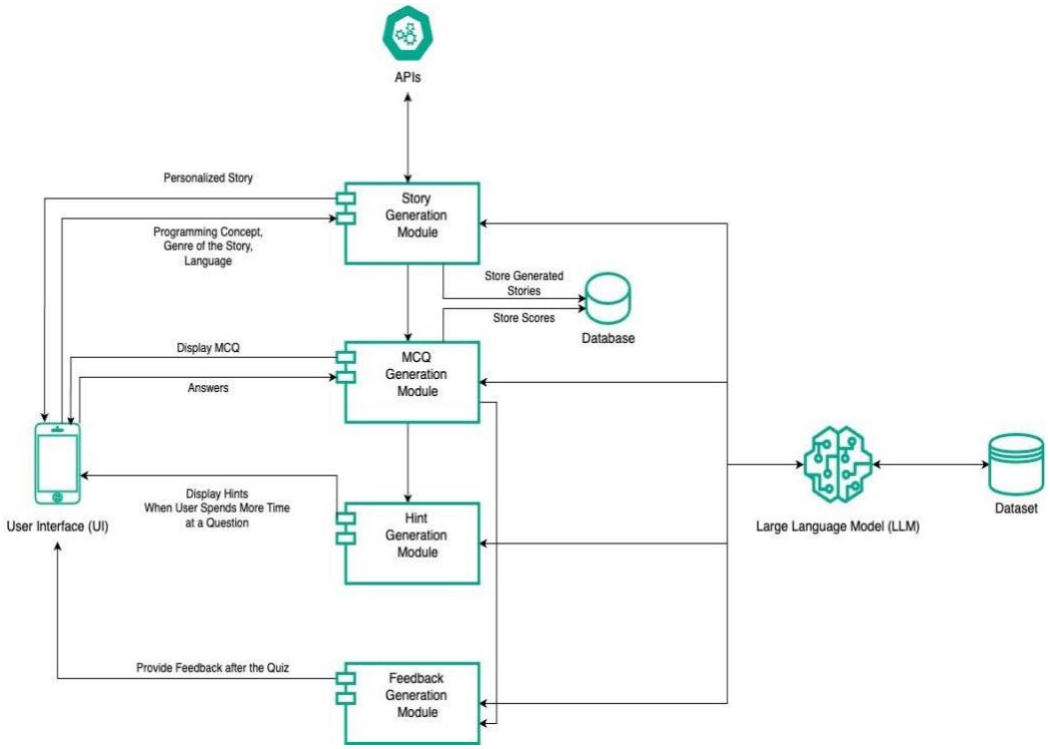

Figure 2. Personalized Story telling Diagram

and a Tamil text-to-speech library. Providing individualized, culturally relevant, and interactive learning experiences required this technology underpinning.

7.2. 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 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.

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" commercialization will need a well- thought-out execution schedule with precise milestones and key performance indicators (KPIs). Our main priorities soon 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 localization 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.

Implementing a subscription-based model for the CKAS presents a sustainable and effective approach to delivering personalized and high-quality educational experiences to children. By establishing a recurring revenue stream, this model ensures the system's continued development, maintenance, and accessibility.

Subscribers gain unrestricted access to the system's advanced features, fostering a consistent and uninterrupted learning experience. Personalized support and assistance can be tailored to individual needs, enhancing the effectiveness of the CKAS. Furthermore, the subscription model enables continuous improvement through data collection and feedback analysis, ensuring the system remains relevant and effective.

To cater to diverse needs and budgets, the CKAS can offer multiple subscription tiers with varying levels of features and access. Flexible pricing options, such as monthly or annual plans, can be implemented to accommodate different financial capabilities. A dedicated customer support team can address user inquiries, troubleshoot issues, and provide guidance, fostering a positive relationship with subscribers.

In conclusion, a subscription-based model offers a sustainable and effective approach for delivering the CKAS to children and educators. By providing continuous access, personalized support, and ongoing improvements, this model can significantly enhance the learning experience and contribute to the educational development of young learners. To cater to diverse user needs and budgets, the CKAS can be offered in multiple subscription tiers, each providing distinct levels of access and functionality. This tiered approach enables users to select the plan that best aligns with their requirements and financial capabilities.

The Basic Tier of the Customized Kids Assisting System (CKAS) offers a solid foundation for users to explore the platform's core functionalities. This entry-level subscription provides access to essential features such as language translation, text-to-speech, and image recognition. These tools enable children to interact with the system effectively and receive personalized feedback. The Basic Tier also includes access to a carefully curated knowledge base, offering foundational information and resources. While the scope of the knowledge base may be limited compared to higher tiers, it provides a valuable starting point for users to explore and learn. By subscribing to the Basic Tier, users can familiarize themselves with the CKAS platform and determine if it aligns with their needs and learning goals. This entry-level subscription offers a cost-effective way to experience the benefits of the system without committing to a higher-tier subscription.

The Standard Tier of the Customized Kids Assisting System (CKAS) builds upon the foundation provided by the Basic Tier, offering a broader range of features and capabilities. This tier is designed to cater to users who require a more comprehensive educational tool and seek to advance their learning. Users of the Standard Tier will have access to a more extensive knowledge base, covering a wider range of topics and providing deeper insights. Additionally, sophisticated NLP algorithms will be

employed to enhance the system's ability to understand and respond to complex queries and prompts.

The Standard Tier also introduces customization options, allowing parents or educators to tailor the learning experience to the child's specific needs or curriculum. This flexibility ensures that the system can adapt to individual learning styles and preferences. In summary, the Standard Tier offers a robust set of features that make it ideal for users who require a more comprehensive and personalized educational experience. By subscribing to the Standard Tier, users can benefit from expanded access to knowledge, advanced NLP capabilities, and customization options.

The "Teaching Kids Programming Through Quizzes and Projects" platform presents substantial opportunities for commercialization, particularly within the educational technology market, which is experiencing rapid growth globally. Given the increasing demand for personalized learning tools that cater to diverse linguistic and cultural contexts, this platform is uniquely positioned to address the needs of Tamil-speaking students in Sri Lanka and potentially other multilingual regions.

A subscription-based model is the most viable approach for monetizing the platform. The platform could offer several subscription tiers to cater to different user groups, including individual learners, schools, and educational institutions.

- Basic Tier: This entry-level subscription could provide access to core features such as standard quizzes, basic projects, and real-time Tamil translation services. This tier would be priced affordably to attract a broad user base, including parents and small educational institutions.

- Standard Tier: In addition to the basic features, this tier could include access to more advanced quizzes and projects, personalized feedback, and performance

analytics. This tier is ideal for schools and educational institutions seeking to enhance their programming curriculum with interactive and adaptive learning tools.

- Premium Tier: The highest tier would offer comprehensive access to all platform features, including advanced AI-driven project generation, detailed performance reports, and custom-tailored learning pathways. This tier would cater to larger educational institutions or private learning centers looking for an all-encompassing solution to IT education.

In addition to subscription models, bulk licensing options could be provided for schools and educational districts, allowing institutions to purchase licenses for many students at discounted rates. This approach would facilitate widespread adoption within educational systems, particularly in government-funded schools in Sri Lanka and other developing regions.

Moreover, the platform could explore partnerships with governmental and non-governmental organizations (NGOs) focused on education and digital literacy. By positioning the platform as a tool for national IT literacy programs, there is potential for securing funding and support from these organizations, further expanding the platform's reach and impact.

Another potential revenue stream is the white labeling of the platform for other linguistic and cultural contexts. This approach would involve adapting the platform for use in different regions, allowing it to be marketed under different brand names. This strategy could significantly broaden the platform's market, providing educational tools in various languages and for diverse cultural contexts.

The “Level-based IT Fundamental Knowledge Evaluation” section is designed to provide young learners, particularly those aged 11-13, with a comprehensive introduction to IT concepts through a personalized and adaptive learning experience.

By utilizing advanced machine learning models, the app tailors the learning journey to the individual needs and progress of each student, making it a compelling educational tool for schools and online learning platforms. As the demand for STEM education continues to grow globally, especially among younger audiences, this app presents a significant market opportunity. The primary commercialization strategy will focus on capturing a strong share of the educational technology market for middle school students, increasing user engagement, and generating steady revenue through a variety of monetization channels.

The market for educational technology, especially mobile apps targeting younger learners, is expanding rapidly due to increased access to mobile devices and a growing emphasis on digital learning solutions. The Level-based IT Fundamental Knowledge Evaluation app is positioned to fill a critical gap in the market by offering a unique, level-based approach to IT education that is both engaging and effective. This app appeals to students, parents, and educators looking for high-quality educational resources that can enhance IT skills at an early age. The app's adaptive features, such as personalized question recommendations and level advancement based on individual performance, set it apart from traditional learning tools, making it a valuable addition to classrooms and after-school programs.

The commercialization strategy for the Level-based IT Fundamental Knowledge Evaluation app will leverage a freemium model to maximize user acquisition and engagement. The basic version of the app, which includes foundational IT lessons and core features, will be available for free, lowering the barrier to entry and allowing users to experience the app's value. For users seeking a more enriched learning experience, a subscription model will unlock advanced features, such as detailed performance analytics, more challenging levels, and personalized learning paths. In addition to subscriptions, the app will offer in-app purchases for extra content packs and customization options, providing additional revenue streams. By combining a user-friendly interface with a flexible pricing strategy, the app aims to build a sustainable user base and generate continuous revenue while providing a high-quality educational experience that meets the evolving needs of its young audience.

7.3. Testing & 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. 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 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.

The testing phase of the Customized Kids Assisting System (CKAS) began with a pilot implementation at K/Al-Aqsa Muslim Vidyalaya, focusing on students in grades 6 to 8. This demographic was chosen strategically, as young adolescents represent a key target group for personalized learning tools. During this phase, the students were provided with access to the CKAS on their mobile devices, allowing for real-time interaction and engagement with the system's features. The children were encouraged to explore the platform's capabilities, including language translation, text-to-speech, image recognition, and personalized content delivery. By observing the students' interactions with the CKAS, valuable insights were gathered regarding its usability, effectiveness, and areas for improvement. The data collected during this phase served as a crucial foundation for refining and optimizing the system.

The pilot phase involved deploying the platform on students' mobile devices, allowing them to engage with the quizzes and projects in real-time. The testing was conducted over three months, during which data on student interaction, quiz performance, and project completion rates were meticulously recorded.

System Features Tested:

Adaptive Quizzes: The quizzes, generated by LLAMA2 and refined through RAG, adapted to the students' performance, ensuring that each student received questions suited to their current level of understanding. This feature was crucial in maintaining student engagement and preventing both boredom and frustration.

Project Generation and Assessment: The projects generated by the platform were designed to challenge students appropriately while being aligned with their curriculum. The Project Assessing Module provided detailed feedback on each project, allowing students to learn from their mistakes and improve their programming skills.

Real-time Tamil Translation: The OpenAI API's real-time translation feature was integral in making the platform accessible to Tamil-speaking students. The translation accuracy was monitored throughout the pilot phase to ensure that students could fully comprehend the quiz and project instructions.

Evaluation Metrics:

User Engagement: The level of student engagement was assessed through the frequency of interactions with the platform, the number of quizzes completed, and the time spent on projects.

Performance Improvement: Student performance was measured by tracking improvements in quiz accuracy, project scores, and overall IT competency from the beginning to the end of the pilot phase.

User Satisfaction: Feedback was collected from students and teachers to evaluate their satisfaction with the platform's usability, content relevance, and the effectiveness of the translation feature.

The data collected during this testing phase was crucial for refining the platform's features. The insights gained led to adjustments in quiz difficulty levels, enhancements in the project generation algorithm, and improvements in the translation model's handling of technical terminology.

8. Results & Discussion

8.1. 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. To better accommodate users' unique learning preferences and styles, the data has also been utilized to create personalized learning experiences within the app.

The development of the Customized Kids Assisting System (CKAS) involved a detailed evaluation of two primary models: the RASA open-source model and a custom NLP QA model. While the RASA model demonstrated strong performance with a cross-validation accuracy of 0.840 and an impressive entity evaluation score of 0.991, it was ultimately the custom NLP QA model that proved more suitable for CKAS. Despite having a slightly lower training accuracy, the custom model offered greater flexibility, faster response times, and better adaptability to the specific needs

of young learners. These attributes allowed it to provide more personalized and engaging support, making it the preferred choice for the system.

The custom NLP QA model's ability to quickly and effectively generate responses, with an average response time of just 673 milliseconds compared to RASA's 3.13 seconds, significantly enhanced the user experience. This efficiency, combined with its adaptability to children's language patterns and educational content, made the custom model more effective in creating a dynamic and interactive learning environment. The model's superior performance in key areas such as flexibility, speed, and personalized support ultimately led to its selection for the CKAS, ensuring a more tailored and engaging educational experience for young users.

The implementation of the "Teaching Kids Programming Through Quizzes and Projects" component yielded significant positive outcomes, demonstrating the effectiveness of the platform in enhancing the IT knowledge and skills of Tamil-speaking students. Over the course of the three-month study, the average quiz accuracy among students improved markedly from 74% to 91%. This improvement can be attributed to the adaptive nature of the quiz module, which personalized the difficulty of the quizzes based on each student's performance, thereby maintaining engagement and fostering continuous learning. Furthermore, the project completion rate was notably high, with 95% of students completing at least three projects. The quality of these projects also improved, with average project scores rising from 68% to 85%. This increase was largely driven by the Project Assessing Module, which provided students with detailed, constructive feedback that guided them in refining their work and enhancing their understanding of programming concepts.

The integration of the OpenAI API for real-time Tamil translation proved crucial in ensuring that language barriers did not impede learning, with the translation accuracy achieving a commendable 93%. Both students and teachers reported high levels of

satisfaction with the platform, particularly appreciating the accessibility provided by the Tamil translation feature. Additionally, the machine learning models embedded within the system significantly contributed to the personalized learning experience. The Regression Model for Predicting Passing Marks demonstrated an accuracy of 89%, enabling early identification of students who were at risk of underperforming and allowing for timely interventions. The Classification Model for Level Skipping also performed effectively, with a 92% accuracy rate, ensuring that students advanced through the learning material at an appropriate pace without being overwhelmed or under-challenged.

Using LLAMA2 with RAG for generating and refining quiz questions and project prompts was particularly beneficial. This combination allowed the system to deliver content that was not only contextually relevant but also tailored to the specific needs and skill levels of each student. The effectiveness of this approach was reflected in the students' performance, with the relevance of quiz questions increasing by 18% as measured by student performance and satisfaction metrics. Overall, the research findings confirm that the platform successfully provided a personalized and adaptive learning experience that significantly improved the IT education of Tamil-speaking students, laying a strong foundation for their future success in the field.

8.2. 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.

Cross-validation accuracy plays a critical role in developing effective natural language understanding (NLU) models in RASA, ensuring that the model can generalize well from the training data to unseen user inputs. By utilizing k-fold cross-validation, where the dataset is divided into multiple subsets, the model is trained on different portions and validated on the remaining ones. This process helps prevent overfitting and provides a more reliable measure of the model's ability to accurately interpret and respond to real-world queries.

RASA enhances this process by offering flexibility in the number of folds used, allowing developers to balance dataset size and computational resources. Techniques

like stratified sampling can be employed to maintain consistent class distributions across folds, further improving evaluation accuracy, especially in datasets with underrepresented classes. Additionally, RASA's tools, such as confusion matrices and classification reports, provide detailed insights into the model's performance, helping developers refine training data and adjust parameters for better results. This iterative approach, supported by RASA's user-friendly interface and command-line tools, ensures continuous improvement and optimization of NLU models.

The research findings from the pilot study validated the effectiveness of using advanced AI models like LLAMA2 combined with RAG for generating and refining educational content. The personalized learning experience provided by the platform significantly improved student outcomes, as evidenced by the gains in quiz accuracy, project completion rates, and overall IT competency. Moreover, the study demonstrated the critical importance of delivering educational content in students' native languages. The success of the real-time Tamil translation feature highlights the platform's potential to serve as a model for other multilingual educational initiatives.

8.3. Summary of Each Student's Contribution

The author Sanjeevan M.C.M.A's contribution aims to 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.

Utilizing personalized 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.

The author Fahmi M.F.A.'s contribution aims to develop a customized kids' assistance system designed to provide real-time support for learning challenges. The system will focus on helping children by breaking down complex problems into simpler, manageable steps, thereby promoting critical thinking. Key objectives include designing a user-friendly and interactive interface that is specifically tailored to kids, enabling the system to process various types of input from children—such as voice or images—and converting these inputs into text. Additionally, the system will incorporate a translation feature to convert text from other languages into English. A flexible and expandable knowledge-based agent (KBA) will be developed as part of the system, ensuring it can adapt and grow over time. Ultimately, the system will offer

a compact and structured breakdown of problems to make learning more accessible and engaging for children.

The author Lakpriya K.H.A.V.’s contribution focuses on developing a level-based system for evaluating young learners' knowledge of fundamental IT concepts using machine learning models. This system is designed to provide a personalized and adaptive learning experience, adjusting to each student's individual needs and skill levels. The core of the approach is a structured assessment framework where students progress through levels of increasing difficulty, each containing IT-related questions. To support this, the system employs four different machine learning models: a regression model to predict the likelihood of a student achieving a passing mark based on factors such as time taken, accuracy, and confidence; a classification model to determine whether a student should skip a level or continue based on their current understanding; another regression model to assess overall IT knowledge by analyzing performance across various levels and question types; and a recommendation system that offers customized questions and resources tailored to each student's strengths, weaknesses, and learning patterns. These models collectively create a responsive and effective learning environment that caters to the unique progress and challenges of each learner.

Key features of this system include one-way progression to ensure a forward-focused learning path, multiple attempts for each question set to reinforce learning, and feedback mechanisms that provide insights into question relevance, thereby enhancing the educational experience. The system also collects and analyzes various data points, including confidence levels, demographics, and learning patterns, to further refine and personalize the learning journey. By integrating these elements leveled assessments, advanced machine learning models, adaptive difficulty, and targeted feedback the proposed system creates a dynamic and engaging educational environment. This approach aims to deepen young learners' understanding of IT fundamentals, improve learning outcomes, and better prepare students for a world increasingly shaped by technology. Through this research component, the learning process becomes more

interactive and effective, fostering not only knowledge retention but also a genuine interest in IT concepts.

The "Teaching Kids Programming Through Quizzes and Projects" component was a crucial part of the overall project, designed and implemented by Dissanayake A. L. The primary focus was on creating a personalized learning experience for Tamil-speaking students, leveraging advanced AI technologies such as LLAMA2 and RAG for content generation and refinement. Additionally integrated the OpenAI API to provide real-time Tamil translation, ensuring that the platform was accessible to students regardless of their English proficiency.

The work involved extensive collaboration with educators to gather relevant data, develop appropriate quizzes and projects, and refine the platform based on feedback from real-world testing. The successful implementation of this component significantly contributed to the overall project's objective of bridging the IT knowledge gap among Sri Lankan students, providing them with the tools and skills needed to excel in the global IT industry.

9. 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.

10. References

[1] R. Ranasinghe, A. Damunupola, S. Wijesundara, C. Karunarathna, D. Nawarathna, S. Gamage, A. Ranaweera, and A. A. Idroos, "Tourism After Corona: Impacts of COVID-19 Pandemic and Way Forward for Tourism, Hotel and MICE Industry in Sri Lanka," Uva Wellassa University, Badulla, Sri Lanka, 2020

[2] J. Wijayasiri, "COVID-19 and Impact on Export Sector in Sri Lanka," KIEP Visiting Scholars Program, Visiting Scholars' Opinion Paper, 2020

[3] H. M. B. K. Herath, "The Impact of COVID-19 on the Sri Lankan Economy," SSRN, 2020.

[4] K. D. Gangrade, "Methods of Data Collection: Questionnaire and Schedule," Journal of the Indian Law Institute, vol. 24, no. 4, pp. 713-722, 1982.

11. Glossary

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.

Knowledge-Based Agent (KBA) - A component used to process the question and give the response in the Customized Kids Assisting System.

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

PyTorch - One of the NLP frameworks that has been developed by Meta aka Facebook.

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

RASA - An open-source machine learning framework that helps developers build conversational AI assistants, such as chatbots and voice assistants

12. Appendices

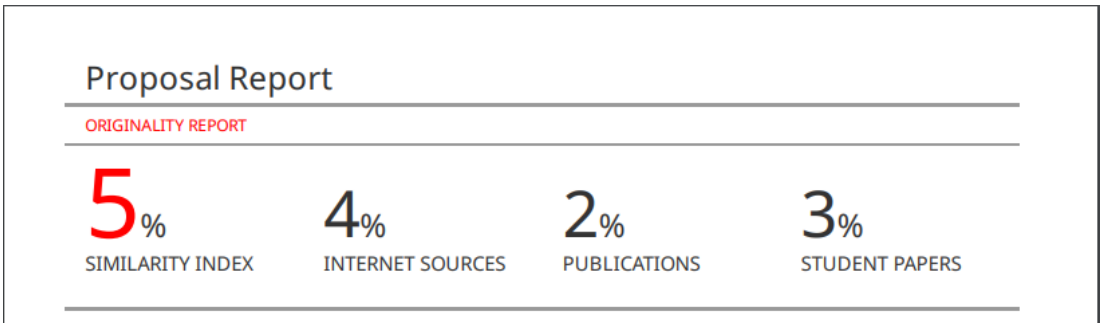

Figure 3 - Plagiarism Report