PROGRAMMING APP FOR KIDS

R24-109

Project Proposal 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

February 2024

PROGRAMMING APP FOR KIDS

R24-109

Project Proposal 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

February 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.

…… 2024/02/29

Signature of the supervisor: Date

…… 2024/02/29

Signature of the co-supervisor: Date

Abstract

This is an interactive mobile application aimed at facilitating gamified learning of fundamental programming principles among youngsters aged 8 to 10. The programme uses storytelling as its major instructional strategy and incorporates Tamil native language support to improve accessibility and engagement for young learners.

The proposed application's key components are the Story Generation Module, the Multiple-Choice Question (MCQ) Generation Module, the Hint Generation Module, and the Feedback Module. Children start by personalizing their learning experience by selecting from a choice of story topics including detective investigations or fairy tales, as well as cartoon character avatars and settings. The application then generates personalized stories in written format, supplemented by voice narration in Tamil, that incorporate the selected facts. These stories are useful for presenting essential programming concepts like loops and conditional statements in an interesting and contextually relevant manner.

Following each story, interactive quizzes based on the narrative are offered to reinforce learning results. When users experience challenges, tips are supplied to scaffold comprehension without disrupting the learning process. Upon completion of each session, children receive individualized feedback based on their performance, instilling a sense of accomplishment, and driving further improvement.

This application's development makes use of advanced technologies such as Large Language Models (LLM) and the Open AI API, in addition to the Open AI Python Library and a Tamil Text-to-Speech library. These solutions enable the seamless integration of storytelling and instructional information delivery, resulting in an effective and engaging learning experience for young users.

Keywords: Gamified Learning, Mobile Application, Storytelling, Native Language Support, Programming Concepts, Educational Technology, Interactive Learning, Children's Education.

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

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 8 and 10, 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.

1.2 Literature Survey

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).

1.2.1 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].

1.2.2 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].

1.2.3 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].

1.3 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 5-7). 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 8 and 10 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 8–10-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 in order 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 1.3: 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 young children[19].	Educational games with visuals and sounds and interactive elements with Progressively difficult levels.	This study's technique was qualitative. Individual and group interviews that were semi-structured and unstructured were used to gather data. In addition, participant observation was carried out. Field notes and	42 Students	Young children have difficulty understanding the concept of repetition. They also found that children had difficulty understanding interaction elements. (Brazil) Age (5-7) Years old	Personalization and Engagement Story Telling Language Support Adaptive Quizzes Feedback and Hint Age (8 – 10) Year-old

			audio recordings were used to record data.			
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 Personalisation 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	User-centred design (UCD) Experimental studies Case studies Surveys and questionnaires	-	Assessment methods Design methodologies Interdisciplinary approaches (integrating programming concepts with other subjects like mathematics, science, art, or language arts)	Personalisation and interactivity Multimodal learning (incorporating both visual and

		Game development platforms Web-based programming environments Mobile application development	Prototyping and iterative design Longitudinal studies Comparative studies			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	Personalised Storytelling Integration of Language and Culture Interactive learning activities Adaptive feedback and support Child-Centric Design

1.4 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 5-7 and 8-10 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 5-7 and 8-10 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?

2. Research Objectives

2.1 Main 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 individualised and interactive learning. By utilising 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. Through the use of features like leaderboards, badges, and prizes, the programme encourages engagement and reinforces learning objectives.

2.2 Specific objectives

2.2.1 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.

2.2.2 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.

2.2.3 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.

2.2.4 Feedback Generation Module

In order to give students individualised feedback depending on their performance, the Feedback Generation Module is essential. This module creates customised 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 recognising accomplishments and motivating perseverance in the face of difficulties.

3. Methodology

3.1 Overall System Architecture

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 3.1 describes the overall system diagram.

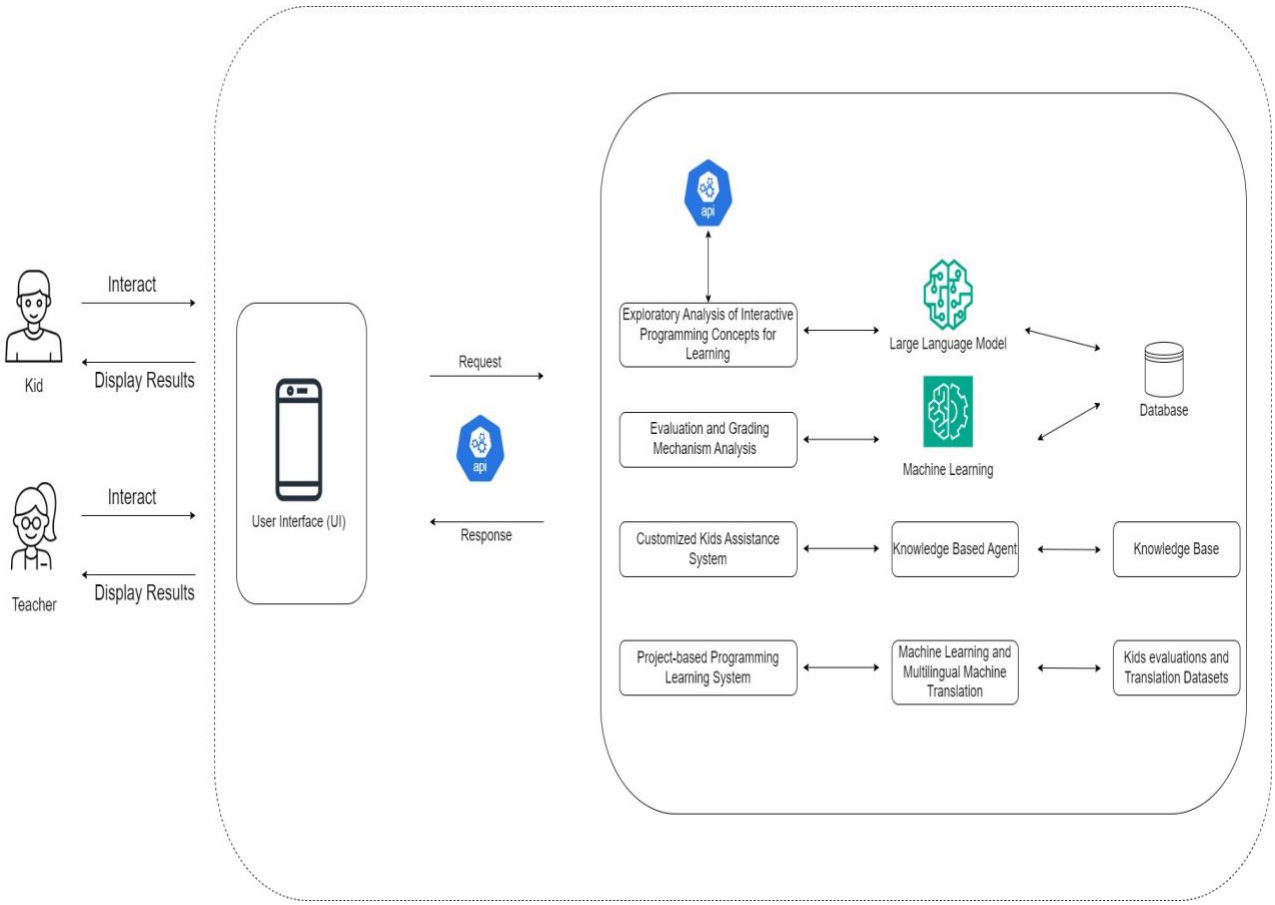


Figure 3.1: Overall System Diagram

3.2 Individual System Architecture

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 3.1 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.

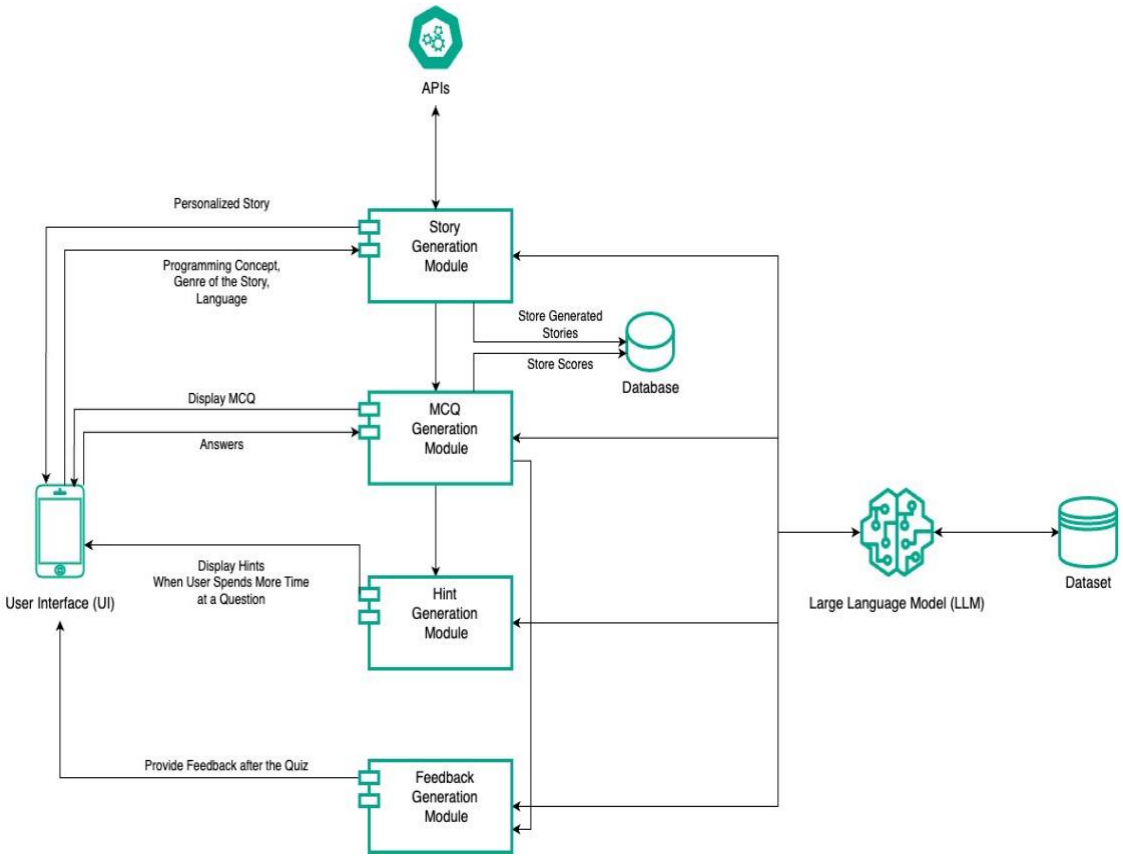


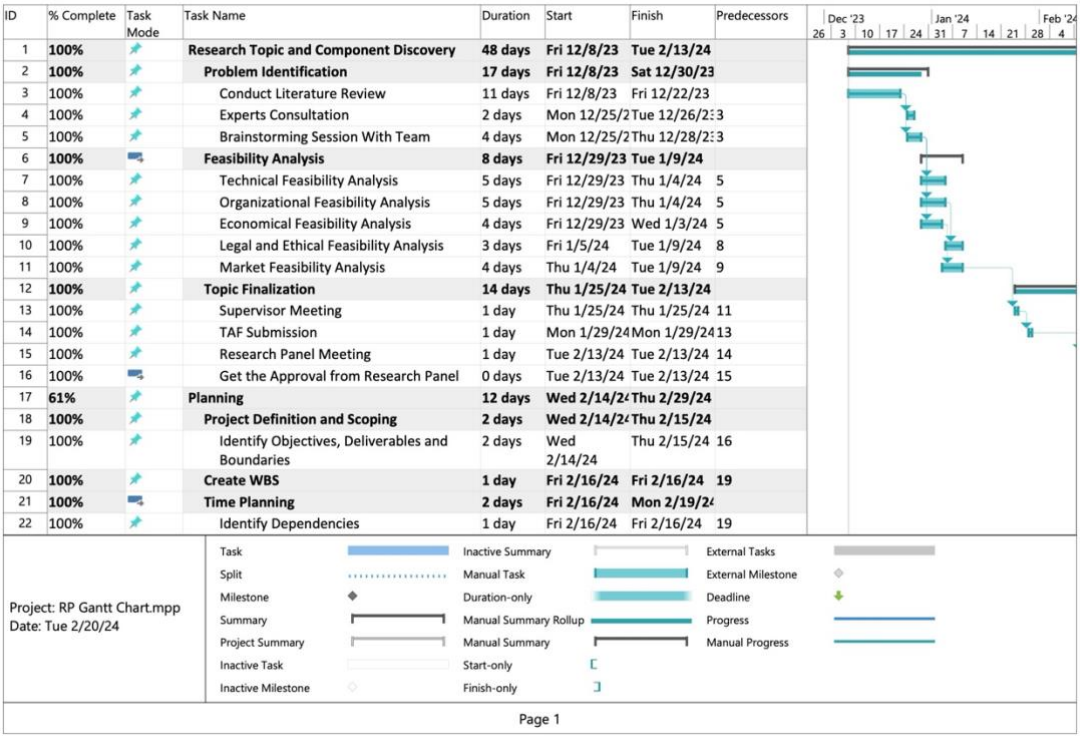
Figure 3.2: System Diagram

3.3 Project Execution Plan

The project will be carried out in phases, with each phase concentrating on particular tasks and subtasks that have been selected in order to accomplish the overall goal. Among these tasks are:

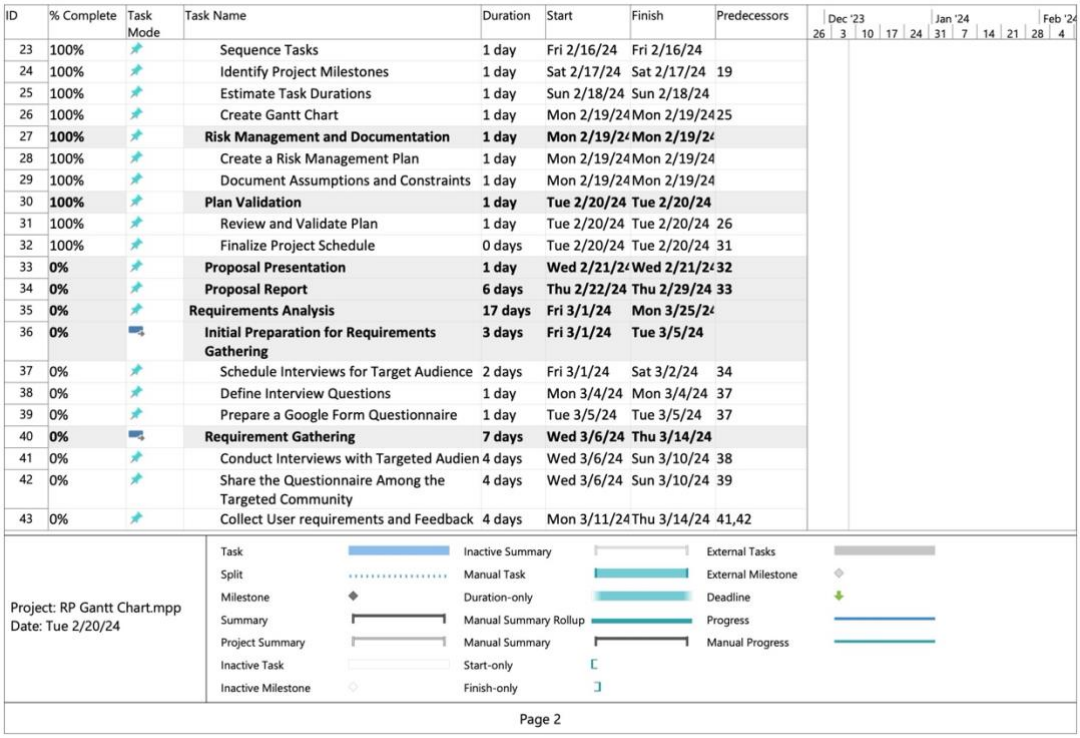
- Using Flutter to design the user interface.
- Using Python to create backend logic that generates stories, tests, answers, and feedback.
- Integrating Firebase to store and retrieve data in real-time. Putting into practice LLM and the OpenAI API for AI-driven features and natural language analysis.
- Using the Text to Speech Library to deliver Tamil audio feedback. checking and fixing the application's performance, usability, and usefulness.

A Gantt chart outlining the schedule for completing these tasks is provided in Figure 3.3.1 and Figure 3.3.2



ID	% Complete	Task Mode	Task Name	Duration	Start	Finish	Predecessors
1	**100%**		**Research Topic and Component Discovery**	**48 days**	**Fri 12/8/23**	**Tue 2/13/24**	
2	**100%**		**Problem Identification**	**17 days**	**Fri 12/8/23**	**Sat 12/30/23**	
3	100%		Conduct Literature Review	11 days	Fri 12/8/23	Fri 12/22/23	
4	100%		Experts Consultation	2 days	Mon 12/25/2	Tue 12/26/23	3
5	100%		Brainstorming Session With Team	4 days	Mon 12/25/2	Thu 12/28/23	3
6	**100%**		**Feasibility Analysis**	**8 days**	**Fri 12/29/23**	**Tue 1/9/24**	
7	100%		Technical Feasibility Analysis	5 days	Fri 12/29/23	Thu 1/4/24	5
8	100%		Organizational Feasibility Analysis	5 days	Fri 12/29/23	Thu 1/4/24	5
9	100%		Economical Feasibility Analysis	4 days	Fri 12/29/23	Wed 1/3/24	5
10	100%		Legal and Ethical Feasibility Analysis	3 days	Fri 1/5/24	Tue 1/9/24	8
11	100%		Market Feasibility Analysis	4 days	Thu 1/4/24	Tue 1/9/24	9
12	**100%**		**Topic Finalization**	**14 days**	**Thu 1/25/24**	**Tue 2/13/24**	
13	100%		Supervisor Meeting	1 day	Thu 1/25/24	Thu 1/25/24	11
14	100%		TAF Submission	1 day	Mon 1/29/24	Mon 1/29/24	13
15	100%		Research Panel Meeting	1 day	Tue 2/13/24	Tue 2/13/24	14
16	100%		Get the Approval from Research Panel	0 days	Tue 2/13/24	Tue 2/13/24	15
17	**61%**		**Planning**	**12 days**	**Wed 2/14/24**	**Thu 2/29/24**	
18	**100%**		**Project Definition and Scoping**	**2 days**	**Wed 2/14/24**	**Thu 2/15/24**	
19	100%		Identify Objectives, Deliverables and Boundaries	2 days	Wed 2/14/24	Thu 2/15/24	16
20	**100%**		**Create WBS**	**1 day**	**Fri 2/16/24**	**Fri 2/16/24**	**19**
21	**100%**		**Time Planning**	**2 days**	**Fri 2/16/24**	**Mon 2/19/24**	
22	100%		Identify Dependencies	1 day	Fri 2/16/24	Fri 2/16/24	19

Figure 3.3.1: Gannt Chart 1



ID	% Complete	Task Mode	Task Name	Duration	Start	Finish	Predecessors
23	100%		Sequence Tasks	1 day	Fri 2/16/24	Fri 2/16/24	
24	100%		Identify Project Milestones	1 day	Sat 2/17/24	Sat 2/17/24	19
25	100%		Estimate Task Durations	1 day	Sun 2/18/24	Sun 2/18/24	
26	100%		Create Gantt Chart	1 day	Mon 2/19/24	Mon 2/19/24	25
27	**100%**		**Risk Management and Documentation**	**1 day**	**Mon 2/19/24**	**Mon 2/19/24**	
28	100%		Create a Risk Management Plan	1 day	Mon 2/19/24	Mon 2/19/24	
29	100%		Document Assumptions and Constraints	1 day	Mon 2/19/24	Mon 2/19/24	
30	**100%**		**Plan Validation**	**1 day**	**Tue 2/20/24**	**Tue 2/20/24**	
31	100%		Review and Validate Plan	1 day	Tue 2/20/24	Tue 2/20/24	26
32	100%		Finalize Project Schedule	0 days	Tue 2/20/24	Tue 2/20/24	31
33	**0%**		**Proposal Presentation**	**1 day**	**Wed 2/21/24**	**Wed 2/21/24**	**32**
34	**0%**		**Proposal Report**	**6 days**	**Thu 2/22/24**	**Thu 2/29/24**	**33**
35	**0%**		**Requirements Analysis**	**17 days**	**Fri 3/1/24**	**Mon 3/25/24**	
36	**0%**		**Initial Preparation for Requirements Gathering**	**3 days**	**Fri 3/1/24**	**Tue 3/5/24**	
37	0%		Schedule Interviews for Target Audience	2 days	Fri 3/1/24	Sat 3/2/24	34
38	0%		Define Interview Questions	1 day	Mon 3/4/24	Mon 3/4/24	37
39	0%		Prepare a Google Form Questionnaire	1 day	Tue 3/5/24	Tue 3/5/24	37
40	**0%**		**Requirement Gathering**	**7 days**	**Wed 3/6/24**	**Thu 3/14/24**	
41	0%		Conduct Interviews with Targeted Audien	4 days	Wed 3/6/24	Sun 3/10/24	38
42	0%		Share the Questionnaire Among the Targeted Community	4 days	Wed 3/6/24	Sun 3/10/24	39
43	0%		Collect User requirements and Feedback	4 days	Mon 3/11/24	Thu 3/14/24	41,42

Figure 3.3.2: Gannt Chart 2

3.4 Materials and Data Collections

The following supplies are needed for the project: Flutter and Python development environments; Firebase access for database management; and an internet connection for open AI API and LLM access. The project will require data in the form of quiz questions, programming principles, and sample tales. These will be gathered from pertinent learning materials and modified for usage in the application. If target consumers' input is needed, a survey, interview, or control group will be created to obtain it. For our research, we are taking sample students from CP/K/W/Al-Aqsa Muslim Vidyalaya, Gunnepana, and there are a total of 19 students studying from grade 3 to grade 5, and we confirmed that 95% of the students' houses have touch-screen mobile facilities. So, I planned to not only give Google Forms to them to collect their data but also go directly and do some interviews to find their knowledge and understanding of this subject.

3.5 Functional and Non-functional Requirements

3.5.1 Functional Requirements

The suggested mobile application's functional requirements cover a wide range of features intended to promote efficient gamified learning of core programming principles. These consist of functions that allow for user login and registration, allowing for customized learning paths for every user. A Story Generation Module built into the application will enable users to construct interactive stories that visually present programming principles. A Hint Generation Module will aid users who run into problems, and an MCQ Generation Module will provide users quizzes to gauge their comprehension. To further improve users' learning, a Feedback Generation Module will provide customized feedback based on users' performance. Additionally, the application will give top priority to language support, guaranteeing accessibility by offering audience-specific native language support.

3.5.2 Non-Functional Requirements

non-functional requirements are essential to guaranteeing the mobile application's overall efficacy, usability, and performance. With a focus on

usability, the programme will have an easy-to-use interface that meets the needs of young learners and makes navigating simple. Performance will be a primary concern, with the programme made to run quickly, minimize loading times, and provide seamless screen transitions. Since dependability is crucial, the programme will undergo extensive testing to guarantee stability and little downtime. Strong security measures will be in place to protect user data, including secure authentication procedures and encryption. Additionally, compatibility is crucial; the programme must be designed to work flawlessly across a range of hardware, operating systems, and screen sizes.

3.6 Anticipated Conclusion

The creation of an interactive mobile application with support for native languages that successfully teaches basic programming principles through gamified storytelling is one of the project's intended outcomes. The intended user base will test the programme to assess its usability, efficacy, and practicality. The outcomes of user testing will help with any necessary alterations or enhancements to the application prior to its possible release for broader usage in educational environments.

4. Description of Personal and Facilities

4.1 Personal

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 3–5.

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.

4.2 Facilities

For the project to assist the mobile application's development and testing, access to a variety of facilities and resources will be necessary. First and foremost, every team member will require a separate development environment, which usually consists of a PC loaded with the necessary software tools for their individual duties. This includes Python interpreters for backend functionality, Firebase consoles for database management, and integrated development environments (IDEs) like Visual Studio Code for Flutter development.

Project evaluations, team meetings, and discussions will need to take place in real or virtual meeting venues. These areas help team members collaborate and communicate effectively, which guarantees agreement on project objectives, assignments, and deadlines. Maintaining momentum and addressing obstacles or problems that arise throughout the project lifetime are facilitated by regular

meetings and progress reports. For the virtual meeting, we use Teams or Google Meet. And we use Trello for allocating teamwork (refer to figure 4.2).

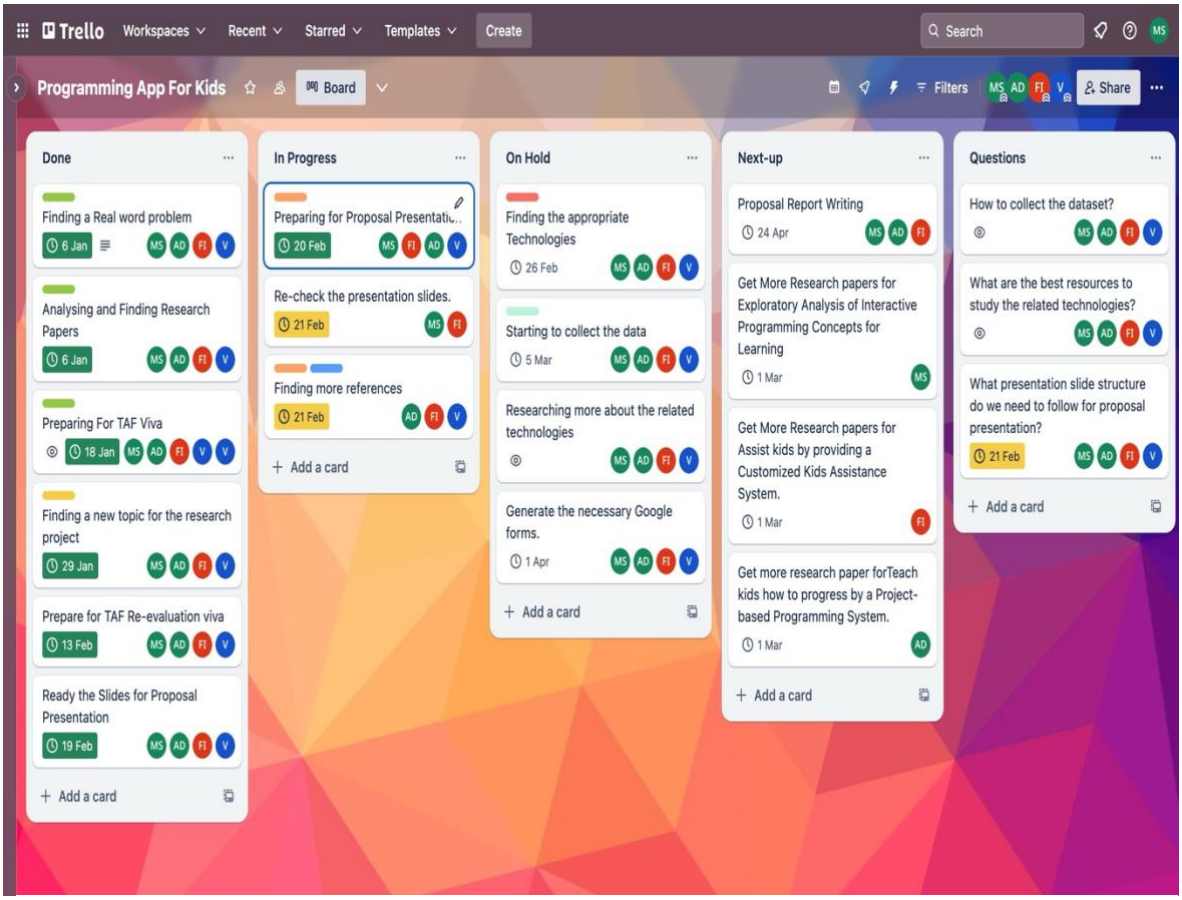


Figure 4.2: Trello

5. Budget and Budget Justification

The projected mobile application project budget is divided into several categories: staff costs, equipment and software, cloud services, professional development and training, other expenses, and a contingency reserve. The salaries of team members in charge of development, testing, and project management are covered by personnel costs. Computers and laptops, mobile devices for testing, and software licenses for development tools are examples of equipment and software costs. Firebase cloud services are crucial for hosting the backend architecture. Possibly, to improve skills, training seminars or workshops may be necessary. Miscellaneous costs include those associated with travel, communications, and other incidentals. Included is a contingency fund to cover unforeseen costs. All things considered; efficient budget management guarantees the distribution of funds required for the mobile application project's successful completion. In Table 5, an estimated cost is shown.

Table 5: Estimated Budget Table

No	Expenditure	Cost($)
1	Domain	14.43$/M
2	Hosting	2.99$/M
3	Travel Expenses	12.84$/Trip
4	Google Pay Developer Fee	25$(One Time)
Total		**55.26$**

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. Appendix

Plagiarism Report (Turnitin)