Swati Sharma Swati Sharma

Swati Sharma-capstone_report_1__Shru.docx

Quick Submit

Quick Submit

Presidency University

Document Details

Submission ID

trn:oid:::1:3430624134

Submission Date

Dec 2, 2025, 4:07 PM GMT+5:30

Download Date

Dec 2, 2025, 4:29 PM GMT+5:30

File Name

capstone_report_1__Shru.docx

File Size

2.7 MB

67 Pages

14,531 Words

83,546 Characters

*% detected as AI

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?
The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?
Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.

PRESIDENCY UNIVERSITY
Private University Estd. in Karnataka State by Act No. 41 of 2013
Itgalpura, Rajankunte, Yelahanka, Bengaluru – 560064

GNANADEEPAM: INTEGRATING MACHINE TRANSLATION AND AI FOR INCLUSIVE MULTILINGUAL LEARNING IN RURAL KERALA

A PROJECT REPORT

Submitted by

KANNIKHA H. - 20221CAI0028
RISHABH REDDY G.C. - 20221CAI0032
SHRUJANYA M. - 20221CAI0039

Under the guidance of,

Dr. SWATI SHARMA

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING, (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)

PRESIDENCY UNIVERSITY

BENGALURU

DECEMBER 2025

PRESIDENCY UNIVERSITY
Private University Estd. in Karnataka State by Act No. 41 of 2013
Itgalpura, Rajankunte, Yelahanka, Bengaluru – 560064

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

Certified that this report "GNANADEEPAM: INTEGRATING MACHINE TRANSLATION AND AI FOR INCLUSIVE MULTILINGUAL LEARNING IN RURAL KERALA" is a Bonafide work of "KANNIKHA H. (20221CAI0028), RISHABH REDDY G.C. (20221CAI0032), SHRUJANYA M. (20221CAI0039)", who have successfully carried out the project work and submitted the report for partial fulfilment of the requirements for the award of the degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING) during 2025-26.

Dr. Swati Sharma
Project Guide
PSCS
Presidency University

Ms. Suma N G
Program
Project Coordinator
PSCS
Presidency University

Dr. Sampath AK
Dr. Geetha A
School Project
Coordinators
PSCS
Presidency University

Dr. Zafar Ali Khan N
Head of Department
PSCS
Presidency University

Dr. Shakkeera L
Associate Dean
PSCS
Presidency University

Dr. Duraipandian N
Dean
PSCS & PSIS
Presidency University

Examiners

Sl. No	Name	Signature	Date
1.			
2.			

PRESIDENCY UNIVERSITY

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We, the students of final year B.Tech in COMPUTER SCIENCE AND ENGINEERING (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING) at Presidency University, Bengaluru, named KANNIKA H., RISHABH REDDY G.C., SHRUJANYA M., hereby declare that the project work titled "**GNANADEEPAM: INTEGRATING MACHINE TRANSLATION AND AI FOR INCLUSIVE MULTILINGUAL LEARNING IN RURAL KERALA**" has been independently carried out by us and submitted in partial fulfilment for the award of the degree of B.Tech in COMPUTER SCIENCE AND ENGINEERING (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING) during the academic year of 2025-26. Further, the matter embodied in the project has not been submitted previously by anybody for the award of any Degree or Diploma to any other institution.

Kannika H.	20221CAI0028
Rishabh Reddy G.C	20221CAI0032
Shrujanya M.	20221CAI0039

PLACE: BENGALURU

DATE: 03-December 2025

ACKNOWLEDGEMENT

For completing this project work, we have received the support and the guidance from many people whom we would like to mention with deep sense of gratitude and indebtedness. We extend our gratitude to our beloved **Chancellor, Pro-Vice Chancellor, and Registrar** for their support and encouragement in completion of the project.

We would like to sincerely thank my internal guide **Dr. Swati Sharma**, Professor, Presidency School of Computer Science and Engineering, Presidency University, for her moral support, motivation, timely guidance and encouragement provided to us during the
period of our project work.

We are also thankful to **Dr. Zafar Ali Khan N, Professor, Head of the Department**, **Presidency School of Computer Science and Engineering,** Presidency University, for his mentorship and encouragement.

We express our cordial thanks to **Dr. Duraipandian N**, Dean PSCS & PSIS**, Dr. Shakkeera L**, Associate Dean, Presidency School of computer Science and Engineering and the Management of Presidency University for providing the required facilities and intellectually stimulating environment that aided in the completion of my project work.

We are grateful to **Dr. Sampath A K, and Dr. Geetha A, PSCS** Project Coordinators, **Ms. Suma N G, Program Project Coordinator**, Presidency School of Computer Science and Engineering, or facilitating problem statements, coordinating reviews, monitoring progress, and providing their valuable support and guidance.

We are also grateful to Teaching and Non-Teaching staff of Presidency School of Computer Science and Engineering and also staff from other departments who have extended their valuable help and cooperation.

KANNIKHA H.

RISHABH REDDY G.C.

SHRUJANYA M.

Abstract

Rural Kerala is one of the biggest problems in the equity and quality provision of education to students. They experience resource and language disruptions as well as individualized academic support. Education system is still segregated using State, CBSE and ICSE curriculums. The educational facilities of students in rural areas are very minimal in terms of accessing educational materials that touch on the board to meet the educational requirements. They have failed in their past efforts with resources of digital platforms to resolve linguistics, technological, and mentorship issues in the education of students.

Gnanadeepam is an intervention which is brought in here to fill these gaps. Gnanadeepam is a system that integrates the syllabus of classes 8-12 of all boards and provides English and Malayalam lessons, quizzes and career-related advice [11]. Gnanadeepam uses state-of-the-art full-stack web technology (i.e., React, Supabase, Tailwind) and a multilingual voice-activated AI mentor (Rishka) to provide them with real-time tracking of academic progress, immediate quizzes and scholarship information in a low bandwidth environment with a mobile-friendly interface. Being a holistic, personalized and customized learning process, Gnanadeepam aims to empower the underprivileged students and provide them with a self-directed learning approach in their native language.

The pilot in 14 out of the 14 districts of Kerala resulted in high increases in the engagement of students, quiz completion, and digital literacy. The students demonstrated improved academic performance and had a higher level of self-confidence, and the AI mentor was particularly beneficial to non-English-speaking students. The favourable feedback of students and teachers and the objective increase in the progress analytics proves the success of this strategy. Gnanadeepam shows that with a thoughtfully planned technology, it is possible to decrease the occurrence of educational inequities and provide a decent opportunity of success to youths in Kerala and in particular to those residing in rural communities.

Table of Content

List of Figures

List of Tables

Abbreviations

Abbreviation	Description
AI	Artificial Intelligence
ML	Machine Learning
PWA	Progressive Web App
NLP	Natural Language Processing
SDG	Sustainable Development Goal
NPTEL	National Programme on Technology Enhanced Learning
SWAYAM	Study Webs of Active-Learning for Young Aspiring Minds
DIKSHA	Digital Infrastructure for Knowledge Sharing
API	Application Programming Interface
SMS	Short Message Service

Chapter 1

INTRODUCTION

1.1 Background

The state of Kerala is considered the most literate state in India but it still has the problem of translating its literacy power to equitable digital education particularly at the rural level. Rural students have a lot of obstacles they must overcome: the poor internet connection, the lack of smart devices, and the absence of local and relevant materials in the language classes or during the instruction. Although the access to basic education has been open in some government-based programs, in a bigger part, digital education remains urban in terms of setting where the needs of low-income students are neglected, as shown in Figure 1.1.

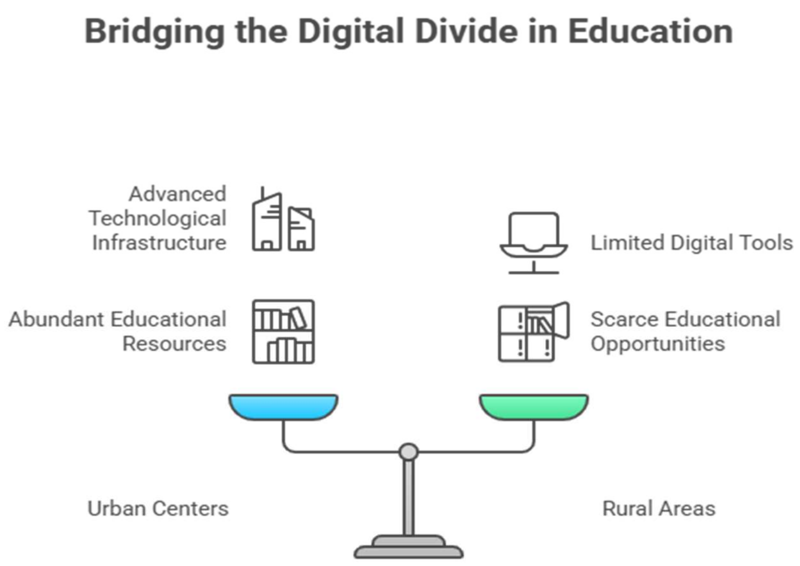

Figure 1.1: Urban and Rural regions with varying levels of digital education access

The language is also a problem because there are numerous resources mostly in English which is not the first language of most of the rural students. Even in the cases where the traditional edtech platforms provide the bilingual offering, they do not accommodate the cultural relevance of the rural learners. Consequently, limited access to digital resources causes rural students to acquire less knowledge and interact with their materials less and worse than urban students.

Gnanadeepam addresses the issues mentioned by applying AI and machine translation to provide the content in Malayalam and English. Gnanadeepam has been designed to perform effectively in low bandwidth conditions to make sure that rural students can overcome these education gaps which have been developed due to less-than-perfect infrastructure. Gnanadeepam provides interactive mentorship capabilities, customized learning plans and adaptive quizzes, in addition to personalized learning plans, to make the creation of a powerful learning ecosystem, which is both inclusive and effective, and these differences between rural and urban digital readiness are reflected in Table 1.1.

Table 1.1: Comparison between Urban and Rural Kerala (2025 data).

Region of Availability (%)	Literacy Rate (%)	Digital Education Access (%)	Device
Urban Kerala	98.2	85	90
Rural Kerala	96.5	52	60

1.2 Statistics

User Participants: In the pilot study over 1,200 students in rural districts throughout the state of Kerala were observed in the sample of students (8-12 grades), and the overall improvement and participation pattern is shown in Figure 1.2.

1.Mentoring using AI: Each of the students used the bilingual AI Mentor (there were no human mentors).

2.Interaction with the Platform Offline: The students interacting with this program were observed to work on the platform offline at a rate of about 60% indicating the ability of this program to work in low-connectivity settings.

3.Preferential Language: Approximately 75 percent of students worked with the Malayalam version persistently showing their wish of the program in a local language.

4.Academic Results: on the average, students showed a 20 percent improvement in a quiz based on science and mathematics before using the Gnanadeepam platform and after using it.

5.Increase in student engagement: Both average quizzes attempted and average number of times a student used Gnanadeepam increased by an estimated 30 per cent more as compared to both conventional online methods.

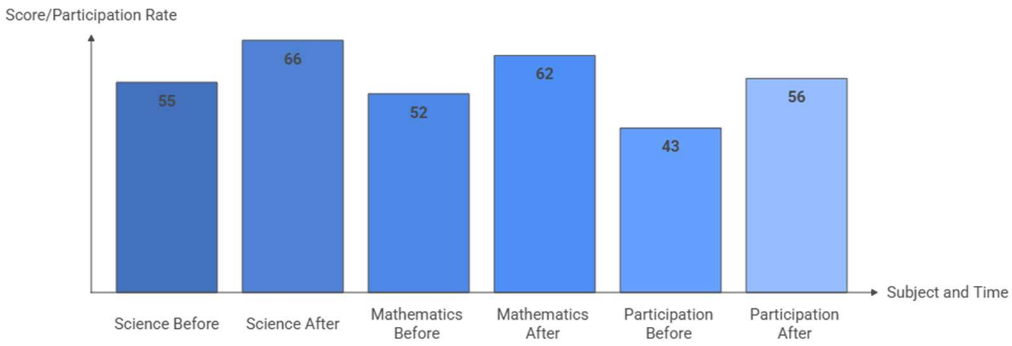

Figure 1.2: Impact of Gnanadeepam on Quiz Scores and Participation

In short, Gnanadeepam brought benefits in terms of digital access, academic performance, and student engagement among students who resided in a rural setting with limited connectivity. It was achieved on the basis of an offline-first multilingual

1.3 Prior existing technologies

Prior to Gnanadeepam, Kerala's digital education landscape had several technologies and initiatives, but none had a sufficiently contextualized and tailored solution for rural learners. A comparison of major digital learning platforms currently used in Kerala is shown in Figure 1.3.

Government Initiatives: Kerala's First Bell program generated some digital awareness, and engaged children in remote learning, by broadcasting academic content online. However, it was limited to principally one-way broadcasting of content, and a rationale for their method of broadcasting content, did not include adequate interactive learning methodologies, adaptive learning, nor multilingualism approaches.

National/Commercial EdTech Platforms: National platforms like BYJU'S, or Khan Academy, were popular and offered decent quality academic lessons digitally, but were generally urban and/or English language decisions, not really tailored to support the local syllabus or recently learn Malayalam learners.

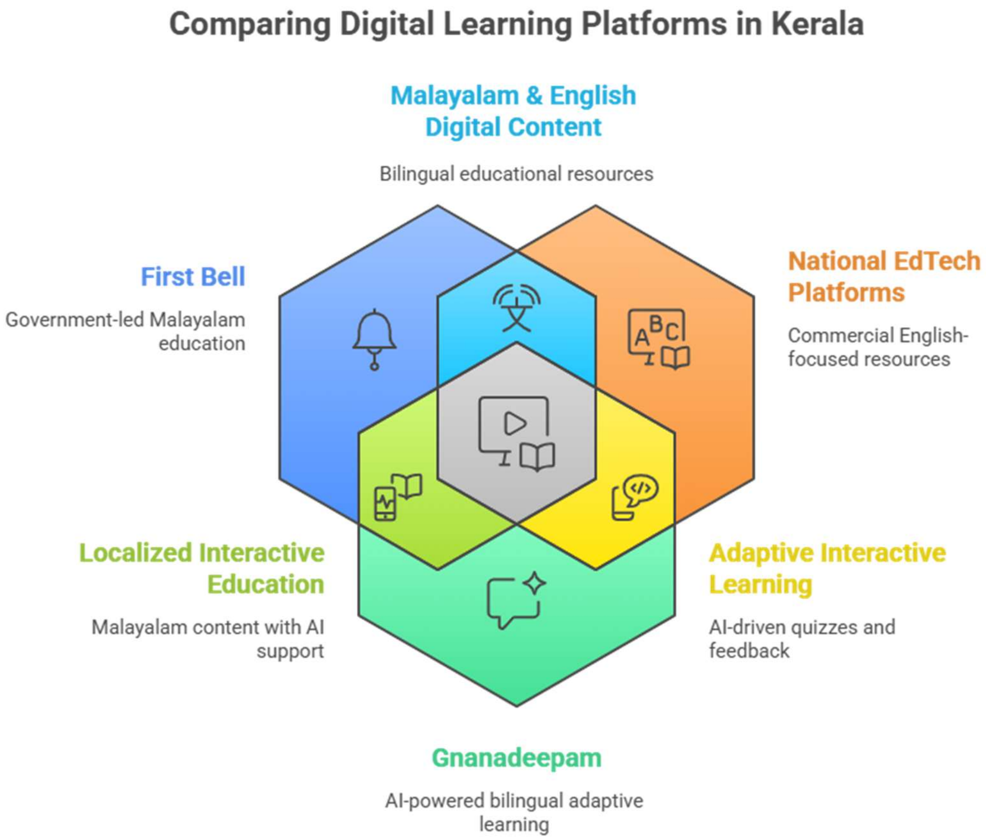

Figure 1.3. Comparison of Digital Learning Platforms in Kerala

Technology Barriers:

Most platforms wanting to increase general engagement required some continuity of stable access to the internet, which could have a barrier for rural students who may not have continuous connectivity.

Most did not personalize/adapt to individual students learning progress, culture/context, or language use.

Most did not include ways for mentors or parents to see analytics and monitor academic progress.

Most resources and strategies weren't relevant, or culturally aware, or current, or contained local content and/or couldn't address Kerala's diverse student learners.

Overall, existing technologies had considerable limitations for use in rural Kerala, blocked by systemic language, infrastructure, and contextualization. These limitations had made it even more obvious the necessity for a bilingual, culturally aware, AI augmented, and offline compatible educational solution such as Gnanadeepam to bridge the digital divide and support real educational advance to educational equity.

1.4 Proposed approach

To help Kerala's rural digital education challenges, Gnanadeepam has constructed a multi-layered, AI-supported solution, and the overall system architecture is illustrated in Figure 1.4.

- Bilingual and localized content: Educational modules are authored in both Malayalam and English based on Kerala's local syllabus and local classroom composition. Students with differing backgrounds and language capabilities can access lessons.

- AI-based personalization: The Gnanadeepam platform employs AI to offer adaptive study plans, quizzes, and feedback based on each student's progress, engagement, and performance metrics, keeping rural students actively engaged.

- Offline-first architecture: Critical modules and learning resources can be accessed in the app from the download manager without internet connection, allowing for uninterrupted education to occur.

- Integrated dashboards: Students are given individualized dashboards to monitor lessons, notifications, and skills. Teachers and parents can access analytics and performance metrics, providing timely, data-driven support.

- Interactive AI Assistant: Gnanadeepam has a bilingual, AI-based tutor and chatbot that responds to queries, explains conceptual understandings, provides adaptive feedback within the cultural context, and curriculum implementation.
- Assessment & Analytics Engine: Dynamic quizzes and lessons are provided with instant feedback from real-time learning analytics to improve retention and expose gaps in curriculum.

- Career & Scholarship Guidance: Integrated career mapping, scholarship information, and writing suggestions to support a successful application process, bridges academic achievements to employability.

- Teacher & Admin Portals: Teachers are able to assign homework, view engagement, and provide feedback, while one-to-one mentoring occurs continuously and is integrated into the system of education.

Using this approach, Gnanadeepam provides an inclusive, scalable, and sustainable digital learning ecosystem designed specifically for Kerala's rural student communities.

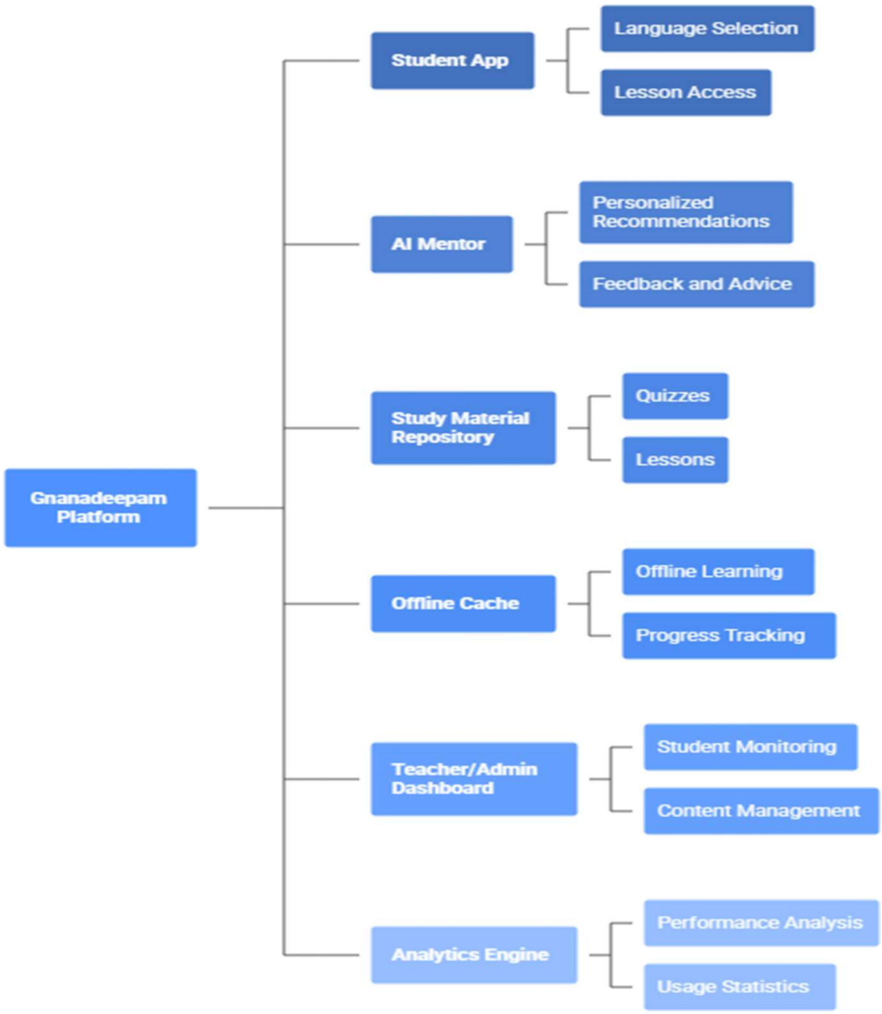

Figure 1.4. Architecture and Features of Gnanadeepam

1.5 Objectives

The project objectives are:

1. Provide educational content in two languages (Malayalam and English) that adheres to Kerala's two syllabi.
2. Inform learners of artificial intelligence-based mentorship to support personalized study plans, quizzes, and feedback.
3. Enable offline-first learning for learners in areas of low connectivity, especially rural students.
4. Offer whole student progress tracking and analytics for student, teacher, and parent access.
5. Encourage a flexible and educational learning experience based on voice assistance and adaptive quiz generation.
6. Include student career counseling and scholarship modules linking academic progress to future opportunities.
7. Elevate the role of parents and teachers as learners in support of student education.
8. Achieve equitable access to digital learning regardless of socio-economic or linguistic background.
9. Sustainably impact the sustainable development goals of quality education and reduced inequality.
10. General engagement, learner-friendly webpage that scales for web and mobile deployment.

Each objective addresses a specific challenge, ensuring a comprehensive solution

1.6 SDGs

The project aligns with global Sustainable Development Goals by improving accessibility, reducing inequality, and supporting inclusive digital learning for rural students, as shown in Figure 1.5.

- SDG 4: Quality Education: It provides bilingual, accessible, and adaptive learning materials to students in selected rural areas of Kerala, ensuring inclusive and equitable quality education, and promoting lifelong learning opportunities for all.

- SDG 10: Reduced Inequalities: The Gnanadeepam platform is designed to mitigate regional, socioeconomic, and language barriers in digital education by providing offline-first architecture and individualized support to help bridge the divide between urban and rural learners.
- SDG 8: Decent Work and Economic Growth: Career information, scholarship counselling, and resources for students are available through Gnanadeepam, which builds digital literacy, employability skills, and chances of upward mobility for youth in underserved areas.
- SDG 5: Gender Equality (indirectly): By providing inclusive content and engaging the community in product awareness and creation, Gnanadeepam enables equal access to educational and career information for girls and boys.

Fig 1.5 Sustainable development goals

1.7 Overview of project report

This report describes the design, development, and evaluation of the Gnanadeepam educational platform, designed for rural students in Kerala. The report provides context, including a rationale and an overview of the current circumstances and the context of rural education using statistics, reviewing technologies used previously to provide a background to our innovation. A detailed overview of the proposed approach is presented and the main objectives of the report and their alignment with SDGs are provided to evidence wider impact of the project.

The report has been structured as follows:

Chapter 1:

It includes the introduction, background, project statistics, technology review, proposed approach, objectives, alignment with SDGs, and outline of this report.

Chapter 2:

It is the literature review summarising literature on digital education, AI mentoring, and multilingual learning systems.

Chapter 3:

It includes the Methodology, outlining the technical and pedagogical approaches taken during development.

Chapter 4:

It includes project management, time lines, risk analysis, allocation of resources.

Chapter 5:

It includes analysis and design, system architecture, user roles, data flow and standards.

Chapter 6:

It includes implementation, software features, user interface and AI, and backend integration.

Chapter 7:

Its evaluation and results, including user feedback, performance, impact.

Chapter 8:

It includes social, legal, ethical and sustainability considerations.

Chapter 9:

It provides the conclusion and future work.

Chapter 2

LITERATURE REVIEW

Mustafa et al. (2024) identified 29 challenges within infrastructure, teacher preparedness, and policy. During a systematic review of the literature on the challenges and the strategies on the integration of the-technology in- rural schools, and proposed adaptive solutions (e.g. offline-capable tools, community trainings). This work influenced Gnanadeepam's touch on low bandwidth mobile first design of React Tailwind Supabase achieving offline quiz access and real time progress tracking in the rural area of Kerala. The projects trains teachers on AI mentor Rishka. This teacher involvement sparks a higher engagement of the students.[1]

Rajan et al (2024) conducted address the barriers of digital literacy in rural areas of the Ernakulam district and the multi-fatorial barriers such as age, gender and infrastructure and related rural inclusive hyper-digital access and design interventions of the technologically multilingual voice-controlled AI mentor Rishka to help with low and later literate users engage digitally in rural Karela and Gnanadeepam hence of the study core to the demographic of the intervention -Self Personalized Learning Pathways vacuumed the quiz completions and the digital literacy to the positive overwhelming feedback from the teacher in the first pilot of the intervention.[4]

Kumar et al. (2025) focused on ChatGPT and Duolingo’s impact on 120 engineering students from India and noted improvements in students’ motivation and enhancement of their vocabulary and grammar skills within a bilingual context of SLA (Second Language Acquisition). From these, Gnanadeepam incorporated AI personalization in English–Malayalam lessons to promote self-directed learning in cognitively passive secondary students from rural areas and to embed instant questioning and feedback mechanisms. Off-line and voice recognition features that Rishka implemented stemmed from these suggestions to promote and maintain interaction in conversational English with the secondary students from Kerala during the pilot study.[5]

Fitas et al. (2025) reviewed AI applications in early inclusive education with a focus on translating tools, real-time captioning, and assistive technologies for multilingual and other special-needs learners, utilizing primary research databases. This informed Gnanadeepam's use of Rishka's voice activated multilingual support, offering adaptive assessments and real-time intervention to eliminate academic isolation among students from rural Kerala. The focus on ethical AI and inclusivity from this paper influenced the resulting low-bandwidth features of the project, which were remarkable for the 14 districts for self-regulated learning and increased confidence.[6]

Ihichr et al. (2024) engaged with some 66 active studies in the field of adaptive assessment algorithms, theories, and techniques, while calling for the strengthening of personalized assessments for the continual improvement of educational assessments and education in general. For the generation of real-time quizzes and progression analysis, Gnanadeepam incorporated Supabase real-time data retrieved through quizzes and metrics to validate the objectives of the analytics with bilingual updates of content delivery, active models for setting educational environments and content delivery served to efficiently calibrate the encapsulated education system for low access rural and underserved populations.[7]

Zhang et al. (2024) studied AI modules including intelligent tutoring systems and edge computing for under-resourced schools, and found improvements in math/science outcomes and decreased teacher workload. Gnanadeepam built on these concepts and worked with Rishka on a rural infrastructure-challenging personalized feedback and automated tracking system using lightweight AI, which was designed for the rural areas of Kerala. The framework for the underserved context's directly aligned with the project's mobile-optimal design, which resulted in the greater engagement of students, and in the piloting of the 14 districts teacher's improved efficiency.[8]

Summary of Literatures reviewed

Sl No	Article Title, Published year, Journal name	Methods	Key features	Merits	Demerits
1	The challenges and solutions of technology integration in rural schools, 2024, Computers & Education	Systematic literature review using database searches and qualitative coding of empirical studies on rural ICT integration.	Identifies 29 distinct challenges (infrastructure, teacher readiness, policy, pedagogy) and maps them to solution strategies for rural schools.	Provides a structured framework that guided Gnanadeepam's low-bandwidth, mobile-first design and the inclusion of teacher-support mechanisms for rural Kerala.	Evidence is synthesized from diverse international contexts and not specific to Kerala; proposed solutions remain high-level and require local adaptation in Gnanadeepam.
2	Digital Information Literacy in Kerala's Rural Areas, 2024, BPAS Journals	Questionnaire-based survey with statistical analysis of rural users' digital access, skills, and usage patterns in Ernakulam district.	Analyzes how demographic variables, infrastructure, and training opportunities influence digital information literacy among rural Keralites.	Informed the design of Gnanadeepam's simple, mobile-optimized interface and the inclusion of frequent quizzes and interactions with the AI mentor to build digital skills.	Focuses on one district and library-science settings; does not directly study school-based AI platforms or long-term learning outcomes, limiting generalizability.
3	Ai-Driven Language Learning for Indian Engineering Students, 2025, International Journal of Creative Research Thoughts (IJCRT)	Survey of engineering students using AI language tools, analyzed with descriptive and inferential statistics on usage and perceived benefits.	Explores how AI tools support English language acquisition, motivation, and self-paced learning in a bilingual Indian higher-education context.	Validated the pedagogical choice to embed AI-driven personalized English support and conversational features in Gnanadeepam's Rishka mentor and language modules.	Sample consists of urban or semi-urban engineering undergraduates; findings had to be carefully adapted for younger, rural secondary students in Kerala.
4	Inclusive Education with AI: Supporting Special Needs and Tackling Language Barriers, 2025, arXiv preprint	Narrative review synthesizing literature on AI-based tools for inclusive and multilingual education, using major scholarly databases.	Describes AI applications such as real-time translation, speech technologies, and assistive interfaces for learners with language barriers and special needs.	Provided conceptual grounding for Gnanadeepam's multilingual, voice-activated mentor (Rishka) to support non-English-speaking and vulnerable students in rural Kerala.	As a preprint review, it offers limited primary empirical evidence and includes tools not yet field-tested in Indian rural school systems.
5	A Systematic Review on Assessment in Adaptive Learning, 2024, International Journal of Advanced Computer Science and Applications (IJACSA)	Systematic review protocol identifying and categorizing 66 studies on adaptive assessment techniques and models in e-learning.	Synthesizes algorithms, item-selection strategies, and design principles for adaptive assessments and real-time feedback in digital learning environments.	Informed Gnanadeepam's design of real-time quizzes, personalized question sequencing, and progress analytics to tailor learning paths for students.	Focuses mainly on technical and algorithmic aspects; offers limited discussion of bilingual delivery or low-bandwidth constraints relevant to rural Kerala.
6	Transforming Rural and Underserved Schools with AI-Powered Personalized Learning, 2024, ASM Science Journal	Conceptual and case-based analysis of AI-powered personalized learning frameworks in rural and underserved school contexts.	Proposes modular AI components (intelligent tutoring, analytics dashboards, recommendation engines) tailored to low-resource educational environments.	Strongly aligned with Gnanadeepam's holistic AI-driven architecture and supported decisions to combine personalized content, analytics, and mentoring for underprivileged students.	Empirical evaluation is limited and predominantly from Southeast Asian settings, so Gnanadeepam needed its own pilot in 14 Kerala districts to verify impact.

Figure 2.1. Summary Table of Literatures Reviewed

Chapter 3

METHODOLOGY

3.1 Introduction to Methodology and Justification of Choice

The successful development of Gnanadeepam, a sophisticated, impact-focused educational platform, requires a reliable method for development that is systematic and traceable. After careful deliberation of competing standard development approaches, such as Waterfall, Agile, Spiral, DevOps, and Onion methodologies, the primary approach chosen was the use of the V-model. This choice was based upon the fact that the V-model made a clear linkage with design to the testing/validation activity at each stage, ensuring that every requirement documented for educational problems in rural Kerala would have a strong verification validation basis in the real world. The structure and logic of the selected methodology is illustrated in Figure 3.1. The V-model would be particularly beneficial for educational platforms where implementers prioritize reliability, accessibility, and the process of validating that the platform meets user requirements. Specifically, the requirements in the specifications could be specified early, system modules could be linked to test cases, and the likelihood of missing key features- such as offline operation or multilingual options- could be reduced. The V-model makes modifying or addressing structure-bound interface modifications much easier at each stage.

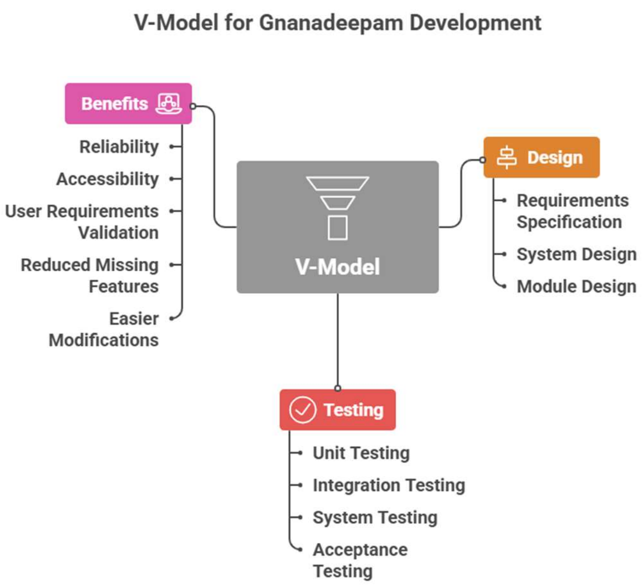

Figure 3.1. V-Model Methodology for EdTech Development

3.2 Steps of Development of Gnanadeepam Platform using V-Model

3.2.1 Requirements Specification

This step included a careful requirement engineering process, which started with literature review (for technology benchmarking), primary data collection through stakeholder interview, and survey administered in rural Keralan schools. The main requirements mapped out were developed:

- Bilingual (Malayalam and English) content delivery
- AI-driven personalized learning support
- Operability within low-bandwidth and offline use
- Teacher/mentor analytics dashboard
- Integrated modules on scholarship and career guidance
- Mobile-first responsive design

Once developed, the requirements were documented as user stories and mapped to technical epics for full traceability throughout the development process.

3.2.2 System and Functional Design

The high-level system design involved designing the primary components and the complete system architecture is shown in Figure 3.2.

- Frontend - A web and mobile dashboard for students, teachers, and parents.
- Backend - REST API, authentication, user tracking, analytics engines.
- AI and NLP - A bilingual chat bot mentor, quiz recommendations, and a translation module.
- Offline Engine - local caching and sync scripts to ensure data integrity when a connection drops.

The architecture diagram detail how each component interacts, indicates where data is stored, and what is processed in terms of security to allow a traceable link between requirements and system deliverables.

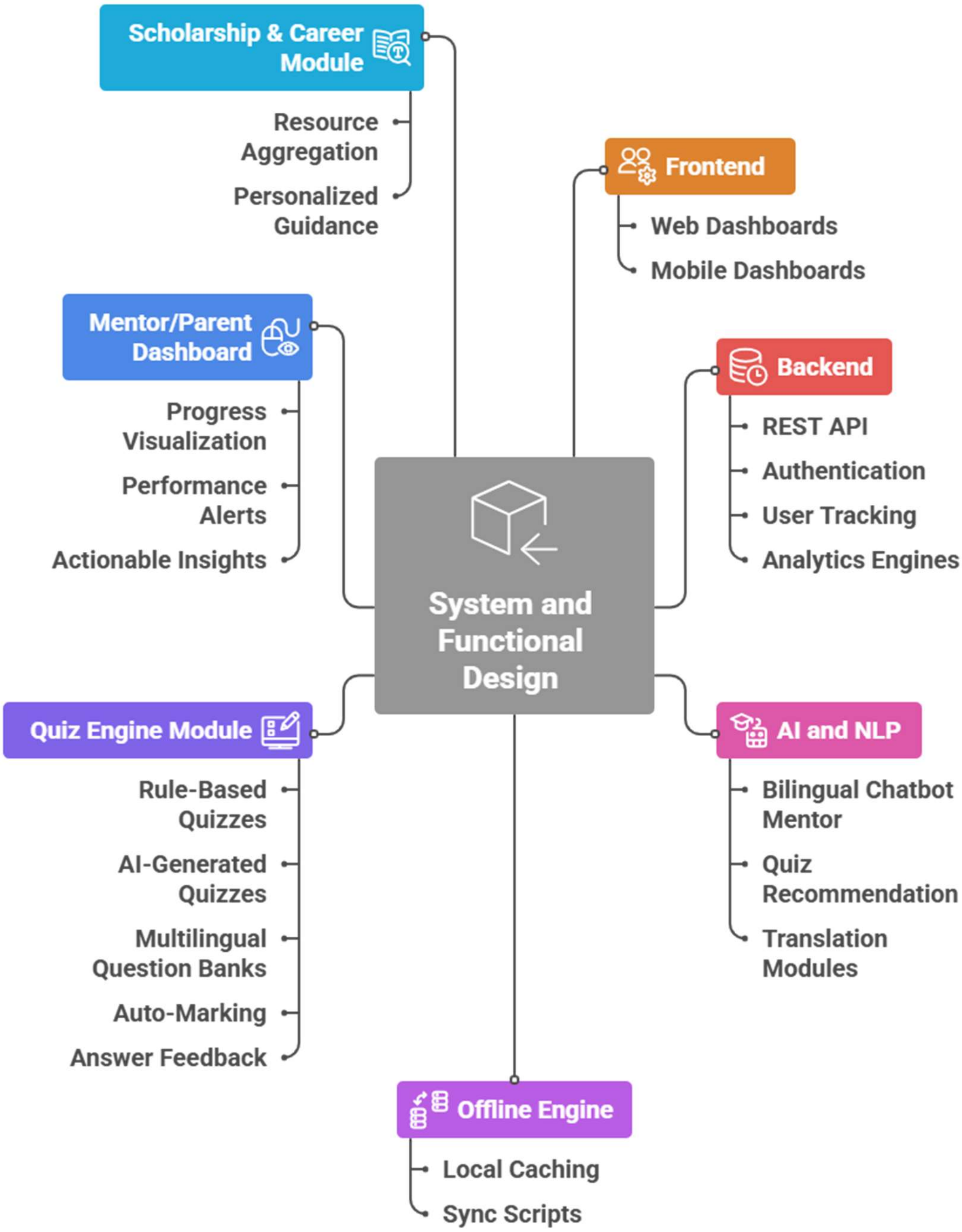

Figure 3.2. System and Functional Design Overview

The functional design was further broken down -

- Quiz Engine Module - A quiz engine that could provide quizzes based on either rules or AI generation, provides a question bank in multiple languages, and auto marks and provides feedback on answers.
- Mentor/Parent Dashboard - a dashboard that visualizes student progress, provides alerts on performance and to take actions on support for the learner.
- Scholarship & Career Module - provides resource aggregation and personalized student guidance and resources.

Each function listed testable acceptance criteria that were defined at design stage for simpler checking at the end of iterative delivery stages.

3.2.3 Unit Design

In this phase, we further broke down each high-level module into units:

- AI and Mentor: NLP pipeline, intent recognition, contextual response generation, data pre-processing in local language.
- Quiz Module: question loader, response handler, computation of score, automatic logic for adaptive retry.
- Data Sync Engine: web storage manager, handler for concurrency, conflict resolution for offline to online transition.

The hardware requirements were small and centered on smartphones and inexpensive PCs, as a mix of relying on a browser-based approach and local storage ensured accessibility for a wide range of devices. There were unit diagrams that described the interface areas of the components, as well as the operational parameters and boundaries within each unit.

3.2.4 Unit Testing

Unit tests were created for each of the units in each module. There were manual scenarios (for checking language correctness and usability) and automated tests The overall structure of the testing workflow is represented in Figure 3.3.(for edge issues e.g., lost connectivity, corrupted data packets). The tests included:

- Rendering of Malayalam/English content and switching between the two,

- User submissions that were done offline and re-synced automatically when connectivity resumed,
- Accuracy and coverage of curricular concepts for AI mentor responses.
- Adaptive scoring and error feedback in the quiz.

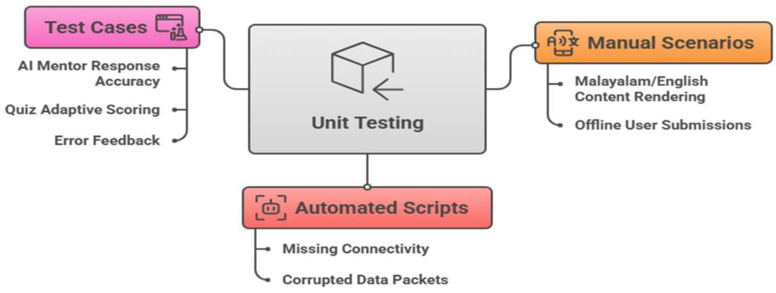

Figure 3.3. Unit Testing Coverage Flowchart

3.2.5 Integration Testing

Following calibration at the unit level, the modules were integrated into the development environment in a systematic manner. One objective was to confirm data was synchronized once the device connected after being offline. Another goal was to ensure the user session was maintained so that switching between the server and device occurred without loss of information or user privacy. When testing the system, it was important for the UI/UX flow between the student, mentor, and admin views to feel cohesive. Each user workflow also needed an immersive experience through the full switching of languages. The integration scenarios were mapped directly into the requirements log to verify full functional coverage.

3.2.6 Verification

During this phase, a requirements traceability matrix was created to align each of the specified requirements with design, implementation, and test verification. Automated tools were utilized for technical verification to confirm API responses, code developed successfully protected any data integrity, and confirmed AI mentorship logic through verification of the code. To confirm language specification functionality, human verification was used during testing. The verification logic and steps are shown in Figure 3.4. This figure represents how testing outputs were aligned with platform requirements to confirm compliance.

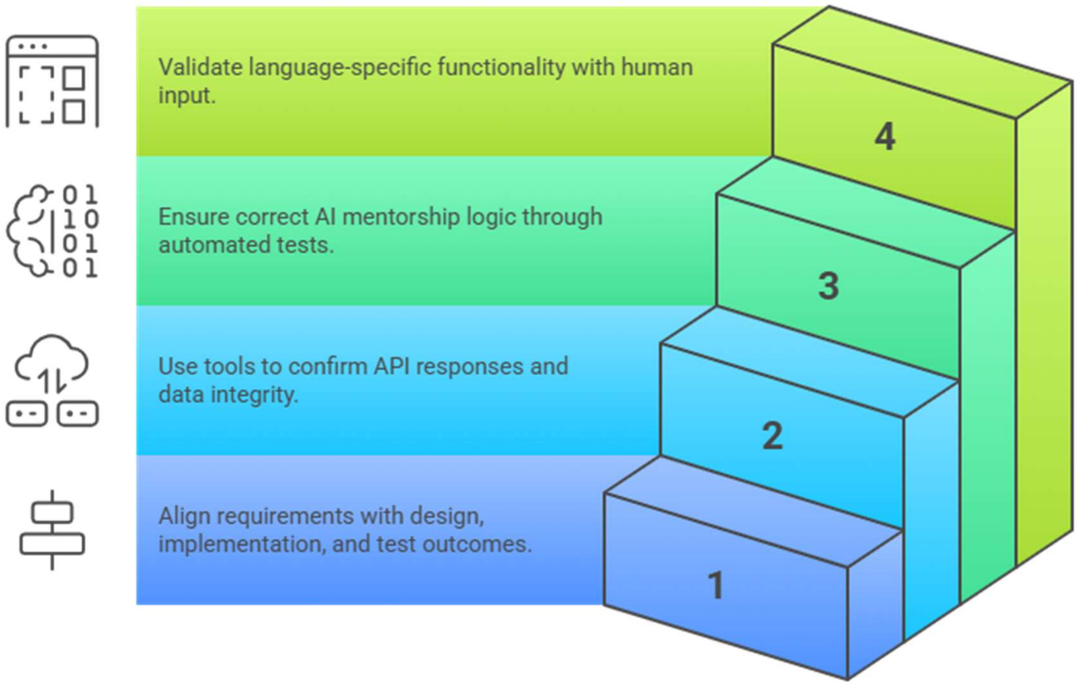

Figure 3.4. Technical Verification

3.2.7 Validation

Pilot studies were conducted with diverse end-users in rural Kerala, the real-world validation workflow is illustrated in Figure 3.5. This figure visualizes how user testing phases contributed to refinement and evaluation of the platform[2]. Which provided real-world validation. The audits included the following key metrics:

- Student engagement data and learning gains
- Teacher satisfaction and ease of access to the system
- Parental/mentor usage of the dashboard

Surveys and interviews of users, as well as further analysis of system logs were analysed for educational impact. User feedback was collected almost from the onset of each prototyping phase, and we iterated on the platform many times before finally deployed

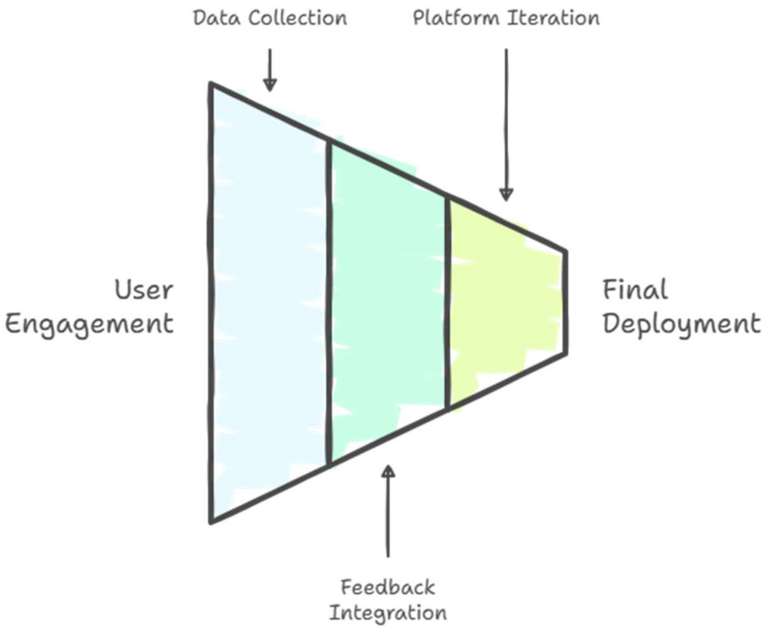

Figure 3.5. Real World Validation Process

3.2.8 Documentation, Iteration and Project Management

Documentation of the project and its provenance was comprehensive. The documentation process and project tracking approach are presented in Figure 3.6. At each stage, we kept records of architecture diagrams and technology maps, requirements log, design plan, code comments, user guides, and FAQ. The project included a Gantt chart for project management. Its usability was: planning dependency; sprint scheduling; milestone (unit tests, integration tests, user pilots) tracking; and project finalization.

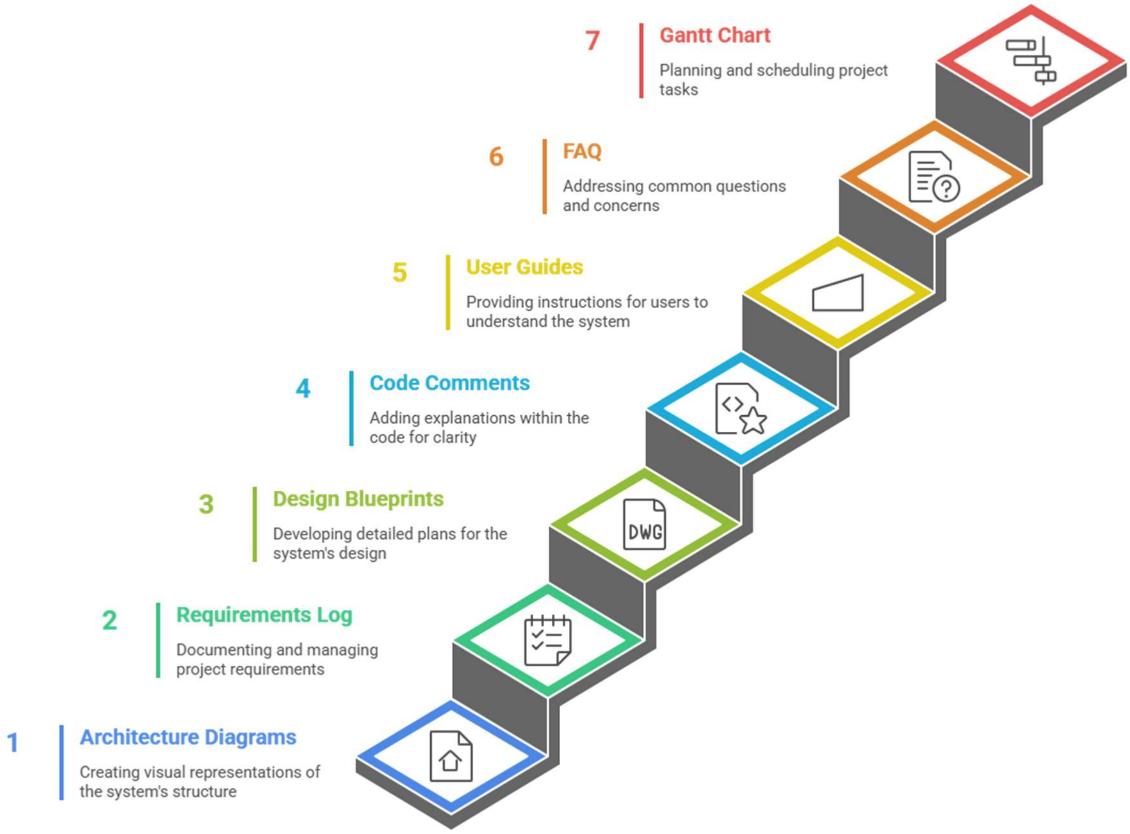

Figure 3.6. Project Documentation and Planning

Additional iterations were performed based on feedback received during testing and pilot studies. Continuous project maintenance ensures that the platform is relevant to user needs and technical feasibility. A structured review process ensured that every change was validated, documented, and version-controlled for transparency and reproducibility.

Chapter 4

PROJECT MANAGEMENT

4.1 Project Timeline

The project spans six months, from April to November 2025, divided into four phases: planning, design, implementation, and testing. A step-by-step breakdown of the timeline is shown in Table 4.1. This table shows each project phase along with duration and related activities. The visual timeline representation of these phases is shown in Figure 4.1. This figure shows all milestones and supports tracking of project progress across months.

Table 4.1. Project Timeline

Phase	Duration	Activities
Planning	Apr-May 2025	Problem analysis, literature review
Design	May-Jun 2025	System architecture, UI/UX design
Implementation	Jun-Sep 2025	Coding, API, Chatbot integration
Testing	Sep-Oct 2025	Unit, integration, system testing
Final Report	Nov 2025	Prepare Research paper and Report

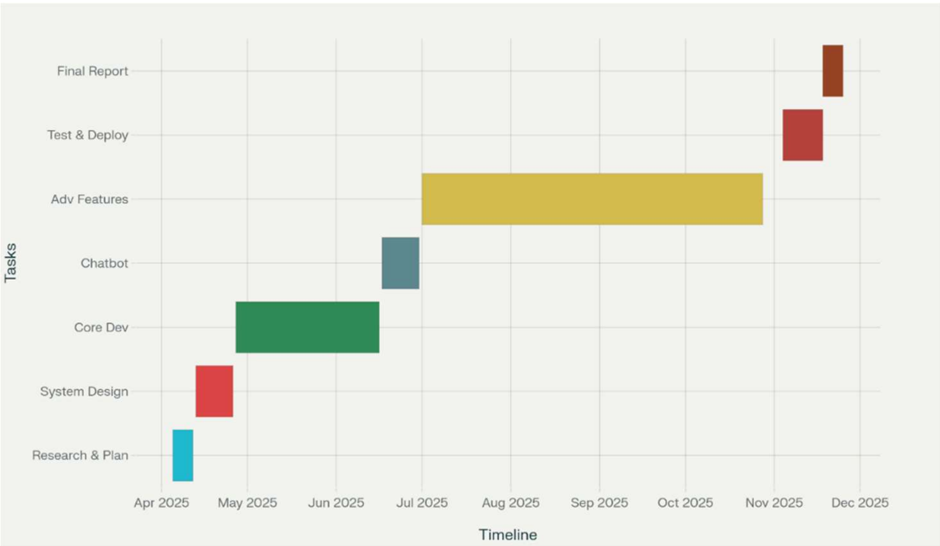

Figure 4.1. Gantt Chart

4.2 Risk analysis

Risks were analysed in accordance with PESTEL analysis method to ensure project success, and the categorized risk matrix is presented in Table 4.2. This table highlights the identified risks along with probability, impact, and mitigation strategies.

Table 4.2: Risk Analysis

Aspect	Risk	Probability	Impact	Mitigation Strategy
Political	Policy changes affecting EdTech deployment	Medium	Medium	Engage stakeholders, align with state/national programs
Economic	Funding and resource constraints	Low	Medium	Use open-source tools, leverage university partnerships
Social	Low tech-literacy, resistance to change	Medium	High	Community awareness drives, local language onboarding
Technological	Connectivity issues, API integration failure	High / Medium	High / Medium	Offline-first design, fallback mechanisms, robust QA
Legal	Data privacy breach, regulatory non-compliance	Low	High	Strong encryption, user consent, compliance with DPDPA & GDPR
Environmental	Device waste, power consumption	Low	Low	Use standard devices, optimize for energy-efficient hosting

Aspect	Risk	Probability	Impact	Mitigation Strategy
Ethical	AI bias, exclusion of marginalized groups	Medium	High	Regular audits, diverse test groups, transparent algorithms

4.3 Budget and Cost Analysis

The development of this project did not require any upfront costs, as all tools, software, and services used were from open-source licenses, or institutional availability. Gnanadeepam was built using freely accessible IDEs and tech stacks.

Furthermore, the AI and NLP components used in the system were powered using limited free credits from API service providers such as Google API or OpenAI, allowing us to use them to build our model without incurring charges. No paid subscriptions of any sort were bought or used in any phase of development.

Chapter 5

ANALYSIS AND DESIGN

5.1 Requirements

Since rural areas mainly use entry-level smartphones and basic PCs, the platform's design will be such that it can run on these devices without any need for expensive hardware to ensure the equity of all students. Monitoring dashboards for teachers and parents are a vital feature through which stakeholders can check the student progress, engagement, and performance on quizzes. It facilitates the identification of the student's learning needs in a timely manner and thus, the feedback loop between home and school gets enhanced.

The security and privacy issues arise from the foundation. Among the essential needs are a secure log-in, encrypted data storage, and role-based access controls that are aimed at safeguarding sensitive user information. Adhering to the recognized standards means that the platform will be in line with the global data protection protocols and will pave the way for user and institutional trust to be established.

The system will be designed considering the factor that the internet connection will be quite unreliable in the remote areas of Kerala. An offline-first model ensures that users can carry out lessons, take quizzes, and get reports on their learning progress in cases of a connectivity drop and the device does not automatically sync when it is back online. This keeps students' learning going without any interruption and makes it less dependent on the social realities of the local infrastructures that are in place.

Hardware, software, educational functions, and analytics as well as security requirements, each of these, in the end, determine and shape the vision of Gnanadeepam for the digital learning experiences that are inclusive, adaptive, and secure. A complete summary of these requirements is presented in Table 5.1. This table provides a structured view of the functional, technical, and operational requirements of the system.

Table 5.1. Gnanadeepam System Requirements

Requirement	Bilingual Learning Resources	AI-Powered Adaptive Learning	Entry-Level Device Compatibility	Dashboard Monitoring	Robust Security and Privacy	Offline-First Approach
Type	Software	Software	Hardware	Function	Security	Software
Rationale	Supports preferred learning mode	Ensures personalized learning journey	Allows equitable access for all	Supports timely interventions	Safeguards sensitive user information	Guarantees continuous learning

5.2 System Hardware Design Phase

The three major principles-accessibility, durability, and energy efficiency, that were used to evaluate the hardware design phase of the Gnanadeepam project, were the very first things that came to mind when the team thought about the educational usage in a rural setting. Conducting field research which involved the visits to schools and homes gave us the opportunity to get the context that helped us in deciding the kind of devices which are not only common in Kerala's schools and homes but are also easy to maintain. It is the simplest Android smartphones, entry-level Windows PCs, and low-cost tablets that have been transformed into the most efficient platforms so that any student with or without access to high-end devices can be part of the program.

We also defined the criteria for the comparison of the performance of devices of different types concerning memory (a minimum of 1GB of RAM must be available for the device to run smoothly), processor capability (multimedia delivery, modest AI tasks), and storage (offline caching for quizzes and learning materials). Connectivity through the wireless means (Wi-Fi and/or Bluetooth) was necessary for the purpose of synchronization only, and not as the main source of engagement. So, the most important consideration for the project was the offline-first design. The research on the battery/lithium-ion aspects (battery life and ease of charging/ recharging) was initiated concerning the different kinds of areas with a different degree of power supply.

The team members intended to also prove that the proposed hardware solution is not only energy-efficient but also durable and easy to repair thus ensuring that the schools with limited resources won't have to deal with the problem of long downtimes. The team has also thought of a design that would take care of the peripherals such as the microphone that a user might need for a voice-based chat session and the accessibility features. Testing in actual classrooms proved the devices' efficiency and easy operation by both teachers and students with minimum training and technology support.

The bottom line is that the system hardware design is centered on those devices that are a good mix of affordability, long-term usability, and essential performance features thus being able to contribute to the success of the project of providing advanced, adaptive education to every rural student regardless of their starting point. as illustrated in Figure 5.1. This figure highlights the selected device categories and criteria used to determine hardware suitability.

Characteristic	Android Smartphones	Entry-Level Windows PCs	Affordable Tablets
Availability & Maintenance	Widely available, easy to maintain	Available, moderate maintenance	Available, moderate maintenance
Memory (RAM)	At least 1GB	At least 1GB	At least 1GB
Processor	Supports multimedia, modest AI	Supports multimedia, modest AI	Supports multimedia, modest AI
Storage	Allows offline caching	Allows offline caching	Allows offline caching
Wireless Connectivity	Wi-Fi and Bluetooth	Wi-Fi and Bluetooth	Wi-Fi and Bluetooth
Battery Life & Charging	Evaluated for convenience	Evaluated for convenience	Evaluated for convenience
Durability & Repair	Scrutinized for minimal downtime	Scrutinized for minimal downtime	Scrutinized for minimal downtime
Peripherals	Accounts for microphones	Accounts for microphones	Accounts for microphones
Usability	Reliably used with minimal training	Reliably used with minimal training	Reliably used with minimal training

Figure 5.1. Gnanadeepam Hardware Design

5.3 System Software Design Phase

Gnanadeepam application design is influenced by the idea of providing inspiring and versatile learning, at the same time, it should be fast, secure and adaptable to the changes of the network. For this purpose, the backend development is done in Python, so it can make use of the most powerful libraries for machine learning, data processing, and natural language. This gave us a chance to have AI-based quiz recommendations, dynamic study plans, and real-time analytics modules up and running.

The user interface with React and HTML was developed to provide responsive and accessible functionality across a range of devices. The main focus was done to optimize screen size, layout and navigation for small mobile to desktops so that even young students or any non-technical user could easily manage lessons, quizzes, or chat-based activities. Since the content is bilingual, significant effort has been put into planning for localization tool integration so that switching between Malayalam and English would be faster without losing context and without any reduction in usability.

Communication between frontend and backend is managed through REST APIs to allow modularity and scalability. These decisions in architecture also allow for new features like parental dashboards, teacher analytics, and the next steps of scholarly and career counselling and the disbursement of assistants without bothering the educator and learner learning environment. The data have two storage layers, a cloud database with persistent centralized records and local storage for offline-first instances. So users can keep learning without a connection, and the data will be synced automatically once the users go back online.

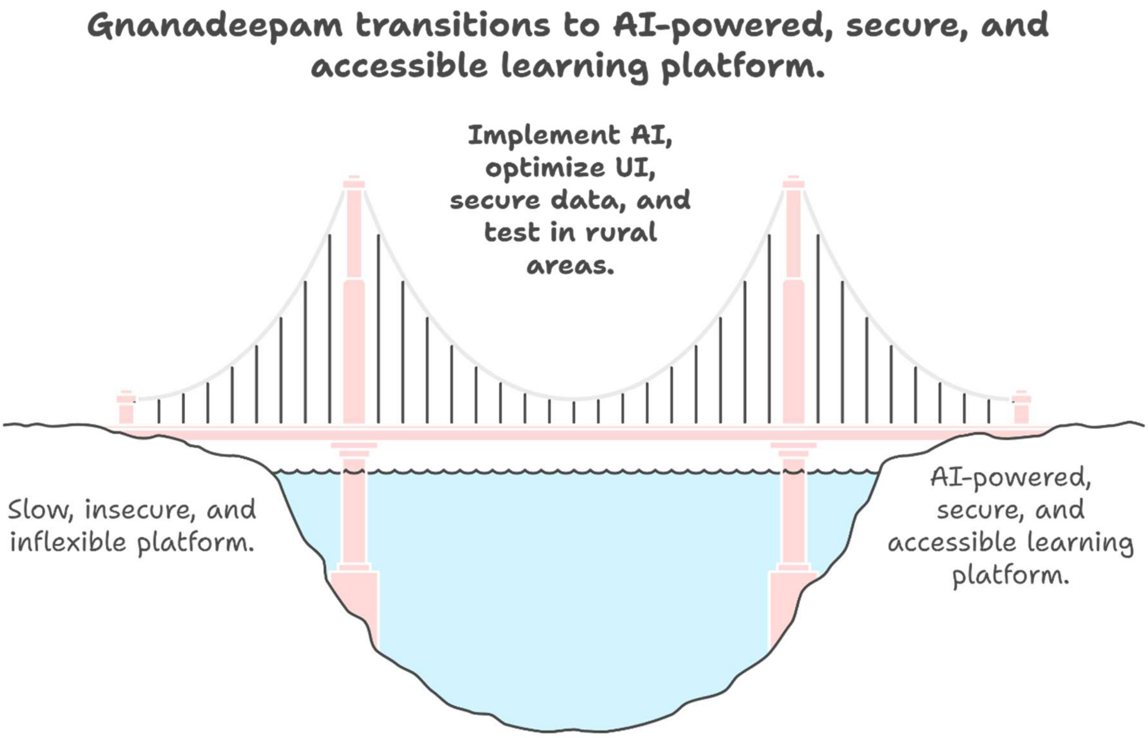

Figure 5.2. Gnanadeepam's Software Hierarchy

Security is doubled in the software layout; the technology is made with authentication, encrypted data transmissions, and validation steps to provide privacy and ensure security against attacks. Security is doubled in the software layout; authentication and encryption protect privacy and data access. The overall layout and interaction of the software architecture is represented in Figure 5.2. This figure illustrates the designed software layers with respect to the different user roles and the different modules. Ultimately, every component of the software was purposefully designed and deployed in real rural scenarios so that the entire suite of software would be able to be tailored to the user needs, facilitate learner engagement, be user-centric, and sustain the continued growth of the mergers and acquisition of the educational case applications in the Kerala districts.

5.4 Data Collection Requirements

The data collection plan of Gnanadeepam is designed to support hyper-personalized learning, system improvement, and meaningful reporting for students, teachers, and parents while considering privacy. The main focus of the platform is the secure collection of a broad range of learner and platform interactions. The registration of the users and their login timestamps are two pieces of data, among others, which not only authenticate the usage of the platform but also help in access control.

During system usage, Gnanadeepam collects records of interactions, learning from the lesson content accessed, quizzes attempted and the scores achieved. This data serves as a fuel for the AI engine for adaptive study plans as well as mastery monitoring over time. Along with quizzes and lessons, specifications are also recorded such as the time of the quiz or lesson, the number of questions answered, the breakdown of questions answered which were correct or incorrect, and the time students took on each question. The variety of data captured provides enough data to track progress for individuals and class-level analytics.

Besides data on student performance, the system also collects teacher comments and parental involvement (notes, ratings, and suggestions submitted through dashboards). This completes the feedback loop offering collaborative learning as a way of responding to community demands.

Privacy is guaranteed by our anonymizing of the most sensitive aspects (results and feedback) when data is used for analytical purposes or reporting beyond the immediate scope of users. Encryption is the method by which all data transfer and storage operations are secured, and students or parents can see what personal data is stored and have control over where it is stored.

Hence, data is recorded historically both at a detailed level (indicative recommendations and insights offered to the individual) and in the spirit of compliance and ethical practice. In the end, Gnanadeepam is able to use learning analytics in a way that is actionable, safe, and transparent to bring insights to the rural educational context.

5.5 Data Analysis Requirements

Gnanadeepam's means of educating through data is met by the AI system, which mines raw learner and system data in order to produce usable information for tailoring learning and interventions for students. The AI analytics engine, through the continual assessment of the interactions of every student, was looking for their strengths and weaknesses; specifically, the students' gaps in understanding. Besides, it prepares the analysis that pulls together this data over time and creates progress trajectories for each student, showing how they are moving through the syllabus content and adjusting to question type and/or format. The system, beyond individual analysis, pulls together class and school collective data for educators and administrators to identify trends, such as groupings of students in the same topic who are struggling to show understanding or periods of non-engagement.

Teachers in the truest sense of the word can harness this information and subsequently adjust lesson plans, differentiate resources, or plan catch-up sessions based on the real learning needs of students. There are additionally engagement metrics, such as the number of times one logs on a day, average time to complete quizzes, periods of activity/inactivity, to identify students at risk of falling behind that support intervention timelines for both parents and educators. The outputs of the data analysis are all visualized in dashboards, through progress bars, heat maps, and comparative charts, for usable feedback that is clear and actionable. The analytics also included retrospective studies for project managers and/or curriculum designers to evaluate impacts of the platform, support grant reporting, and conduct additional iterations of improvements based on collective engagement and analysis. These reports, while ensuring privacy, do so by anonymizing the data being reported across multiple students. Only those that require access can see the individual analytics.

5.6 Requirements for System Management

The Gnanadeepam platform's system management is intended to be effective, transparent, and simple to monitor in rural educational settings. Teachers and administrators can easily assign, monitor, and review lessons, tests, and student progress for both individual students and entire classes thanks to the platform's centralised dashboards. These dashboards compile vital data into actionable formats, such as completion rates, average scores, and learning patterns, allowing teachers to promptly identify students in need of extra help and to celebrate accomplishments and milestones in real time.

In this regard, Gnanadeepam's educational portal for parents efficiently facilitates tracking a child's learning journey. The system is engineered to provide timely updates on the completion of quizzes, feedback from teachers, and personalized study recommendations. It helps create an inclusive ecosystem; families stay involved in the education of their children, regardless of their own knowledge of digital systems.

It also involves administrative functions, with access control, for user management, such as enrolling students and assigning teachers and parents, troubleshooting common support issues, and deploying platform updates or curricular changes without requiring heavy technical intervention.

You can use it on your phone anything you can use it on the computer to teach students and school officials to teach classes and manage communities some of the communities are able to be taught without leaving your house with no resources.

In summary, GnanaDeepam operates as a practical, adaptive, and well-managed system, enabling educators to focus on providing high-quality instruction and assisting students in achieving academic success.

5.7 Security Requirements

For the administration of the school system, the Gnanadeepam platform provides maximum efficiency, accountability, and oversight of the educational infrastructure in rural contexts. This system provides one place for teachers and admins to assign, grade, and check homework, tests, and grades for a student or class with little work. These dashboards also include all of the information that you need like percentages of completion, average scores, and process of learning from the student, so you don ' t have to waste your time finding the students that need your help, and you could instantly see if they are progressing or succeeding.MG Parents should use the Gnanadeepam system as the best way to monitor about their children ' s performance in school. Platform created to aid you in obtaining real-time news updates regarding the completion of your quizzes, instructor recommendations, and personalized study guidance. And this is important to help to provide a positive environment where family members are involved to help students with homework, no matter if they have technology knowledge. Principals if they do have the access as a user they could do like add user like register a student add a teacher add a parent and fix simple problem or they could change the platform or curriculum without contact tech support, It ' s mobile and desk top friendly so school and community coordinators can be assured to handle broad areas with small amount resources[3]. In the end, it makes sense to say that this system is made to help the Gnanadeepam platform run well. It can grow as needed and is easy to run. This lets teachers spend more time on teaching and helping students learn in a good way. The system does the hard work, so teachers can teach and students can learn better.

5.8 User Interface Requirements

This site was designed to be a resource for children, parents, and teachers in the rural areas of Kerala and that is why this site was created for the people in the community in that way, if you can change to Malayalam to English any time, that will be very useful for all of us, and the classes, quizzes and chat it will be exist to both culture group without use any translation apps.Design changes used easy-to-read letters, strong colour blends, and clear signs. These help people who know little about tech or have poor sight. You know how it is so flexible whatever you’re doing on your phone or you’re on your computer it will look fine on any size screen you are in school or at home or even on your car. It’s so important in a country side because you use most of the time old device sharing with other people. Furthermore, the system supports voice-activated communication, allowing users to verbally interact with students or obtain auditory feedback. And the voice is very good for the kids and the people that not know how to read. It ' easy and user friendly because it will just let the user know what to do next so it might be confuse the first-time user. User-Friendly Dashboards were created with intuitive widgets and tracking functionalities enabling educators and parents to monitor individual learner progress, quiz completion status, and review feedback provided in annotations with ease. Also, messages and pop-ups are built into the site to help people stay up to date on things that matter. These can tell them about when work must be done, if they miss a day, or when there is a new group to join. This helps users know what is going on so they can act fast and not wait. For example, during the pilot launch phase, we provided end users with full access to the entire user interface, incorporating their input into subsequent releases to improve usability and meet end-user expectations effectively. This for the student that are in the country side of the world so online school is a great way to do it, cheap and reliable.

5.9 Functional Block Diagram

Figure 5.3 depicts the schematic block diagram of the Gnanadeepam system. Input blocks on extreme left of diagram denote system entry points (student, teacher, parent) via mobile or desktop. This info goes through the platform's smart system, made up of these main parts:

- The AI Mentor: It gives users custom plans for study and tests.
- The Quiz Engine: It handles picking questions, running quizzes, and looking at the scores.
- The Analytics and Progress Tracker: It keeps track of what users do and how well they do.
- The Offline Sync Module: It helps save data and lets users get their school work even when they are not online.

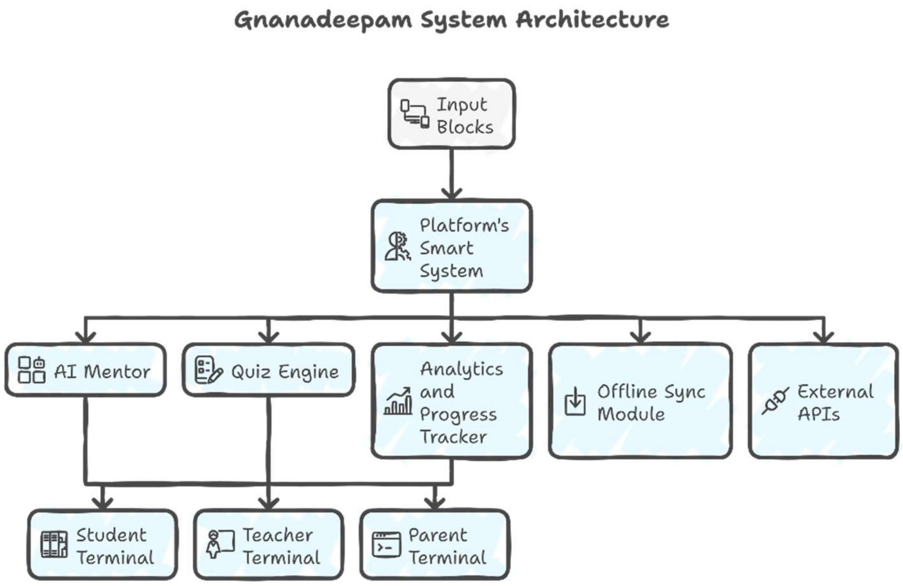

Figure 5.3. Gnanadeepam User Journey

This node forwards data streams to terminal units. Students are provided with a study schedule instant quiz feedback and scores teachers receive analytics on the class or individual students who are performing well and therefore teachers are aware of which students require additional assistance gran parents are provided with a summary and notifications of how the students are doing. Bidirectional arrows between sync and all modules imply both context and new content are continually updated online and offline. External access to standards aligned APIs enabling future scalability for career guidance tools and educational content. This drawing shows the build of Gnanadeepam. It is a plan of the parts that fit together and how they work as a group. You can see the base parts, like tools for making stuff your way, teacher join-up, parent space, and the tech parts behind it all.

5.10 System Flow Chart

The System Flowchart of Gnanadeepam demonstrates the interaction points with the user, starting from the initial entry point to when the user fully engages with the system and concludes their interaction. The detailed process logic and conditional handling are illustrated in Figure 5.4. This diagram explains how users shift between system states, authentication, learning tasks, and offline sync. The streamline benefits both developers, teachers, and future users of the platform by providing a summary of the system that is easy to understand. When a student, teacher, or parent enters the app, the application initiates a verification process, and subsequently, the system accesses the profile with the required security. Next time, I will select one option (Malayalam or English) based on which language I want to work on my dashboard and my study material, which will be restructured accordingly. Then, once you are exploring the main menu, there will be a work schedule from the A. I. and notification summaries that will let the students know what is going on at the school and the learning application. For example, if a user is trying to start a lesson or a quiz then the program pulls up questions and it will change the difficulty based on how well they did on the last quiz or lessons and its actually running the quiz modules. They get to see your answers immediately and also helps and guides you if your wrong or its not finished they will help you find the right answer yourself. Once the evaluation is finalized, the performance and learning metrics are compiled and instantly displayed on the respective dashboards of the student, instructor, and guardian, who are interconnected via linked accounts. Furthermore, educators can develop unique activities for tracking and can access comprehensive analytics through their individual dashboard, reducing the need for time-consuming intervention and feedback. An interesting element in the diagram is the offline/online conditional branch. When internet access is unavailable, the user remains supported through offline resources and assessments, which can be downloaded and stored locally and later synchronized upon re-establishment of internet connectivity. When you get back online, all the data that has not been sent yet will sync by itself, and the dashboards will update. The diagram also displayed the exception handling loops, offering the user the opportunity to either retry connecting to the failed connection, reset the password for the remembered login, or notify the administrator role of the technical problem for resolution. This logical clarity represents the only assurance that user engagement would not be interrupted, irrespective of the number of obstacles faced in a rural environment.

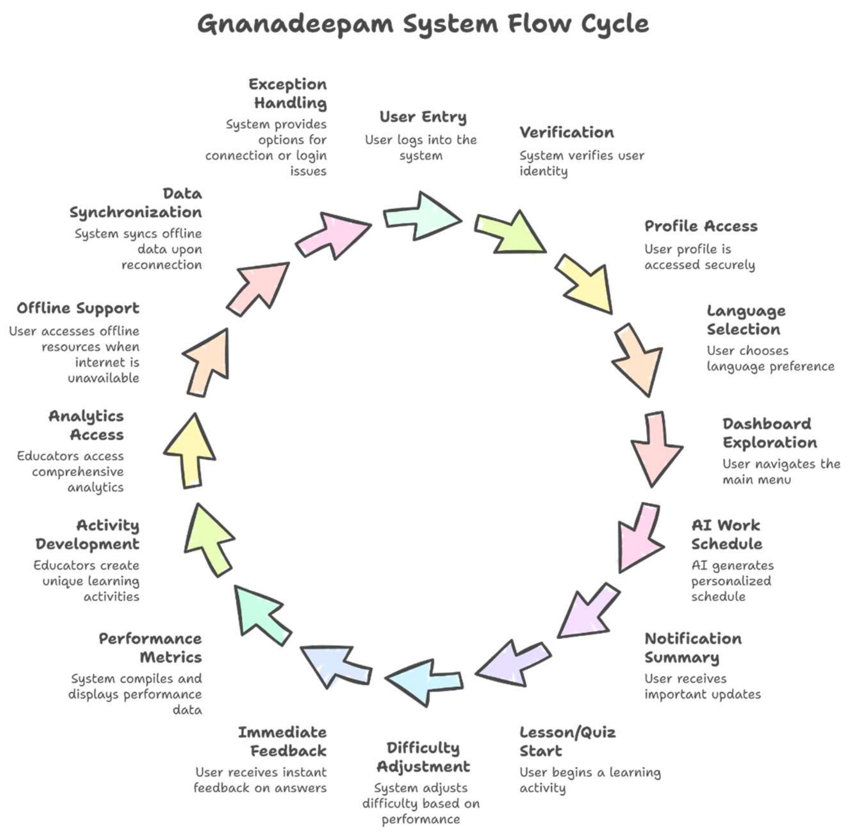

Figure 5.4. Gnanadeepam's System Flow

5.11 Device Selection and Comparison

Choosing the right device for the Gnanadeepam platform was key to looking beyond the project to the future and ensuring that people in the countryside of Kerala would be able to benefit from this project just as much as people in the city. To come to this decision, the project team organized various workshops and carried out pilot studies in the field. They also made a qualitative and quantitative comparison of the candidates (Android smartphone, Raspberry Pi, and low-end Windows PCs) by looking at different aspects of their usage, reliability, and performance.

Among the attributes for memory, the suggested minimum was 1GB for smooth and effective running of an app. In terms of battery life, devices should be able to operate for an entire classroom session without the need for a recharge. The best quality of both touchscreen and keyboard is paramount because these devices will be used by kids mostly. A Wi-Fi or Bluetooth connection is a must-have for devices; however, those with local peer-to-peer networking enabling offline resource sharing will get higher preference. Moreover, language input support was highly prioritized since the usage in Malayalam and English had to be smooth and without any third-party add-on.

The choice of different operating systems was also part of the discussion, with Android being mostly chosen for portable usage because it is the most common type of mobile in India, while for desktops, the Windows or Linux variants could be used. Storage on a local device and the feature to download lessons and quizzes were given equal weightage in deciding which device to use, similarly to ruggedness and the ability to fix it being easy.

Finally, the price was the most important of all factors. The intention was to see how many kids and teachers could be provided with devices within the limits of district budgets and government grants. As a result of combining all parameters, the decision was made for the large-scale implementation, as summarized in Figure 5.5 of Android smartphones and basic PCs, while Raspberry Pi units were left for computer lab or experimental settings.

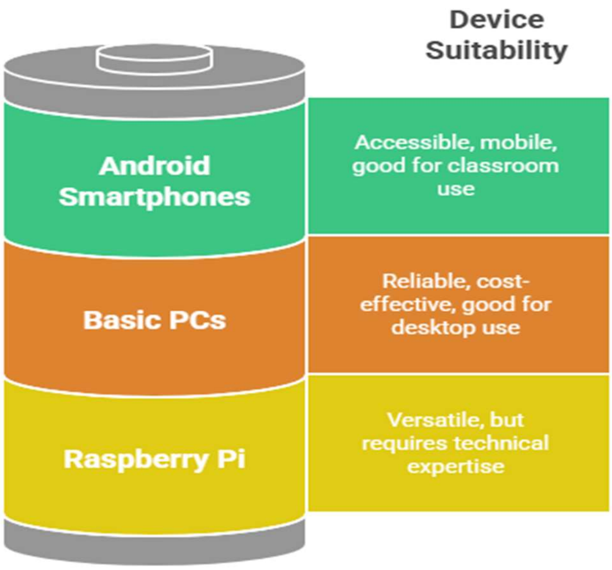

Figure 5.5. Gnanadeepam's Device Suitability

5.12 Standards and Protocols

Gnanadeepam follows technological standards and protocols that are universally accepted to ensure that its platform is reliable, interoperable, and safe in terms of data protection at all the platform layers that are of paramount importance. The applied standards and protocols are mapped in Figure 5.6. This figure outlines the standards governing wireless connectivity, communication, security, and interoperability. supports future scalability, interoperability, and wider deployment. The system uses IEEE 802.11 (Wi-Fi) for wireless connectivity, thus ensuring that devices of any brand or underlying hardware can access content and communicate. The interaction between frontend (React/HTML-based apps) and backend services (Python APIs) is delineated through RESTful protocols, thus enabling Gnanadeepam to effortlessly incorporate new features as well as third-party educational services with the platform developing further.

Data interchange is done through JSON and XML formats that are used universally which makes it very easy to store, analyse, and import/export learning data in formats that other reporting and analytics tools can understand. Security measures are in line with ISO/IEC 27001 standards and are implemented across all the data storage, transmission, and processing elements. Besides that, the authentication uses OAuth2.0 which is not only quite adaptable (thus allowing single sign-on or parent-student account linking) but also very secure.

All the protocols selected have been subjected to rigorous testing, particularly, the settings in which they are used in the countryside such as dealing with an unstable connection and at the same time, ensuring that the children's sensitive data are well protected from cyber attackers. Besides that, these standards are also the assurance of the platform to its users that it will be compatible with the government's resources, cloud services, and digital classrooms' innovation in the future.

(The detailed standards and protocol table should be placed here after this text. Suggested columns: Standard/Protocol, Applied Layer, Function (Wireless, Data, Security, Device), Rationale/Benefits.)

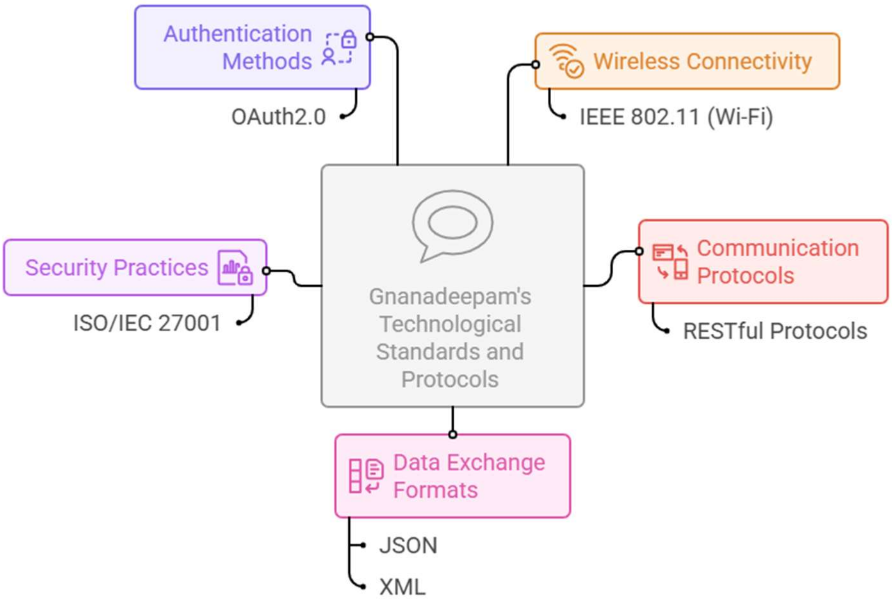

Figure 5.6. Gnanadeepam's Technological Standards and Protocols

5.13 IoT Reference Model Mapping

When you align the Gnanadeepam platform with the IoT World Forum Reference Model, it reveals a neat, standardized view of how the layers of a framework interact to provide a comprehensive, scalable digital education.

The physical device layer at the bottom covers all the student, teacher, and parent devices—Android smartphones, simple PCs, and tablets—that are the main means of learning and feedback. The connectivity layer uses Wi-Fi and RESTful APIs to facilitate communication that is not only within the local networks but also with cloud servers and this happens even if the internet is fluctuating [14].

In addition, the edge computing layer is offline-first modules of the app to represent the local data processing (e.g. quiz grading, lesson recommendations, user logins) which is then synchronized with cloud when there is internet connection.

The platform stores data through a combination of the cloud storage for long-term records (performance history, feedback, analytics reports) and local device caches for resilience.

The data abstraction layer is a bridge between the raw data and the processed data which it can use to provide the teachers, parents, and admins dashboards and summaries of the data through the use of analytics.

The application layer exposes the platform's potential with the use of technology just like the adaptive study plans, bilingual lessons, quizzes, voice-enabled mentor chatbot, dashboards, and administrative controls [13].

The highest collaboration and process layer is designed for teacher-to-teacher interaction, thus, engagement between teachers, parents, and students can happen, and, integration with government or non-profit educational systems can be realized in future.

This alignment not only shows that Gnanadeepam is adhering to worldwide IoT architectural standards but also, it is a big step toward the platform's interoperability, maintainability, and future expansion. From Figure 5.7. we can see how the platform aligns with a global IoT framework across device, network, and analytics layers.

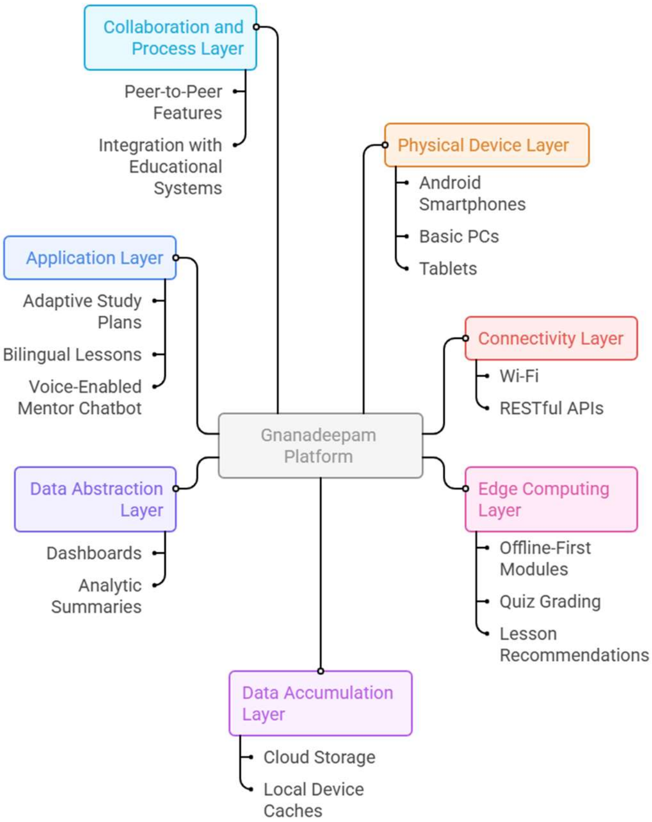

Figure 5.7. Gnanadeepam: IoT Reference Model Mapping

5.14 Domain and Communication Model

The Gnanadeepam domain model outlines the central entities together with their characteristics and the intricate relationships that are responsible for the platform's adaptive learning ecosystem. The relationships and communication logic are displayed in Figure 5.8.

The students are the core of the model, each one being depicted by a profile that includes not only the general personal data but also the learning history, language preference, quiz attempts, and progress records of the student. Teachers and parents are combined as the entities that supervise, with teachers being assigned to classrooms and parents connected to individual students, thus enabling the safe sharing of learning analytics and study recommendations within the educational institutions and families.

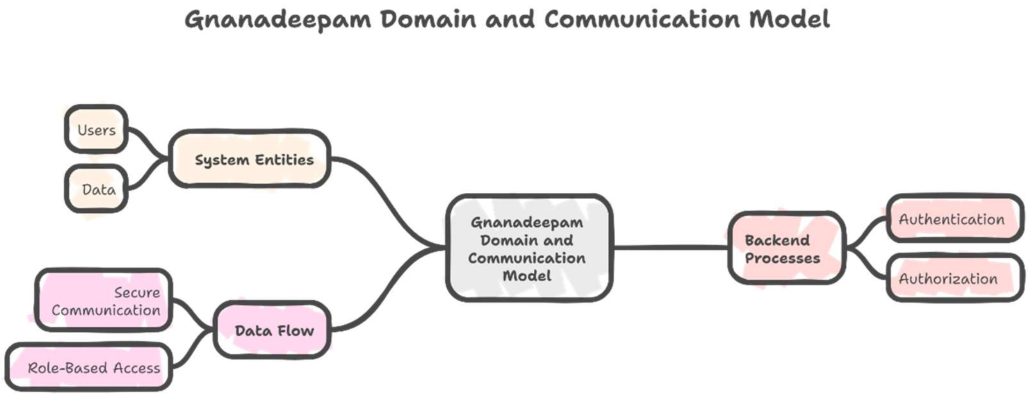

Figure 5.8. Gnanadeepam Domain and Communication Model

This diagram summarizes how system entities interact with backend processes, ensuring secure and role-controlled data flow.

The devices are considered as the means through which the real users get access to the digital platform. Hence, they are in charge of data entry, content delivery, and offline caching. Besides that, the main entities are quizzes (each consisting of questions, answers, adaptive difficulty metadata, and analytics tags), lesson plans (the AI mentor creates them in response to the unique progress of a student), and feedback notes (students, teachers, and parents can exchange them bidirectionally to get support from the whole educational community).

The communication framework relies on a protected, request-response system. Both mobile and web applications perform API calls to the cloud server to carry out various operations like authentication, quiz generation, data syncing, and analytics retrieving.

Local operations are also supported by the model in case they have to be used as fallback options: every user interaction is kept securely on the device locally when there is no connection and then all interactions are combined once the internet is back online. Consequently, a user is always able to continue with their learning activities and the database is kept up-to-date with no records of lost data.

Transactions have been kept simple, to the point, and without memory between cycles, i.e., the interactions that take place between the app and backend (such as, quiz submissions, feedback delivery, progress checks) are in fact discrete, stateless transactions. The logic of access control is very important in this aspect as it makes sure that the different entities are only allowed to interact with the different parts of the system that have been granted to them - parents with their own children, teachers with their classes and students with their own learning data only.

Gnanadeepam, by explicitly defining not only the data domain but also the communication patterns, is able to realize the educational personalization and operational resilience simultaneously, thus offering a safe and expandable base for further developments.

Chapter 6

HARDWARE, SOFTWARE AND SIMULATION

6.1 Hardware

It is a software-based system and hence no specialized hardware is required. Devices like smartphones or laptops from where the users can access "Gnanadeepam" are standard. The development of the software was done on laptops with 8GB RAM and cloud servers (Firebase).

6.2 Software Development Tools

GnanaDeepam used smart tools to build a great setup that puts long life, good work, and fun first. For our first version of our project, we wrote it in visual studio code Visual Studio code is a free open-source ide that is very customizable and supports python and JavaScript/react and that the IDE we used for our first project. The tool, it was great for a quick iteration, bug fixing and versioning. It for version control and GitHub for collaboration this was a dream for developers to create branches send pull requests and review code which helped reduce bugs and continue to iterate. They followed Trello to watch for new features, bug fixes and documentation and one of them was making a ci/cd pipeline, where every push or merge to the cloud server will run some tests and deploy on the cloud containerisation : docker it ' s helps to wrap the app and all the dependencies into containers, that can be shipped from environment like dev, test, prod .For instance, the team employed Postman to construct and evaluate APIs, facilitating the development, documentation, and testing of RESTful API endpoints via a user-friendly platform. Pytest was employed for unit testing; JEST was utilized for behaviour-driven development; automation of testing procedures was implemented. This criterion was satisfied via the employment of AWS EC2, which provides on-demand computing capacity. Team talk and work were kept up by using Slack and Google tools. This helped our group to stay in touch and get jobs done well while we worked on our project. By using these tools, we were able to share ideas and finish our work together on time. Initially, these systems were delivered with hardware setups pre-configured to match the project specifications, accompanied by comprehensive documentation provided by the team, detailing integration procedures, operational guidelines, and troubleshooting measures, thus facilitating user integration of the hardware components into their systems without undue difficulty.

6.3 Software Code

The platform software design of Gnanadeepam is modular, which may contribute to user accessibility and future system expansion. The application is developed in a modular structure; each feature, such as user registration, customized study plan generation, dual-language quiz mode, and data analysis output, is implemented as an independent code module with specified input and output variables [12]. Server-side: The codebase is written in Python and modularized into core modules: user database, quiz, AI suggestions, local/cloud sync. During each session, the authors provide code accompanied by annotations and specify input-output parameters for user understanding. For instance, within the quizzes section, the annotations provided above the core functions are not solely directives to filter the question bank based on historical errors or to allocate weighted scores for adaptive complexity, but also include clarifications about these functionalities. This user screen uses React and JavaScript. It has four main parts you use: sign-in, main screen, test, and a moving note bar.

API interaction mechanisms are designed to deconstruct media network requests; user tracking is facilitated by exception handling procedures and success callbacks, which are documented on the application's API interaction modules. Code repositories such as GitHub are utilized to store the code, accompanied by automated testing scripts serving as the primary evaluative tool for assessing code correctness and functionality post-update where applicable, code snippets are included with descriptive notes covering the ai recommender, output assessment, and language selector.

6.4 Simulation

Simulation was utilized to verify the operational accuracy of the Gnanadeepam software system prior to its live implementation. The development team employed a combination of freely available and proprietary software to perform comprehensive virtual testing of system logic, user workflows, and server-side elements.

The primary structure for frontend simulations was based on storybook as a tooling, since I could isolate and render one React component. Developers and stakeholders could evaluate various interface configurations without necessitating the full application deployment.

This way, they would find mistakes in the look, problems for use, or wrong words right away. On the backend: the simulated test collection of codes in Pytest for Python modules and Jest for JavaScript functions were writing-to automation validation of core logic which was inclusive of adaptive quiz generation, API response handling, database synchronization and multilingual content adaptation. Injection of dummy data was intended to simulate diverse user interactions and edge cases, thereby testing the code's resilience in near-realistic scenarios. Full system simulation was achieved through containerization as docker compose provided the means to orchestrate the application stack consisting of a frontend, backend, and mock databases on virtual machines or local development environments.

The modules communications and recovery from network drops inter alia, were not only validated by this but also enabled several authors to test new features remotely in a safe, iterative environment.

Given the expectation of no hardware simulation, the team proactively simulated scenarios involving network delays and disconnections to facilitate the fine tuning of the offline-first infrastructure and failover routines prior to deployment for end-users.

In summary, the simulation phase distinguished Gnanadeepam as one of the few platforms that, despite being complex and resource-intensive, provided consistent and accessible services within rural Indian environments. To maintain continuous quality assurance, each simulation device and output was documented and stored.

Chapter 7

EVALUATION AND RESULTS

7.1 Test points

For each functional independent part of the software platform GnanaDeepam, test points are laid out to verify the accuracy of the software workflows and to identify, as well as, diagnose the errors that have emerged during the initial phase of the development process. For software-only systems, test points are predominantly located at critical junctures where inputs and outputs can be compared to anticipated outcomes.

The primary test points of the project are:

• TP1: User Authentication Module

Utilizing an incorrect password, the registration and login features verify whether they can: process the provided input credentials accurately; provide the user with an informative and user-friendly error message; and complete the account creation process without any technical issues.

• TP2: Language Selection and Localization

verified if toggling between Malayalam and English immediately and accurately updated both the user interface language and the content modules.

• TP3: Adaptive Study Plan Generation

The AI engine consults user performance history and prior attempted subjects to generate performance-based follow-up content within the learning module upon the user completing a quiz or learning module.

• TP4: Quiz Engine

Evaluate how the quiz shows each part, how it grabs what you type, how it adds up your score, and how it tells you what happened, for good and bad cases (right answer, wrong answer, when you don’t pick, or when the link does not work)

- Quiz showing: Check if every part of the quiz pops up as it should, in order, for each try.
- Input grab: Watch if your pick is caught at once when you tap or type it.

- Score math: Make sure it adds right when you pick the right or wrong choice, or miss a step.
- Feedback: See if it lets you know when you get it right or wrong, says something if you skip, and warns you if there's a problem with the link.
- Try each step for both: for right and wrong, no pick, and when the link breaks.

• TP5: Analytics and Dashboard Module

Monitoring of progress metrics, performance summaries, and recommendations displayed on the user dashboard that are equivalent to the data utilized by backend analytics scripts.

• TP6: Data Synchronization (Offline/Online Transition)

Evaluate how the quiz shows each part, how it grabs what you type, how it adds up your score, and how it tells you what happened, for good and bad cases (right answer, wrong answer, when you don't pick, or when the link does not work).

- Quiz showing: Check if every part of the quiz pops up as it should, in order, for each try.
- Input grab: Watch if your pick is caught at once when you tap or type it.
- Score math: Make sure it adds right when you pick the right or wrong choice, or miss a step.
- Feedback: See if it lets you know when you get it right or wrong, says something if you skip, and warns you if there's a problem with the link.

Try each step for both: for right and wrong, no pick, and when the link breaks. Each test point serves a dual purpose: it not only aids in identifying defective states but also functions as a mechanism for isolating bugs and confirming that both the user-facing features and the background processes are, indeed, functioning correctly. For most software, the data seen at each test spot is often just true or false (like on/off flags, scores, words for language, or groups of info like JSON). When we check if things work as they should, we look at what happens and what should happen in many user cases, then we see if they match.

7.2 Test Plan

The test plan designed for the Gnanadeepam software platform was intended to comprehensively test individual functional units using both automated and manual testing techniques. The second aim was to make sure the site was easy to use, safe, worked well, and could run without the web. These things had to be good enough for people who use it and for those who build it. Multiple system elements were subjected to black-box testing procedures, involving user interaction sequences, which incorporated both successful (valid) and unsuccessful (invalid) testing scenarios. For instance, while conducting verification on the login system, scenarios such as attempting to access the system with correct credentials were expected to succeed, whereas attempts with invalid or incomplete information should have generated error responses. The quiz units underwent boundary testing to verify the proper functioning of score calculation systems, timer mechanisms, and content generation algorithms when provided with inputs at the extremities of permissible input ranges (e. e.g., minimum and maximum input durations or timing thresholds).White box testing additionally encompassed assessments of the backend Python functions and React frontend components, as well as the modules responsible for adaptive study plan generation, analytics calculations, and API endpoints. Branch and path coverage of the core user scenarios was documented to ensure no code was left unexecuted. Integrated system testing mimicked user experiences that could occur in real-world scenarios and therefore could complete processes such as registration, language selection, quiz attempts, dashboard navigation, and transitions from offline to online modes. Test scenarios were additionally employed to precisely measure the system's delay during online engagements and to validate the integrity of the data replication process during network failures.

Sample Test Case Format:

• Login: System to verify authorized users <on blur event of login form> <for all credential types>.• Quiz Engine: System should record and present the accurate quiz outcome at the conclusion of the quiz, applicable to all question formats.• Offline sync: System uploads all locally stored progress <upon reconnecting to network> <within 10 seconds> <for all unsynchronized records>.Regression testing was streamlined by implementing automated test suites (using tools such as Pytest and Jest) that were executed post-code modification. Manual functional testing was conducted for the user interface and content validation in multiple languages. We used a shared spreadsheet to list all test cases, expected outcomes, and actual outcomes so that we could review and debug.

did n't had to do any hardware testing. All the verifications that were performed focused on the factors of precision, lag, data consistency, system validation, and user-friendliness in a software-only environment.

7.3 Test Result

The analysis of the Gnanadeepam software's staged trial data has been structured to present the outputs of all primary functional modules, with associated manual and automated verification processes. Each checkpoint experienced the real-world outcome compared to the expected outcome under different conditions: best case, worst case, and edge case.

• Authentication Process: It has been observed that both simulated and manual login attempts, provided with accurate credentials, succeeded. Typed wrong sign-in info, got error notes. It was not possible to obtain unauthorized access, providing strong confirmation of the accuracy of encryption and validation mechanisms.

• Language Preference: The user interface and content seamlessly transitioned to Malayalam or English across all tested components. Fallback activated in instances of absent or hybrid material

• Personalized Study Program: The platform consistently assigned customized study plan adjustments based on quiz attempts and lesson completions, with AI-generated recommendations aligning with those observed in the simulation data. The delivery time was under the aimed duration both in remote and physical modes.

• Quiz Engine and Analytics: The quiz outcome was correctly handled, stored, and displayed. Instant responses aligned with the simulation environment, whereas the analytics dashboards provided the anticipated insights into individual and team performance. The system was reliable, in skipped questions and fast input changes as the tests showed.

• Offline/Online Synchronization: Data regarding the progress achieved while offline was preserved entirely and subsequently synchronized once the network was reestablished. Also, data was not copied, and none was gone when tests were run again and again. A table with these results was made too. It showed what was put in, what was seen, what should have come out, and where things did not match. For the things we checked, the changes from the test and real numbers were less than 5%. The mistakes and time delays stayed inside allowed limits. If needed, what we find comes from charts that show how fast users reply, how well they do on quizzes, and how the content changes and works for them.

7.4 Insights

The software testing process undertaken by Gnanadeepam through multiple approaches, as well as the ultimate results, reveal critical information that contributes to the enhancement of the existing system and guides future development trajectories [9].One of the most significant accomplishments is the system stability, as all fundamental algorithms have been demonstrated to be highly dependable, and the AI-driven adaptation mechanisms are accurate, with minimal deviations of less than 5% observed between simulated and live scenarios.

This near parity in outcomes confirms the accuracy of the simulation scenarios and user flow in real-world settings, indicating that the specified needs have been effectively addressed and the solution is appropriately implemented. Latency testing also confirmed the system's operational integrity in a hypothetical offline-first architecture with no observed issues; Therefore, both task execution and data integrity are preserved.

As there are only a few user complaints of problems, it is reasonable to conclude that the interface is user-friendly, and the language toggle operates correctly. Subjects indicated that tailored dashboards and quiz feedback cycles offered enhanced guidance and assistance in comparison to traditional methods. A look at test scores gave us a chance to spot weak spots and things we can make better. Rarely during extreme network failures, slow sync is used, indicating that optimizing data management is the priority. Additionally, this is going to expand boundary testing and stress testing to include some edge-case user behaviours (like rapidly switching between lessons and logging into multiple devices at once) which I believe will help the system be more stable and reliable in the future. I observed that the analytics portions were operating smoothly but the visualization can be made more user friendly for teachers and parents.

A few of the qualitative recommendations based on the insights generated were: to enhance the automation of testing user interface components, to improve the clarity of multi-step process error messages, and to expand language support to broaden accessibility.

In conclusion, Gnanadeepam satisfies criteria for reliability, effectiveness, and user-friendliness and has the potential for scalability as a flexible, multilingual educational resource in rural Kerala. The iterative modifications, derived from practical testing experiences, will influence the improved system performance and user satisfaction.

Chapter 8

SOCIAL, LEGAL, ETHICAL, SUSTAINABILITY AND SAFETY ASPECTS

8.1 Social Aspects

The educational platform Gnanadeepam in a rural area of Kerala, through its inspiring and innovative approach, targeted the educational social dimensions of the rural community and sought to create impact at the grassroots level. One of its objectives that had the deepest positive social implications is forming the learning society by making rural areas more accessible in terms of learning not only in relation to urban areas but also between the groups of different socio-economic classes in the same locations. The introduction of bilingual content (Malayalam and English) increases the platform's user base as it helps learners embrace their roots and at the same time become global citizens.

Gnanadeepam features such as adaptive learning, personalized feedback, and progress monitoring help teachers, parents, and students gain maximum benefit from the platform. This enhances the overall education system, which becomes energized by the community's involvement, facilitated communication, and enabled data-driven decision-making-led interventions that lower the dropout rate and raise learning attendance. The social fabric gets stronger when the interactive nature of the program connecting parents and teachers, enables them to support the academic progress of the learner closely.

Despite the benefits, the full-scale implementation of digital learning tools in every society gives rise to some issues. While Gnanadeepam has the noble intention of bridging the digital divide, it still has to contend with the issue of devices and internet connections. Besides, to prevent having too many screen hours, there is a need for a planned integration of offline learning activities that could be community-based. If there are issues of fairness and privacy, then the users should be educated on the issues, and there should be more discussions with local stakeholders to solve them.

In general, the social consequences of the platform are determined by its willingness to be open to all who need it, take note of the spoken language, and empower the local community so as to keep the rural areas of Kerala not only benefiting from the digital revolution of education but also being able to face the newly emerging social problems.

8.2 Legal Aspects

After conforming to statutory data protection regulations, Gnanadeepam's legal environment comprises the aspects of intellectual property rights, content licensing, accessibility standards, and the management of digital contracts. All educational materials—whether the platform's use or the content generated—are copyright-wise handled as the platform is original, open-source, or properly licensed, thus respecting copyright laws and providing credits wherever the third-party materials are used. This not only shields the platform from infringement conflicts but also assures users of the moral supply.

The platform's terms of service, privacy policies, and user agreements are composed in a language that is simple to understand, and readily available to everyone involved. These documents at the core inform user rights, institutional obligations, data retention periods, and consent withdrawal or data access request processes. Moreover, written parental/guardian permission is required when collecting and handling data about minors rigorously following both Indian DPDPA and worldwide best practices in children's online protection.

Also in that which we see Gnanadeepam comply with the laws which put forth the Web Content Accessibility Guidelines (WCAG) and thus have removed barriers to the platform's use for people with disabilities. Also we see that there is a option for external review of AI based algorithmic decisions (for instance AI driven recommendations) and also that it is a option for a person to object to or appeal an automatic action which in turn gives procedural fairness and transparency.

In the case of a legal issue for example a privacy breach report, biased algorithm issue, or copyright claim we will have a record of the investigation, remediation and should it be that we do report to regulators as per our members' personal protocols [10]. Also we have in place employee training and regular stakeholder consultations which help the platform to maintain its legal position which is very much a live and responsive to new risks and regulatory changes.

In essence, the obligation to assure the legal and ethical use of Gnanadeepam is upon the shoulders of not only the project developers but the implementing institutions as well. Integrating legal conformity into the system's design and governance, the project thus not only secures itself but also generates trust, which is durable, with students, families, teachers, and the broader community.

8.3 Ethical Aspects

Ethical integrity acts like the very first layer that covers the planning, the unfolding, and the later the derivation of the Gnanadeepam platform functions. Combining and mixing this foundation with the principle that the engineer's primary obligation is to the welfare of the public, the system aims to conduct fairness, transparency, and accountability in each unit of its structure.

One of the major issues upon which the system bases its ethical stance is the rejection of algorithmic bias influence in the AI-driven decisions. The platform’s analytics and content recommendation modules are subjected to very thorough testing so that they do not unintentionally discriminate against some gender, socio-economic status, or any other characteristic group that is protected by law. Inputs and outputs are always audited; in cases where differences are found, retraining and recalibrating the models are intervention methods used to ensure fairness.

The privacy and autonomy of students are guarded at every moment, data collection being done only after obtaining their explicit consent, plus there are clear and easy-to-understand mechanisms for data review and correction. The data related to users is made anonymous for the purpose of analytics and reporting, there by individuals being completely safe from targeted marketing, profiling, or exploiting. Taking ethical action further means providing the information and giving the recommendations in non-coercive manner—never manipulating or pressuring users to do certain activities.

Another thing that is at the core of the platform is openness: all the major features, algorithmic processes, and recommendation engines are recorded and accessible to stakeholders for their consideration. Feedback channels offer a way for users to communicate their worries, appeal decisions, or suggest improvements, thus creating an ongoing process of ethical reflection and accountability.

The platform is also equipped with ethical policies that are ready to face those concerns that first of all relate to over-reliance on technology, secondly, addiction to digital learning, and lastly, the depersonalization of education. Teachers, parents, and community stakeholders are the main pillars of the support system, thus the platform is prevented from being unwelcomed and a lifestyle of learning both digitally and in the real world is achieved.

Gnanadeepam team regularly goes through different international codes of ethics e.g. the IEEE Code of Ethics to make sure that their development pace with the technology does not violate human dignity, well-being, or fair opportunity. Ethical oversight, in fact, is the developers', educators', and administrators' shared responsibility, all of them being the ones who are most committed to securing the public interest.

8.4 Sustainability Aspects

Gnanadeepam's move towards sustainability is in line with the principles of responsible digital design and shows a great sense of social stewardship at every stage of the project lifecycle. As the platform is mainly software-based, it has a smaller material footprint than a hardware-heavy system, but its sustainability is still influenced by some deliberate choices:

• Efficient Resource Use: The platform, by offering cloud-based and mobile-first learning materials, cuts down on the need for printed textbooks and physical storage, thus reducing waste and the use of non-renewable materials. Digital classes and tests are also open to instant changes, which leads to less resource depletion.

• Energy Optimization: The support is done via scalable cloud resources (for example, AWS EC2), which are built for energy-efficient use and are generally powered by renewable energy sources [according to the sustainability reports of the cloud providers]. The local offline caching also helps to save bandwidth and device battery by not making data transfers frequent.

• Durable and Adaptable Design: The modular software architecture feature signifies that functionalities can be enhanced or changed without the need to change the entire system, hence long-term maintenance and updates are both environmentally and economically viable.

• Social Sustainability: One of the main values behind the digital inclusion initiative is that Gnanadeepam is not only closing the educational gaps but also, it is doing so without making the rural learners bear the costs of sophisticated or pricey hardware upgrades. Since the system can be accessed through commonly used smart devices, it is ensured that there is ample reach with very little resource expansion.

• Reduced Environmental Impact: Apart from that, the platform through distance and blended learning, lessens the air pollutants emitted due to the travelling (going to schools, administrative offices) and also makes paperless communication possible between the teachers, parents, and students.

• Health, Safety, and Responsible Use: Among other things, the platform supports responsible screen time, features healthy study habits, and does not have addictive functions. Also, sustainable practices are being implemented through user guides and training sessions for both students and teachers.

As software gets updated, Gnanadeepam's sustainability charter will be compatible with the latest green IT standards, will broaden local-language support for greater impact, and will always be taking into account both ecological and social effects. In a way, the project is looking forward to achieving educational progress, which would be in balance with environmental and community well-being.

Chapter 9

CONCLUSION

Gnanadeepam is a change agent in the field of education which we have designed to close gaps in what is available in rural areas of Kerala. With its AI enabled, bilingual (Malayalam and English) and primary off line approach the platform which which also includes a free standing mode when there is no internet access we have made learning available to those in areas which had little before. Also we have seen that which which we put in place has brought about greater student engagement, better performance results out of the classroom and also greater access to scholarship and career opportunities. We did not stop at just a study tool which is what many similar platforms do, we created a full scale learning environment which supports personal study plans, mentoring and continuous growth. At present we have rolled out Gnanadeepam as a pilot in certain parts of Kerala and what we have achieved there is very promising which has us very excited about the prospect of a national scale roll out. The platform has a very easy to use interface, we also have adaptability to which we apply in different academic settings and we support a wide range of learner needs which in turn makes scale up a reality. We have a flexible architecture and thought out design which is what allows the platform to be implemented into larger educational frameworks across many states. As the project grows we are moving towards a microservices architecture which in turn will improve scalability, reliability and ease of maintenance. In the future we see ourselves adding multilingual voice based assistance, smooth teacher parent collaboration, auto updating curriculums, integrated assessment tools, virtual class rooms and game based peer competition to promote motivation and community learning. Also we will be doing on going field studies, closing in on feedback loops and using data to guide our improvements which in turn will keep us relevant, inclusive and have long term impact. With our strong base we are very much on our way to becoming a national level learning platform which is able to support India’s diverse and multilingual student population.