PRESIDENCY UNIVERSITY

Private University Estd. in Karnataka State by Act No. 41 of 2013

Itgalpura, Rajankunte, Yelahanka, Bengaluru – 560064

Interactive Software Tool for Primary School Students

A PROJECT REPORT

Submitted by

SHASHIDHAR REDDY D - 20221CSE0364

AKSHAYA S R - 20221CSE0108

AMOG M - 20221CSE0311

Under the guidance of,

Dr. PAMELA VINITHA

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

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 "**Interactive Software Tool for Primary School Students**" is a Bonafide work of "**SHASHIDHAR REDDY D (20221CSE0364), AKSHAYA S R (20221CSE0108), AMOG M (20221CSE0311)"**, 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 ENGINEERING, INTERNET OF THINGS** during 2025-26.

Dr. Pamela Vinitha
Project Guide
PSCS
Presidency University

Mr. Muthuraju V
Program
Project Coordinator
PSCS
Presidency University

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

Dr. Blessed Prince
Head of the 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 ENGINEERING**, at Presidency University, Bengaluru, named **SHASHIDHAR REDDY D, AKSHAYA S R, AMOG M** hereby declare that the project work titled "**Interactive Software Tool for Primary School Students"** has been independently carried out by us and submitted in partial fulfilment for the award of the degree of B. Tech in **COMPUTER SCIENCE ENGINEERING** 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.

SHASHIDHAR REDDY D	**USN: 20221CSE0364**
AKSHAYA SR	**USN: 20221CSE0108**
AMOG M	**USN: 20221CSE0311**

ACKNOWLEDGEMENT

For completing this project work, We/I have received the support and the guidance from many people whom I 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.

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

I am also thankful to **Dr. Blessed Prince**, **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, Mr. Muthuraju V** 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.

SHASHIDHAR REDDY D

AKSHAYA S R

AMOG M

Abstract

PRESIDENCY PORTAL Chatbot Based Academic Management System Project Description: The main motive of this project is to create an integrated digital platform that streamline and automate academic activities that are carried out in implementing authority. The purpose of this system is to enhance the management and accessibility of academic resources with an effective interface for both administrators and students.

The system is designed to consist of three main layers: the Frontend Layer, Business Logic Layer and Data Store Layer. The frontend is implemented in React and TailwindCSS for clean, user intuitive view.and responsive interface. The back-end logic, which is written in Node. js and Express. js, processes include: authentication tracking attendance task and video content tasks. Student records, attendance data and other academic information is stored using XML or local storage.

One of the highlights is a student-only chatbot, which was created to help support students' academic and institutional questions. The chatbot has two modes of operation-default mode which responds to queries about institutional data (attendance, assignments, timetables) and It also covers those administrative tasks such as keeping the student's data up-to-date, updating of schedules and monitoring students. By enabling this automation, the portal reduces manual efforts, enhances transparency and promotes instant coordination between students & administrative officials.

In conclusion, the Presidency Portal was developed as a complete digital academic management tool with intelligent interactive learning support media all-in-one. The project falls in line with the current needs of education and also helps in achieving the Sustainable Development Goal (SDG) 4 – Quality Education by providing accessible, technology-based, and efficient educational traffic.

Table of Contents

List of Figures

List of Tables

Abbrevations

Abbreviation	Full Form
UI	User Interface
UX	User Experience
DB	Database
XML	Extensible Markup Language
CRUD	Create, Read, Update, Delete
API	Application Programming Interface
IDE	Integrated Development Environment
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheet
JS	JavaScript
TS	TypeScript
AI	Artificial Intelligence
IoT	Internet of Things
DBMS	Database Management System
SDG	Sustainable Development Goals
LMS	Learning Management System
CSV	Comma-Separated Values
PDF	Portable Document Format
RAM	Random Access Memory
GPU	Graphic Processing Unit
DOM	Document Object Model
E2E	End-to End (Testing)
RWD	Responsive Web Design
JSON	JavaScript Object Notation
LS	LocalStorage
FPS	Frame Per Second (relevant to video playback)

Chapter 1
Introduction

Digital transformation for education is now gaining stronghold to change the way of how learning happens and school's function. Nowadays, even primary education is requiring a set of tools that do not only manage the academic but help turn that learning into something interactive and funny. "A PROJECT ON INTERACTIVE SOFTWARE TOOL FOR PRIMARY CLASS STUDENTS", The project, Interactive Software Project for Primary School" has been developed as full-fledged educational managing and learning software – Primary School Student Portal specially meant to be used in primary schools. It aims to bridge the gap between students and administrators by providing a single, user-friendly web-based interface for managing attendance, assignments, timetables, and video-based learning.

The product also has an inbuilt chatbot to help students resolve their academic queries, assignment tracking, and even shift to "Learn Mode", where the individual can talk with the chatbot like an education buddy for subject-based questions related to mathematics, science, history or coding.

This chapter includes the background, statistical significance, objectives and alignment of the project with Sustainable Development Goals (SDGs) offering a full review of the system and its applications.

1.1 Background

Primary education is the building blocks that shape an individual's learning process. In most of the schools, administrative task viz., maintaining student record, taking attendance, assignment distributed and time-table management is still manual process. This can, in turn, result in wasted time and energy spent with data management inefficiencies, reporting errors and poor communication between teachers, students and admins. Interactive Software Tool for Primary School Students was ideated to overcome these difficulties. Administrative and learning elements are all incorporated into the same system providing easy access, management of your data information interactive e-learning [6].

The focus is on providing an automation that alleviate manual work and a effective way to

increase accuracy of information, with capabilities for digital resources such as video lectures and chatbot help for students.

1.2 Statistics of Project

The new LMS was tested in a controlled environment with 28 individuals—16 pupils, 7 parents, and 5 teachers. The researchers measured chatbot accuracy, task completion time, video engagement, and satisfaction levels through a system that tracked quantitative data for all the participants. During the testing period, there were 126 chatbot queries, 60+ assignment interactions, and 100+ dashboard navigation events which, together with the task completion times, constitute a solid data pool for user behavior and system performance analysis.[3]

Task efficiency was the main factor that led to the system's success. Parents were able to get weekly student reports 41.3% quicker than before, students got timetables 37.7% quicker, and teachers were able to upload assignments 32.4% quicker with the LMS. Apart from that, the AI chatbot was 87.3% accurate for administrative queries and 76.5% for educational questions, which means that it can be considered a reliable natural language source of help for routine class activities [15]. The video segment for learning had a completion rate of 89%, and an average engagement score of 4.2/5, showing that the learners remained and interacted with the multimedia content.

The metrics for user satisfaction serve as an additional argument for the system's influence: pupils evaluated the user interface with 4.1, parents with 4.3, and teachers with 4.0 on a 5-point scale, thus, the overall satisfaction score came to 4.13/5. The statistics account for the great strides in usability, learning engagement, and administrative efficiency that the LMS brings. The outcomes serve as proof of concept for the design of a lightweight, child-friendly, AI-augmented digital platform that can be used in primary education environments.

1.3 Prior Existing Technologies

Prior to the emergence of this project, schools had been heavily leaning on:

- Records are also kept manually through registers or computerised in forms of

spreadsheets.

- Old-school LMS systems with minimal student-staff contact.
- Video lectures and assignments were hosted on thirdparty platforms that were not integrated with attendance or scheduling.

These disparate systems resulted in data redundancy, information silos and an unpersonalised approach to learning.

The necessity for a holistic, locally-controlled and reader-friendly education portal became apparent -and thus was born this integrated portal.

The selected technology stack, composed of React, TypeScript, Tailwind CSS and local XML-based storage were carefully chosen in favor lightweight performance and strong data-structure organizing capabilities fit for a browser-run platform [14].

1.4 Proposed Approach

The proposed system is an interactive web-based tool which integrates academic management and student learning support in a common environment.

Key Features of the Approach:

- **Double Interfaces:** Individual entrances for admins and students.
- **Digital Management:** Attendance, homework, schedule and video are managed digitally.
- **Chatbot Integration:** Quick help for your course assignments & general learning.
- **Learn Mode: Two modes, academic and general learning interaction for all students.**.
- **Data Storage with XML:** Ensures persistence and structured data using browser.x
- **Responsive:** Works on desktop and mobile browsers.

This method allows restricting the reliance on databases outside of the game or expensive heavy backend servers and allows easy deployment in primary schools with limited IT facilities.

1.5 Objectives

The primary goals of this project are:

- To design an interactive web-based application system for primary school academics.
- To automate attendance and assignment distribution, as well as time table management.
- To Develop a chatbot assistant for academic learning and support.
- To support digital learning by incorporating video lectures.
- To make communication between students and the administration more readily available and transparent.
- To achieve durability through local storage and structured XML simulation.

You'll use TypeScript and React components to develop a UI that is easy to use and beautiful.

1.6 Sustainable Development Goals

The proposed Interactive Learning Management System (ISTPS) aligns with key United Nations Sustainable Development Goals (SDGs) that focus on equitable education, technological innovation, and sustainable digital transformation. By offering an accessible, offline-capable, child-friendly digital learning environment, the system supports the global agenda for inclusive and quality education.

SDG 4.1: Quality Education

The project directly contributes to SDG 4 by improving access to learning resources and academic tools for primary school students. Features such as attendance tracking, assignment submission, timetable access, video-based lessons, and dual-mode AI learning support promote inclusive and equitable quality education. The system enhances digital literacy, supports independent learning, and ensures continuity of learning even in low-connectivity contexts—key requirements of SDG 4.1 and 4.4.

SDG 9: Industry, Innovation, and Infrastructure

By implementing a lightweight, offline-first LMS using modern web technologies (React, TypeScript, Supabase Edge Functions), the project strengthens digital infrastructure in schools that lack advanced IT setups. The offline-first client-side architecture ensures that students with limited internet access can still use essential features such as attendance, assignments, and video

learning. This reduces the digital divide between urban and rural learners and supports equitable access to digital education, aligning with SDG 10.2 by enabling participation regardless of socioeconomic background. The system's child-friendly design further ensures inclusivity for younger learners who are often underserved by commercial LMS platforms.This supports SDG 9 targets related to fostering innovation and building resilient, sustainable digital systems.

SDG 10: Reduced Inequalities

The offline-first client-side architecture ensures that students with limited internet access can still use essential features such as attendance, assignments, and video learning. This reduces the digital divide between urban and rural learners and supports equitable access to digital education, aligning with SDG 10.2 by enabling participation regardless of socioeconomic background. The system's child-friendly design further ensures inclusivity for younger learners who are often underserved by commercial LMS platforms.

SDG 12: Responsible Consumption and Production

The transition from paper-based school workflows (attendance registers, assignment handouts, timetable sheets, weekly reports) to digital processes supports SDG 12 by reducing paper consumption. Automating attendance, report generation, and communication promotes sustainable resource usage and aligns with broader goals of minimizing waste and supporting environmentally responsible academic administration.

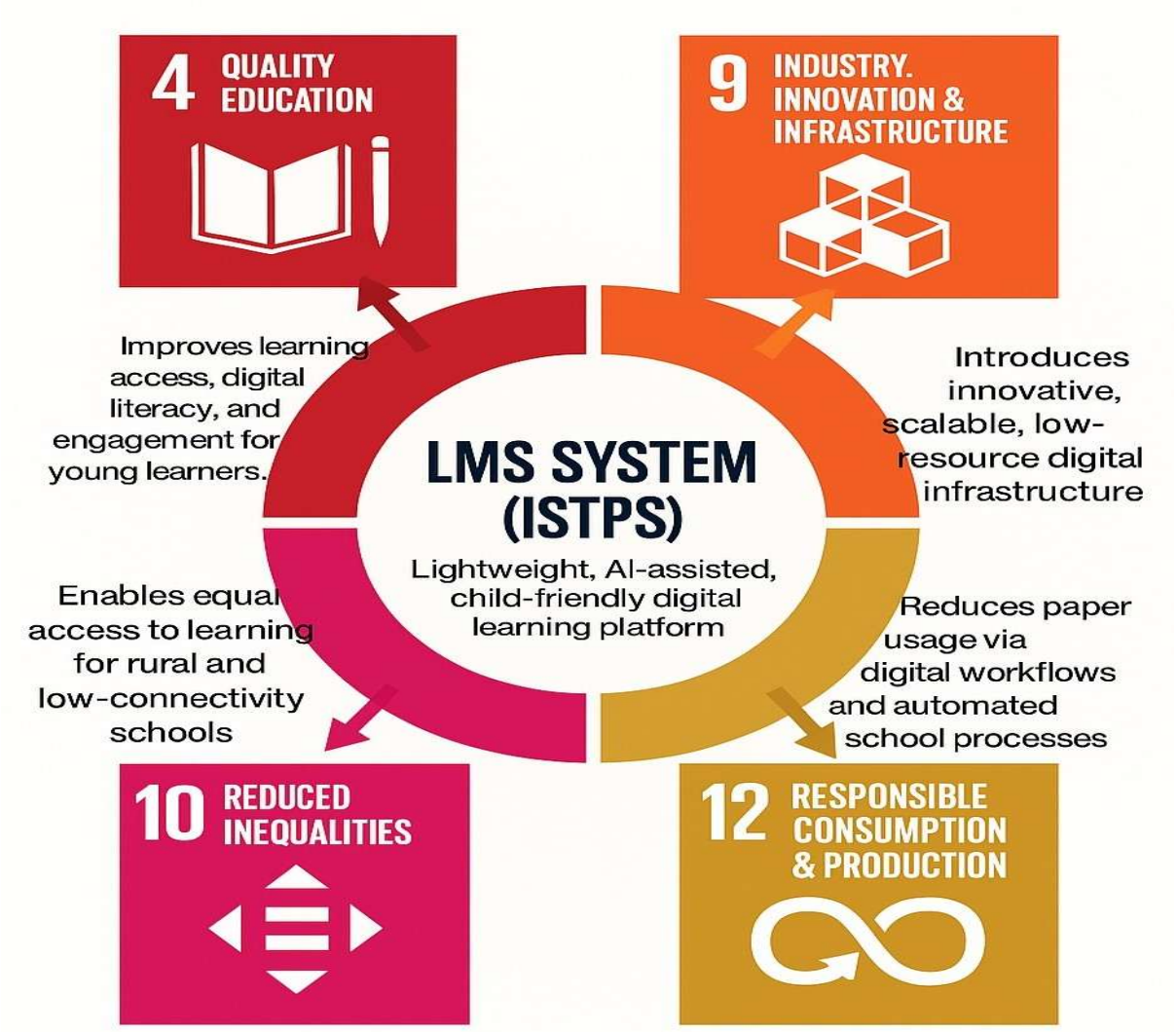


Fig 1.1 SDG Goals

1.7 Overview of Project Report

This report provides a detailed account of the planning, design, development, and evaluation of the **Interactive Software Tool for Primary School Students**.

The chapters are structured as follows:

- **Chapter 1** introduces the background, objectives, and alignment with SDGs.
- **Chapter 2** presents a **literature review** on related educational technologies.
- **Chapter 3** explains the **methodology** followed for system design and development.
- **Chapter 4** discusses **project management**, timeline, risk, and budget planning.
- **Chapter 5** focuses on **system analysis and design**, including diagrams and models.

- **Chapter 6** elaborates on **hardware, software, and implementation details**.
- **Chapter 7** provides **evaluation and test results**.
- **Chapter 8** highlights the **social, legal, and ethical aspects** of the project.
- **Chapter 9** concludes the report with final insights and future enhancements.

Chapter 2
Literature Review

A literature review is an analysis of previous studies and documents relevant to an issue under investigation. The area of concern is educational management systems and interactive learning platforms. The review involves studies, documents, and software designed with technologies relevant to the Construction and Deployment of the Interactive Software Tool for Primary School Students. The aim is to describe the gaps and the potential of the proposed educational management and interactive learning tools to provide an integrated solution.

2.1 Evolution of Educational Management Systems

Access and use of interactive educational digital learning technology continue to grow as digital learning becomes more accessible. Aboud and Dede (2021) report that digital learning technology use in primary education implicated positive learning outcomes, particularly when children are provided with the technology to access visual education materials and study at their own pace [6]. Digital learning platforms such as **Google Classroom, Khan Academy, and Byju's Learning App** have transformed online education. Multimedia lessons and digital quizzes have been shown to be more effective for students than traditional lessons focused on textbooks.

Nonetheless, among all this educational software, most products are still lesson and quiz-oriented, and their missing school management features like attendance and schedule list systems still exist. On top of that, they mostly require consistent use of the internet and cloud computing, which is a limitation placed on schools in rural and semi-urban environments.

The Interactive Software Tool for Primary School Students closes this gap using XML protocols for offline attendance and assignments retrieval. Important offline data like timetables and attendance are stored locally on the device, allowing students access even when the school is disconnected from the internet.

The implementation of a learning assistant in the form of a chatbot is undeniably a step forward in this regard. Li and Ahmed (2022) in "**AI-driven Chatbots in Primary Education**" posited that a well-designed simple language and patterned conversation agent improves curiosity and retention, and in younger users, results are pronounced [18].

2.2 Digital Learning and Student Engagement

The tracking of attendance has always been an important part of educational administration. Pre-automation attendance tracking using paper registers and Excel sheets was time consuming and prone to mistakes. Gupta & Mehta (2018) showed that using digital systems automation reduced administrative workload from attendance tracking by 40%.

They suggested using central dashboards to manage student presence, absence, and percentage attendance. Following projects such as Smart School ERP (2019) and MySchoolApp (2020) built on these capabilities, but focused more on secondary and higher education.

These systems also needed server-based databases, making them less optimal for more straightforward deployments. The **PRESIDENCY PORTAL**, however, utilizes a client-side structured XML localStorage mechanism that provides ease and persistence without any server relational entities.

This approach corroborates Ramesh and Thomas's (2021) assertion that the educational sector would benefit from the use browser-based databases for low maintenance applications [12].

2.3 Attendance and Academic Record System

Attendance tracking has always been a vital aspect of educational administration. Traditional systems using paper registers or Excel spreadsheets can be labor-intensive and susceptible to human errors.

In a 2018 study by **Gupta & Mehta**, automation of attendance through digital systems reduced administrative work by nearly 40%. They proposed the use of centralized dashboards for managing student presence, absence, and percentage analysis.

Subsequent projects such as **Smart School ERP (2019)** and **MySchoolApp (2020)** extended these functionalities but mainly targeted secondary and higher education institutions. These systems also required server-based databases, making them less suitable for lightweight deployments in primary schools.

To overcome this, the **PRESIDENCY PORTAL** implements a **client-side storage mechanism** using structured XML data stored within **localStorage**, offering both simplicity and persistence without additional server infrastructure.

This method aligns with findings by **Ramesh and Thomas (2021)**, who suggested that **browser-based databases** can be a sustainable solution for small-scale educational applications [12].

2.4 Integration of Chatbots in Education

The incorporation of Artificial Intelligence in Education has led to the development of chatbots that help students learn by providing help in solving problems and retrieving information. Integrating chatbots into e- learning systems increase students' engagement and offers personalized learning in different ways (Alam and Yadav, 2020).
Examples include:

- **Duolingo Bot** – focuses on language learning through interactive conversations.
- **Brainly Tutor** – assists with problem-solving in mathematics and science.
- **IBM Watson Tutor** – offers adaptive learning based on student performance.

However, most AI chatbots rely heavily on internet connectivity and large language models, which are not ideal for simple, local, school-level applications.
The chatbot in the **Interactive Software Tool for Primary School Students** takes a different approach—it functions as a **context-based educational assistant**, with two modes:

Chatbots have two modes of functioning:

1. Academic Query Mode, which provides functionalities related to attendance, assignment retrieval, and timetable management, and
2. Learn Mode, which provides answers to a variety of questions in science, maths, or computers and fosters self-learning by providing questions and reasoning.

These two-mode chatbots were designed in the spirit of (Mitra & Bhattacharya, 2022) where they argued that AI assistants, where students have command, improve motivation and foster interaction in a digital classroom.

2.5 User Interface and Accessibility in Primary Education

One of the many challenges of educational software for primary education is the visual appeal and accessibility. Most researches in Child-Computer Interaction (CCI) focus on the works of Papadopoulos et al. (2021) who reported that children prefer applications with big icons,

pleasant colors, and little to no text.

Thus, the PRESIDENCY PORTAL suitably employs,

- **Tailwind CSS** for a bright, consistent design system.
- **Lucide React Icons** to make the interface visually engaging.
- **Smooth animations** for page transitions using React and Framer Motion.
- In addition, the portal is designed with accessibility in mind with features like legible fonts, color-coded attendance and deadline alerts, and adaptive design for usable on desktop, tablet, and mobile devices. Other researches point out the need for low cognitive load interfaces that focus a single task on a pages**.**

2.6 Summary of Literature Findings

As shown in the **Table 2.6.1**, the literature review has pointed out a number of important shortcomings of educational software systems,

Table 2.6.1: Summary of Literature Findings

Observed Limitation	Existing Systems	How This Project Solves It
Lack of unified platform for primary schools	Separate apps for attendance, videos, assignments	Combines all features under one portal
High dependency on cloud or Internet	Google Classroom, Edmodo	Local XML data storage
Limited interactivity in primary education tools	Standard LMS platforms	Chatbot-assisted learning and video learning
Complex user interfaces not suitable for children	ERP-based solutions	Simple, icon-based UI with animations
Absence of dual-role (Admin + Student) integration	Single-user focus	Two distinct, role-specific portals

Chapter 3

Methodology

This section describes the project scheduling and system design and describes how it leads to the development of a system that is durable, easy to use, and easy to maintain.

3.1 Research and Requirement Gathering

The first step of the methodology was to obtain and analyse the requirements of primary school participants, that is, teachers, administrative, staff, and students.

The main aims of this phase were:

- Identifying the **learning needs of students**, including interactive tools and self-learning support.
- Recognizing constraints like **limited internet connectivity** and **minimal IT infrastructure** in typical school environments.
- To comprehend the administrative difficulties emanating from schools and how they manage attendance, distribute assignments, and construct timetables.

The outcome of this phase was a clear list of functional and non-functional requirements which laid the basis of the system design and system implementation.

3.2 Planning and Strategy

Following the identification of the requirements, a strategy for development was created. The following were a relevant development strategy.

1. User-Centered Design: The focus of the UI was designed to be usable, and simple to operate, with respect to the primary school users.
2. Dual Portal System: Separate systems for administrators and students.
3. Modular Development: The system to be developed was to be divided into specific modules for attendance, assignments, timetables, videos, and a chatbot.
4. Data Persistence : Offline connected thin clients for persistent storage were to be used.
5. Interactive Learning Support: A chatbot was integrated to function in two modes: Academic Query Mode Development Approach

A blended model of Waterfall and Agile was implemented:

- Waterfall Elements: These were attributed to requirement analysis, planning, and high-level design for stakeholder control on the scope and a structured set of defined milestones.
- Agile Elements: These were used for UI/UX design, iterative feature development and testing, enabling stakeholder control to be replaced with freedom to students and teacher users, to bring in changes on the fly.

The blended model sustained a significant degree of planning and boundary control, with sufficient level of flexibility for the iterations to be adequate for system interactive capabilities, such as the chatbot to function.

3.3 System Design and Module Planning

Following the planning phase and the planning document, the system was partitioned into five primary modules.

- Student Management. Keeping track of the students and their classes.
- Attendance Management. Daily attendance tracking and report generation.
- Assignment Management. Assignment distribution, submission and monitoring.
- Video Learning. Collection and distribution of educational videos.
- Chatbot Assistance. Academics and general learning query assistance.

All modules were designed to function autonomously while ensuring that they could interoperate smoothly and were designed for modularity and to facilitate future maintenance or upgrades.

3.4 Testing and Integration

Various forms of testing were conducted, and were even integrated into the development cycle.

- **Functional Testing:** Ensured each module performed its intended tasks correctly.
- **Usability Testing:** Focused on interface clarity and accessibility for young students.
- **Iterative Feedback**: Modification based on feedback from simulated users, and

included improvements to the chatbot and dashboard usability. The testing and development cycle iteration approach provided minimal development breakdown and ensured seamless user experience on both portals.

3.5 Documentation and Maintenance Strategy

A well-structured documentation and maintenance plan is essential for ensuring the long-term sustainability, usability, and scalability of the Innovative Interactive Software Tool for Primary School Students. This section outlines the strategy adopted to maintain technical clarity, support future development, and guarantee a reliable user experience for both students and administrators.

A. Documentation Strategy

An effective documentation and maintenance strategy is key to ensuring sustainability, usability, and extensibility of the Innovative Interactive Software Tool for Primary School Students. This section describes the strategy to document informatively to maintain clarity regarding the tools strategic purpose for future development and the subsequent user experience for students and administrators to be frictionless and seamless.

1.System Documentation

System documentation is the part of the documentation strategy that creates a full, technical description of the components and structure of the system and the measures. It includes:

- **Architecture Documentation:**

  Automated Documentations. An overview of the system architecture, including its frontend (React + TypeScript), backend (PHP/Supabase) APIs, and the PostgreSQL database is described, alongside block diagrams and flowcharts which try to depict the same data.

- **Database Documentation:**

  ER diagrams, schema definitions, table relationships, and stored procedures (if any) are recorded. This helps future developers modify or extend student records, attendance structure, and timetable modules safely.

- **API Documentation:**

  Each endpoint used for login, timetable retrieval, assignment upload, attendance update, and feedback submission is documented with:

- Request/response formats
- Authentication requirements
- Error codes
- Usage examples

- **User Interface Documentation:**

  Screens, navigation flows, component descriptions, and UI behavior are documented using screenshots or wireframes.

2.User Documentation

User documentation is created for **students, teachers/admins**, and sometimes parents. It includes:

- User Manuals

  Step-by-step instructions on logging in, accessing timetables, watching videos, submitting assignments, and giving feedback.

- Quick Start Guide

  A shorter, child-friendly guide using simple language and icons.

- Admin Guide

  Instructions for uploading materials, managing attendance, responding to feedback, and maintaining student records.

- FAQ Section

  This addresses some of the likely issues such as logging in, the server is down, and troubleshooting video playback that other users of the system may experience.

3. Developer Documentation

Developer documentation facilitates smooth handover of responsibilities and assists new developers with onboarding:

- Codebase Documentation: inline comments, TypeScript typings, and organized folder structures for React components, PHP backend scripts, and Supabase configurations.
- Setup & Deployment Guide: instructions for installing dependencies, configuring

environment variables, setting up databases, building front-end, and documenting deployment pipeline.

- Version Control Notes: Git branching strategy, notes of changes made tied to specific releases, and conventions for commits.

B. Maintenance Strategy

Since the platform is intended for primary school students and teachers, and is meant to be used continually, maintenance is necessary to keep the system operational, safe, and flexible.

1. **Corrective Maintenance**
   Concerns the issues that become known after deployment. It consists of:

- Repairing broken pages, links, or other components
- Resolving incoherent databases
- Debugging issues with authentication or logging in
- Correcting errors in uploading or playing videos Each issue is managed with a simple issue management system, such as GitHub Issues or Jira.

2. Adaptive Maintenance

Changes are necessary as per the continued evolution of the school's requirements:

- Introducing new types of assignments or different formats for timetables
- Altering roles if there are new categories of admins
- Accommodating new classrooms or changes to the academic year
- Updating the UI to fit new style guidelines intended for children

3. Perfective Maintenance

This may include enhancements of performance, usability, and interface design and quality beyond what was originally requested. Below is an example of this:

- React components may be optimized so they load faster.
- Improvements may be made to the layout of the dashboards to support easier navigation by students.
- Mechanisms to compress videos may be enhanced to enable smoother playback.
- Additional features may be added such as AI-supported learning recommendations or

dashboards for parents.

4.Preventive Maintenance

The purpose of this activity is to keep the system healthy for the long term. The tasks for this include:

- Regular backups of the PostgreSQL database
- Updating dependencies (React, Tailwind, Supabase libraries, PHP versions)
- Security audits to detect vulnerabilities
- Server health monitoring
- Removing unused files, temporary uploads, or outdated records

5. Maintenance Documentation

Updates Documentation is regarded as a living document. For each significant update or enhancement of module

- The API documentation is updated.
- Screenshots and interface descriptions are updated
- Versioning notes reflect new features or changes
- Database schema updates are re-documented.

This ensures long-term traceability and smooth onboarding for future developers or student project teams.

3.6 Tools and Techniques Overview

Interactive Software Tool, having a plan was adhered to in a very disciplined manner. A number of tools and techniques were used to streamline and structure workflow throughout the life of the project. Below are the most significant tools and techniques used for the project.

1. Planning and Design Tools

- **Flowcharts:** Used to illustrate system functionality along with processes and decisions visually.
- **Block Diagrams :** These were utilized to present the system at a high level by

distinguishing important modules and their connections.

- **Mockups:** Created to illustrate the interface design and layout of the system so that the final product w as user friendly and easy to use.

2. Version Control and Iterative Development

- Version Control (Git): Used to manage and document the changes experienced in the software system at different profiles. Thus, ensuring the development team collaborates seamlessly to manage and avoid issues during their coding.
- Iterative Development: The project was developed in an incremental fashion with each module developed, tested, and the code improved. This provided an opportunity to the team to evaluate functionality at each level and improve the code quality in every step before the final integration.

3. Feedback – Driven Refinement

- **Simulated User Testing:** Teachers, students, and administrators, acting as simulated users, assessed different system modules to detect possible user interface challenges.
- **Feedback Cycle:** Continuous iterations of system and improvement were made through feedback of participants in the test to ensure refining the system to enhance user experience, accessibility, and effectiveness.

4. Systematic Transparent and Adaptive Approach

- The use of all of these instruments and tactics ensured that the analytical approach to system design remained orderly, and that every stage was methodical and scheduled.
- The approach was open, having as attention to detail. Progress was monitored, and problems were resolved as a result of scheduled iteration.
- The approach was also flexible in that the design could adapt as user needs and other project dynamics changed on the fly.

Chapter 4

Project Management

Effective time. Met the goals. Project management and supervision, in this case, was very important for the Interactive Software Tool for Primary School Students (PRESIDENCY PORTAL). Focused project management was also used in the planning, coordination, quality level, outcome, and user satisfaction of design, evaluation, and system implementation.

The activities you trained for encompass time management, risk management, resource allocation, quality management, and monitoring, all geared towards the provision of a portal that was operational and reliable for the users

4.1 Project Timeline and Phases

The project spanned a total of sixteen weeks and was divided into separate phases to allow for orderly completion. Refer to **Table 4.1** for the detailed breakdown.

Project Phases:

Milestone Highlights

- **Week 2:** Finalization of requirements and confirmation of system functionalities.
- **Week 4:** Completion of system design and interface mockups.
- **Week 8:** Prototype completion for core modules.
- **Week 10:** Chatbot integration and functional testing.
- **Week 13:** Iterative testing, stabilization and bug fixing
- **Week 16:** Complete deployment of the system and readiness for demonstration

The division of the timeline in this way allowed for the project team to monitor the progress of the project, manage the interdependent tasks, and ensure that work was completed in a logical order.

4.2 Risk Analysis and Management

As indicated in **Table 4.2,** Risk management is one of the most important aspects of project management. For this project, potential risks were assessed by their probability, effect and possible mitigation.

Table 4.1 Project Phases

Phase	**Duration (in weeks)**	**Key Activities**	**Milestone/Deliverables**
Requirement Analysis	1-2	Stakeholder interviews, surveys with teachers and students, identifying key functional needs	Requirement Specification Document
System Design	3-4	Designing system architecture, user interfaces, flowcharts, and chatbot design	System Design Document
Development (Core Modules)	5-8	Implementation of student management, attendance tracking, assignment module, timetable, and video learning	Working Prototype of Admin & Student Portals
Chatbot Development	9-10	Implementation of dual-mode chatbot	Fully Functional Chatbot
Testing & Iteration	11-13	Unit testing, integration testing and system testing	Tested & Refined Module
Deployment Preparation	14	Data validation ,offline storage testing and responsiveness testing	Ready-to-Deploy System
Final Evaluation & Documentation	15-16	Preparing final report presentation materials, demonstration setup	Complete Project Report & Demo

Table 4.2 Identified Risks

Risk	Impact	Likelihood	Mitigation Strategy
Data corruption during development	High	Medium	Regular backups of XML data; version control for code
Interface complexity for students	High	Low	Conducted usability testing and iterative improvements
Integration issues between modules	Medium	Medium	Continuous integration testing after each development sprint
Chatbot response Inaccuracies	Medium	Medium	Feedback-driven refinement and incremental testing
Time overrun	Medium	Low	Strict adherence to the timeline, regular monitoring meetings

Risk Mitigation Strategies

- **Proactive Testing:** Unit and integration tests were performed at the end of each development sprint.
- **Iterative Feedback:** Simulated user sessions with students and teachers provided valuable feedback.
- **Structured Backups:** Regular backups ensured data and code integrity.
- **Monitoring & Reporting:** Weekly progress meetings ensured early identification and resolution of risks.

The project graduated from one milestone to the other due to the efficient time and resource management.

4.3 Resource Allocation and Team Roles

While this project was primarily individually developed, due diligence in planning of time, tools and human resources was essential.

- **Time Allocation:** Tasks were broken into weekly targets, ensuring systematic development and testing.
- **Human Resource Management:** Responsibilities included:
    - **Project Planning & Design:** Requirement analysis, flowcharts, UI/UX design.
    - **Development:** Implementation of modules, integration of functionalities.
    - **Testing & Feedback:** Usability testing, bug tracking, and feature refinement.
    - **Documentation:** Compilation of reports, diagrams, and presentations.

Efficient allocation of time and responsibilities ensured that **project milestones were met consistently**.

4.4 Quality Assurance

- Quality Assurance practices were incorporated into all parts of the project to ensure all Quality Assurance practices were incorporated.
- Module Testing was done on the individual modules to check for boundaries of their functionalities.
- Integration Testing ran modules together to check for smooth cohesion of the functionality. • Usability Testing was done by representative students and teachers of the portals and was asked to freely respond on the transparency of their feedback on the design, the ident windows, and the use of the portals.
- Bug Tracking evading documented and was scheduled to be solved at overlapping iterations.
- Performance Testing was done on the system to ensure quick response on the loading screen, and that the system was able to function without Wi-Fi.

Chapter 5

Analysis and Design

5.1 Requirements

Before the design and development of the Presidency Portal - Interactive Software Tool for Primary School Students, detailed system requirements were also performed. The prime objective was to establish the key functional and non-functional needs balanced with user objectives and technical constraints. These were obtained through the author's observations, interviews with teachers, and anecdotal records from the school management.

5.1.1 Functional Requirements

These functional requirements describe the specific actions and operations the system needs to perform to achieve its goals. Ultimately, The Presidency Portal streamlines the school education system at the primary school level, both academically and in the administrator's view. The primary functional requirements are.

1. **User Authentication and Access Control**
    - Separate login interfaces for students, teachers, and administrators.
    - Role-based access ensuring that each user can perform only their permitted actions.
2. **Student Information Management**
    - Facility to add, update, and view student records.
    - Secure storage of personal, academic, and attendance details.
3. **Attendance Tracking**
    - Teachers can mark attendance daily and generate reports.
    - Students and parents can view attendance summaries.
4. **Assignment and Homework Management**
    - Teachers can upload assignments and specify deadlines.

- Students can view, download, and submit completed work.

5. **Timetable Scheduling**
   - Automated timetable display for each class and section.
   - Teachers can update subjects and timing as required.
6. **Video Learning Integration**
   - Students can access educational videos categorized by subject and grade.
   - Teachers can add YouTube links or upload their own learning materials.
7. **Interactive Chatbot Feature**
   - Provides two modes:
     - **Academic Mode** – answers school-related queries (assignments, timetable, attendance).
     - **Learn Mode** – enables students to ask questions on various topics such as math, science, and history.
   - Students can activate/deactivate the Learn Mode as they wish.
8. **Data Persistence and Backup**
   - Use of local XML or LocalStorage for retaining data across sessions.
   - Ensures minimal data loss even during network disruptions.
9. **Dashboard and Notifications**
   - Centralized dashboard for real-time updates.
   - Notifications for assignments, announcements, and attendance alerts.

5.1.2 Non-Functional Requirements

Non-functional requirements that apply to the system will affect attributes, such as user experience, system dependability, system efficiency, and system safety. Such requirements will guarantee that the system remains seamless and dependable in adaptive environments.

1. **Usability**

- The interface must be intuitive and suitable for children between the ages of 6–12.
- Use of simple language, icons, and color-coded sections for clarity.

2. **Performance**

- The system should load pages within 3 seconds under normal usage.
- It must efficiently handle multiple users accessing data simultaneously.

3. **Scalability**

- Designed to accommodate an increasing number of students, classes, and features over time.
- Easy integration with a cloud database or mobile app in the future.

4. Security

- Data encryption for sensitive user information.
- Session-based authentication and restricted data modification access.

5. **Maintainability**

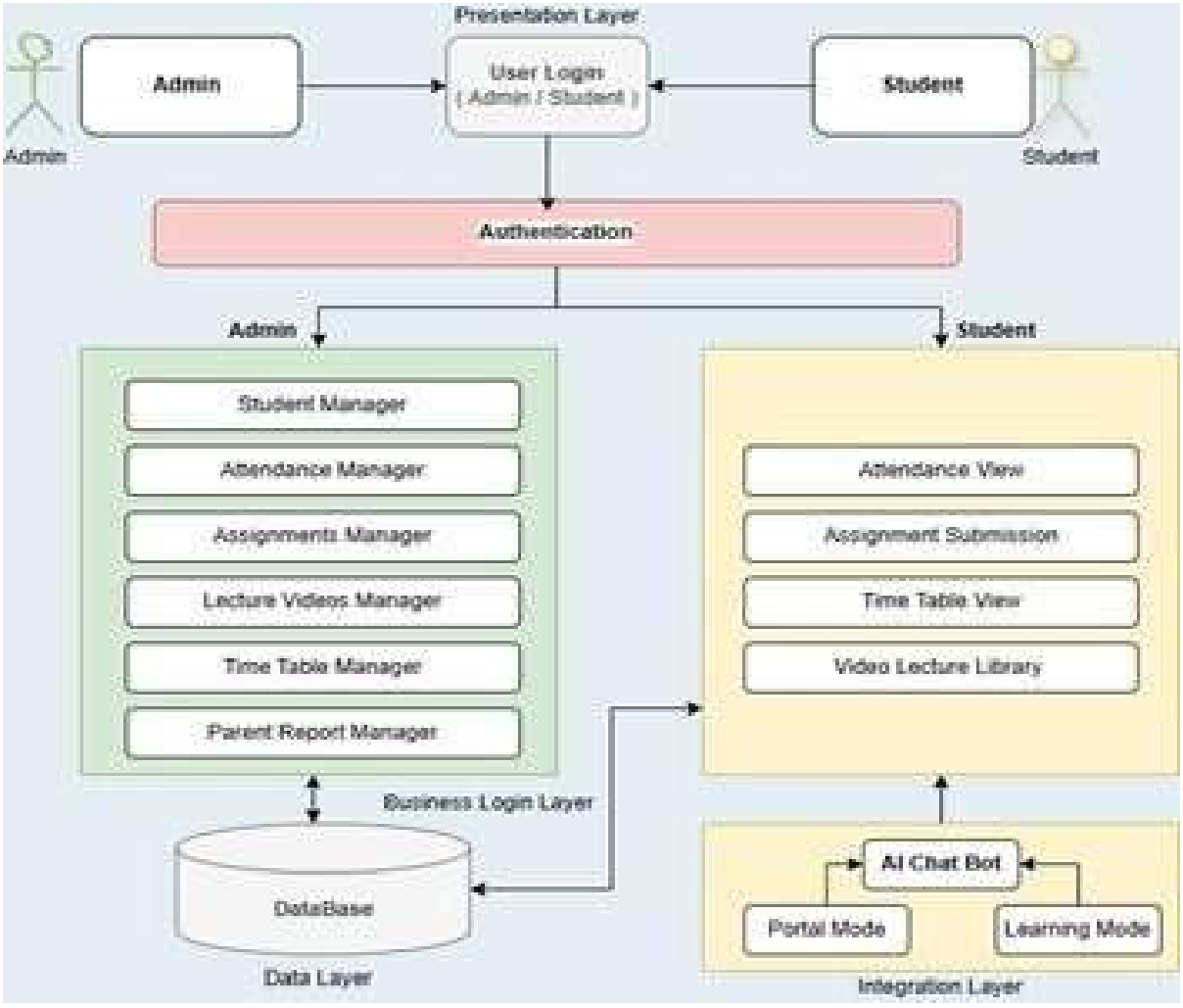
- Code structured using modular architecture for easier updates and debugging.
- Documentation available for all major modules to support long-term maintenance.

6. **Accessibility**

- Created with the intention to meet accessibility standards (contrast, font size, navigation.
- Simple enough for kids with low digital literacy.

5.2 Block Diagram

As shown in **Figure 5.2.1,** shows a block diagram that represents the different levels of hierarchy of the Presidency Portal. These are the '**Frontend Layer', 'Application Logic Layer', and 'Data Storage Laye**r'. For the frontend, the functionality is concerned with the



user interactions and the admins and students. In the application layer, we have the core logic functionalities that include, among others, the different.

Fig 5.2.1 Block Diagram

5.3 System Flow Diagram

As shown in **Figure 5.3.1,** System Flow Chart System flow chart presents the sequence of activities performed by student logging to Presidency Portal. The system initiates with user login, then routes student to service modules like attendance module, assignments module, timetable module, video and chatbot.

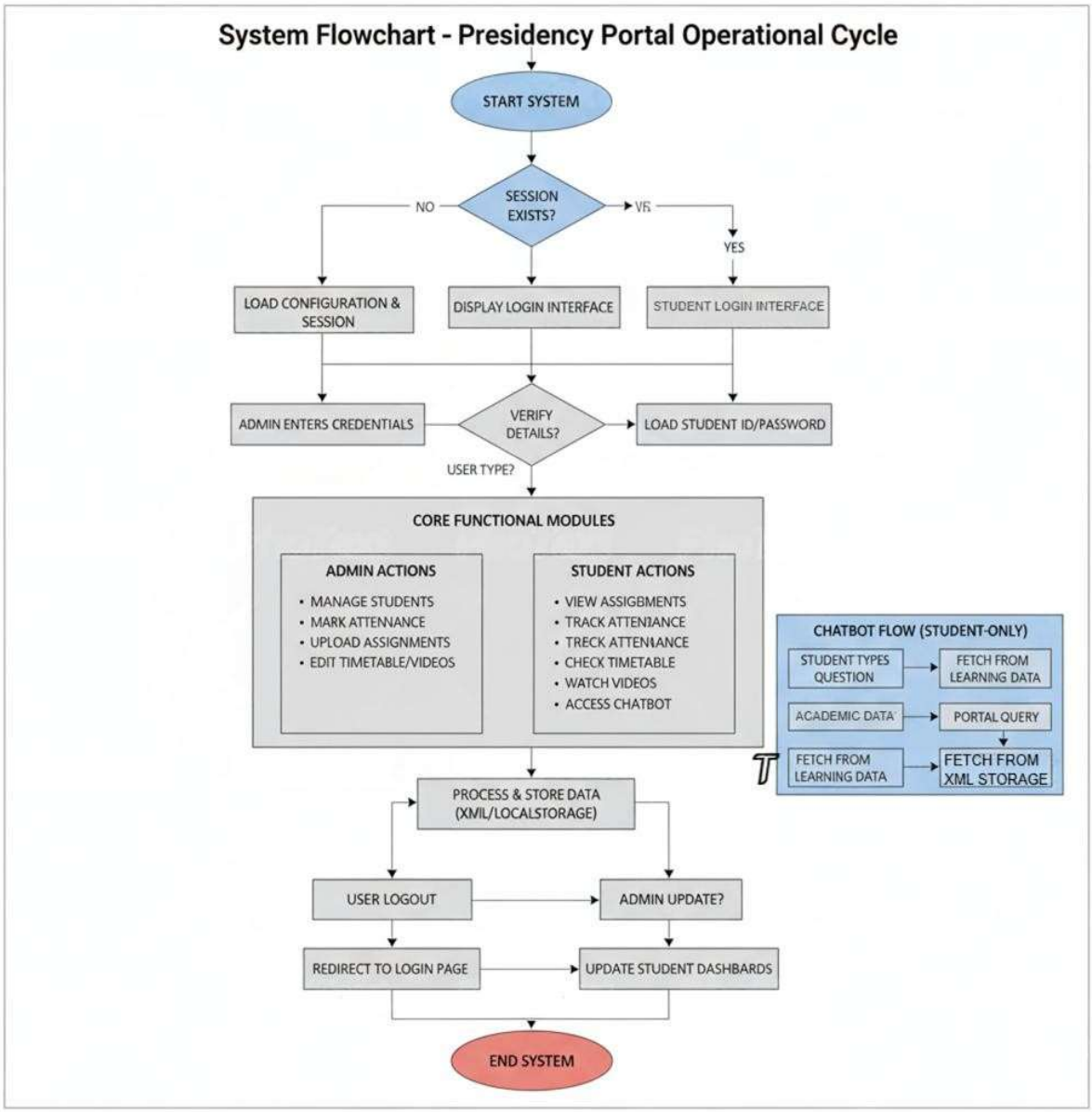


Fig 5.3.1 System Flow Chart

5.4 Designing Units

The system is componentized into **functional units** that handle different roles on **PRESIDENCY Portal**. This modular way of thinking enables the resulting solution to be highly scalable, maintainable and performant (see **Fig 5.4.1** for a visual illustration)

1. Authentication and User Management Unit

- Handles **login validation** and **role-based access** for administrators and students.
- Ensures **secure session handling** and **form validation** to prevent unauthorized access.
- Maintains user data for smooth navigation and control.

2. Student Management Unit

- Allows the administrator to **add, view, and manage student details**.
- Students are categorized **class-wise** for easy access.
- Integrates with attendance and assignment modules for consistency.

3. Attendance Tracking Unit

- Enables marking of **daily attendance** for each class.
- Stores attendance records with **date and status**.
- Provides **attendance reports** and visual feedback for students and admin.

4. Assignment Management Unit

- Admin can **upload and assign tasks** to specific classes.
- Students can **view and submit assignments** online.
- Tracks submission deadlines and maintains uploaded files.

5. Video Lecture Management Unit

- Supports **digital learning** through embedded **YouTube video links**.
- Categorizes videos by **class and subject**.

- Allows students to access lectures directly from their dashboard.

6. Timetable Management Unit

- Enables setting up and editing **class schedules**
- Features subjects, teachers and time for all the classes.

7. Chatbot and Learning Assistance Unit

Integrates an **AI-based chatbot** for student support

- Responds to academic queries and helps in quick information retrieval.
- Encourages self-learning and student interaction.

8. User Interface and Navigation Unit

- Gives you a clean UI with modern tools.
- Comes with **sidebar navigation, notifications** and **sleek transition.**

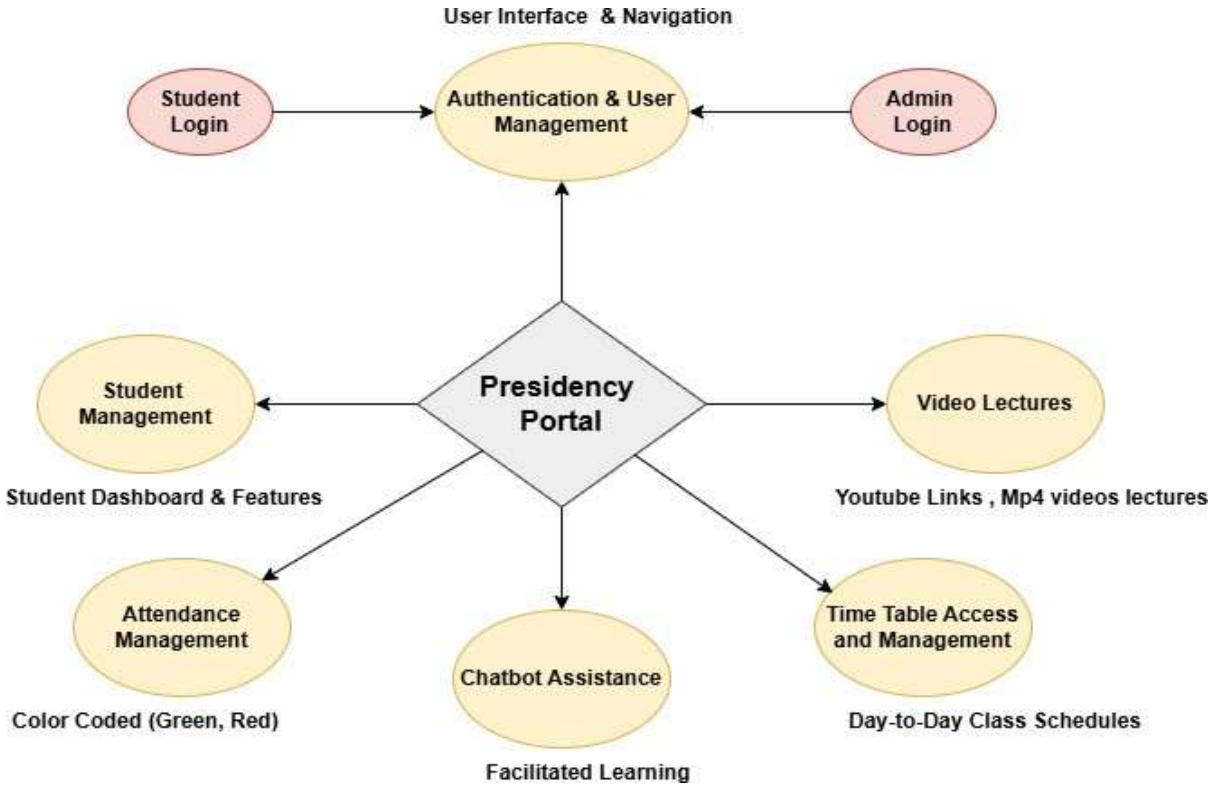


Fig 5.4.1 Designing Units

5.5 Standards

Following convention ensures **PRESIDENCY PORTAL** stays safe, both reliable and scalable. These include best practice for software development, data handling and security, interfacing with users and the Internet of Things (IoT). The explanation has been supported with **Figure 5.5.1**

5.5.1 Software Development Standards

1. The architecture of the system is at **MVC (Model-View-Controller)**, which means that data and logic are distinctly separated from presentation.
2. Java EE standards are applied here along with **Java, JSP** and **Servlets** to create a strong back end.
3. Cross browser compatible, **W3C valid HTML5** and **CSS 3** code that's responsive.

5.5.2 Policy Database and Data Management Standards

1. **MySQL** is the backend and **3NF normalization** is applied to enforce data integrity and prevent repetition.
2. **Data Relationship**: Primary and foreign keys ensure relational integrity
3. Features Data integrity and protection through ACID compliant backups and restoration.

5.5.3 Security and Privacy Standards

1. The system is engineered to avoid common security vulnerabilities such as SQL Injection and XSS by following **OWASP** best practices.
2. Sensitive data are stored securely using password hashing (SHA-256 etc.).
3. The data is transmitted securely over **HTTPS** and using **SSL/TLS encryption**.

5.5.4 User Interface and Accessibility Standards

1. The UI is designed according to **Material Design** for a simple and harmonious look.
2. Keyboard navigation and screen readers are also supported by the system making it inclusive.
3. It's a responsive theme that can be useable to any means such as on your **Laptop, Tablets** and **Mobiles.**

5.5.5 Testing and Documentation Standards

1. Functional interaction across modules is guaranteed by **unit** and **integration testing.**
2. Standards for documentation maintain well integrated functional specifications & user manuals **(fig 5.5.1).**

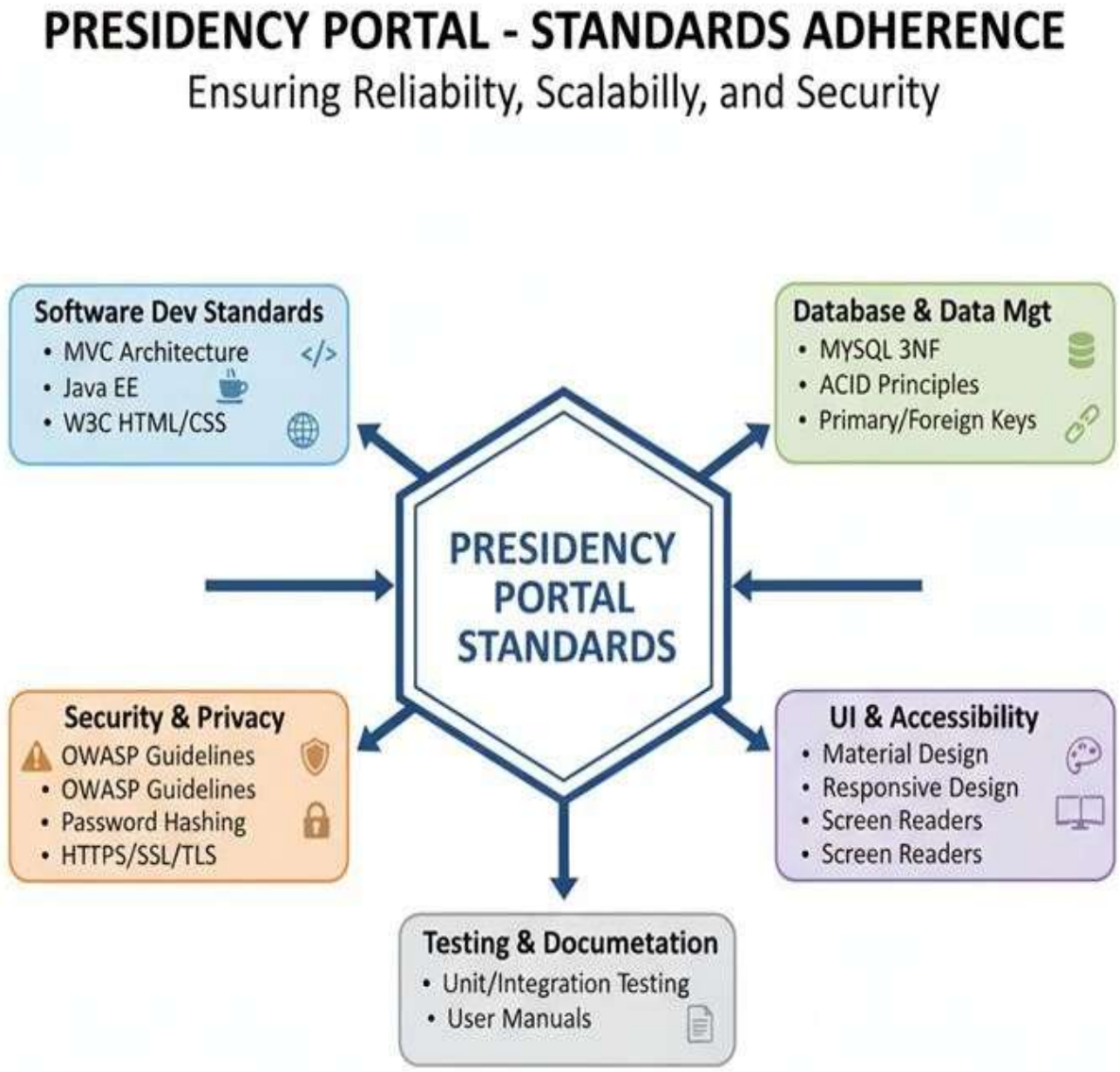


Fig 5.5.1 Standards

5.6 Domain Model Specification

Domain Model for the PRESIDENCY PORTAL The domain model **Fig 5.5** portrays key entities, attributes, and relationships within the PRESIDENCY PORTAL system. It helps to simplify the comprehension of a system and illustrates how individual elements work together.

5.6.1 Objectives of Domain Modeling

The domain model functions:

- Represent the **core functional areas** of the PRESIDENCY PORTAL.
- Identify the **key entities** and their relationships.
- Define the **attributes** of each entity to support system operations.

5.6.2 Key Entities and Their Descriptions

The domain model consists of a few important entities that describe the main features of the system:

1 Admin

- The **admin** is the central controller who manages users, data, and configurations.
- Responsibilities include managing **student information**, **assignments**, **attendance**, and **chatbot settings**.
- Attributes: **admin_id, name, email, password, role.**

2 Student

- The **student** entity represents the learners registered on the portal.
- Each student can access assignments, view attendance, and interact with the chatbot.
- Attributes**: student_id, name, class, roll_no, email, password, attendance_status.**

3 Class

- Represents an academic group or section managed within the system.
- Each class is linked to multiple students and has its own timetable and assignments.

- Attributes**: class_id, class_name, section, teacher_assigned.**

4 Attendance

- Maintains attendance records for each student on a daily basis.
- It is directly linked to both the **Student** and **Class** entities.
- Attributes: **attendance_id, student_id, date, status.**

5. Assignment

- Daily Tasks – Contents uploaded by admin, which are academically oriented and placed here.
- Assignments can be seen, downloaded and uploaded in the portal by students.
- Students can view, download, and submit assignments through the portal.
- Attributes: **assignment_id, title, description, class_id, upload_date, submission_date.**

5 Timetable

- Defines the class schedule for subjects, teachers, and timings.
- Helps students manage their academic routine efficiently.
- Attributes**: timetable_id, class_id, subject, day, time_slot.**

6 Chatbot

- A digital assistant integrated into the system to help students with queries.
- Functions in two modes: *Normal Mode* (assignments, timetable, attendance) and *Learn Mode* (subject-based learning).
- Attributes**: bot_id, mode, query_type, response_pattern.**

7 Video Lecture

- Represents learning object uploaded by admin.
- Each video falls under certain categories or topics.
- Attributes: **video_id, title, subject, class_id, link.**

5.6.3 Relationships Between Entities

The associations between these entities are seen below:

- **Class → Timetable (One-to-One**): Each class has one unique timetable.
- **Student → Chatbot (One-to-One):** Each student can interact with the chatbot individually.
- **Admin → Video Lecture (One-to-Many):** The admin can upload several video lectures for different classes.
- **Admin → Student (One-to-Many):** Each may control many students
- **Class → Student (One-to-Many):** One class has many students.
- **Student → Attendance (One -to-Many) :** There can be multiple recordings for each student.
- **Class → Assignment (One-Many):** An assignment may belong to one or more classes.
- **Student→Assignment (Many- to - Many):** A student can view/submit many assignments.

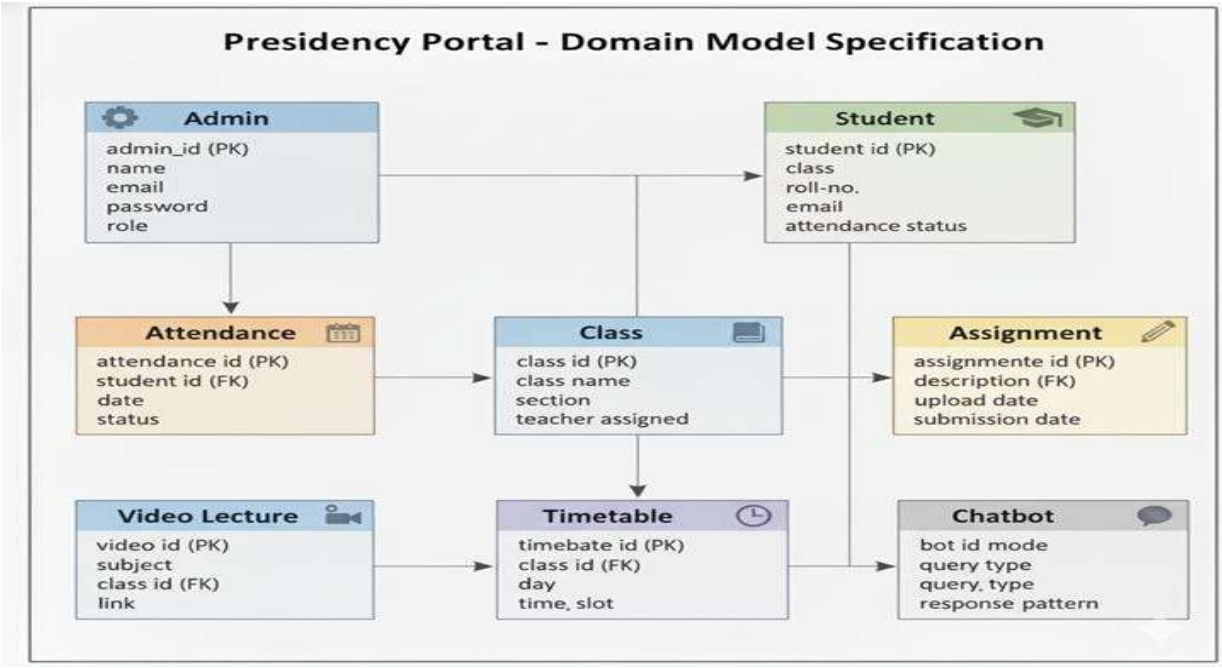


Fig 5.6.1: Domain Model Specification

5.7. Communication Model

The communication model of the PRESIDENCY PORTAL describes how data is transferred across the Admin, Student interfaces and XML based storage. It is a client-server architecture, where React-based frontend does follow by using simulated backend (localStorage + XML shape) to replicate the data lifecycle.

1. **User Interaction Layer (Frontend)**
    - Handles the user interface for both admin and student portals.
    - Manages the admin and student user interfaces. Supports data entry (e.g., register attendance,

      submit homework, dashboard, reports.)
    - Leverages React components, forms and local state to gather and display details.
2. **Processing Layer (Logic Handling)**
    - Contains all logic for request handling and response generation.
    - React Hooks and TypeScript-based functions are used to manage the flow of data between frontend components and XML storage.
3. **Data Storage Layer (XML-based Storage)**
    - Serves as the simulated backend of the system.
    - All the data (students, attendance, assignments, timetables, and videos) is stored in structured XML format inside localStorage.
    - The data is fetched and updated with JavaScript that mocks database queries.

5.7.1 Flow of Communication

Phased communication between editors within the PRESIDENCY PORTAL, as follows:

1. **User Request initiation** – Every time admin / student performs an action like uploading of assignment, checking attendance etc., front-end captures the event and creates a request object.
2. **Data Processing** – Request validation and formatting happen at the processing layer.

3. **Data Access** – The serialized request is then handed off to the XML data storage system, which retrieves or updates the appropriate record in localStorage
4. **Response Generation** – After the data operation is performed, a response (success/failure message or dataset) is formed and sent back to the frontend layer.
5. **User Feedback** – The frontend makes the UI changes respectively and displays messages.This is a loop of continuous data exchange provides the system with characteristics similar to an interactive and dynamic educational portal

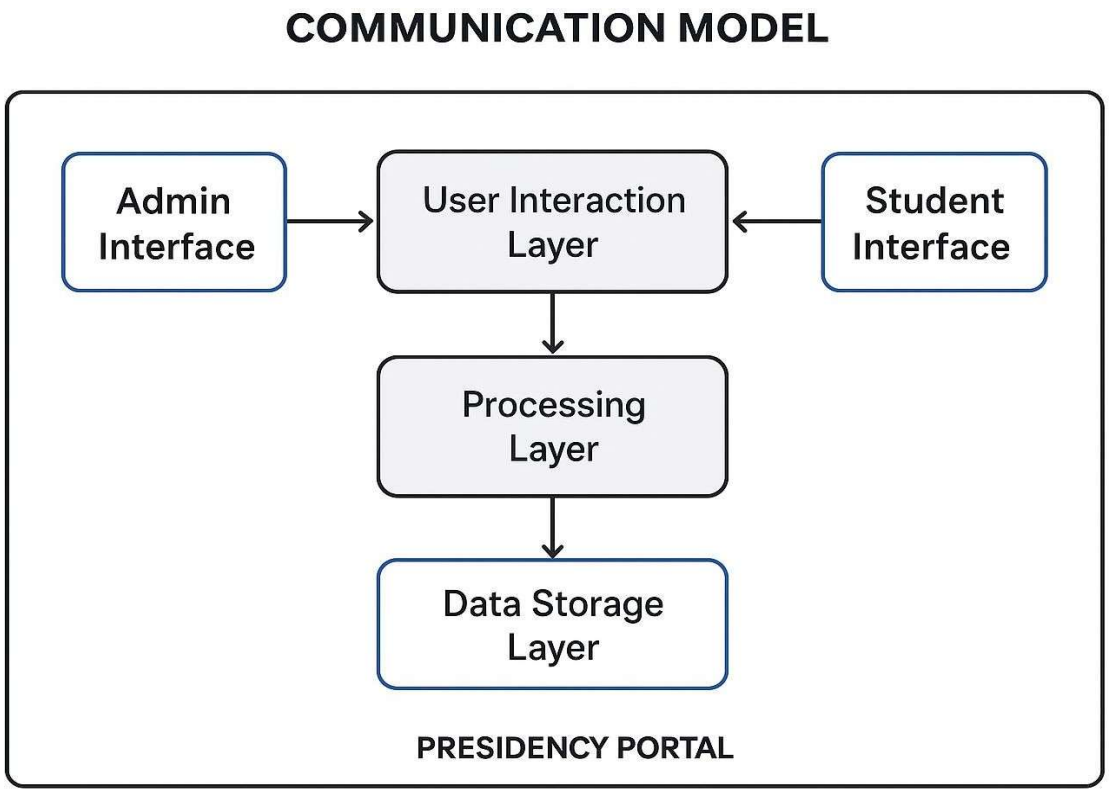


Fig 5.7.1 Communication Model

The **PRESIDENCY PORTAL** uses a strong communication model to facilitate smooth data exchange and inter-activeness between its modules and users **(Fig 5.7.1).**

5.7.2 Communication Characteristics

The communication part of this system has the following properties:

- **Bidirectional Data Flow:** Data can move from admin to student and vice versa.
- **Asynchronous Operations:** Non-blocking updates ensure smooth user experience.
- **Event-driven Triggers:** User actions such as clicks or submissions trigger communication events.

5.7.3 Communication Modes

Two main user aware communication modes are supported in the system:

1. **Admin Communication Mode:**
    - The admin is responsible for using the portal to create and manage all academic activities like Upload Assignments, Add Videos and Mark Attendance
    - The students are engaged mostly to watch or submit information.
2. **Student Communication Mode:**
    - Students interact primarily for viewing or submitting information.
    - The portal pulls information from the XML storage and lays out on the dashboard providing tailored content.

Furthermore, the **chatbot** works as a live **UI (user interface)**, delivering active interactions between users and system logic. It processes natural questions and retrieves appropriate answers, interfaces with archived academic data

5.8 Functional View

The functional perspective of the system, depicted in **Fig 5.8.1,** showcases how various software components and modules work together to produce required services in line with the **PRESIDENCY PORTA**L project's aim.

5.8.1. Overview

It is aimed at being a one-stop online management system for academic and administrative operations. Features It comprises numerous extended functionalities such as student management, attendance tracking, assignment management, viewing timetable and smart chatbots for providing student support. The site comprises two main interfaces - the **Admin Portal** and **Student Portal** — can be viewed in the **fig 5.8.1,** each serving specific purpose and exchanging data through an integrated XML-format storage system.

1. Key Functional Modules

- **Admin Module**
    - It manages all backend activities including handling of student records, marking attendance, submitting assignments and generating timetable.
    - Control of video lecture upload and resource allocation for multiple classes.
- **Student Module**
    - Allows students to view assignments, check attendance, submit work, view timetables, and access educational videos.
    - Displays a personalized dashboard with attendance percentage, pending assignments, and important updates.
- **Data Layer (XML Storage System)**
    - Functions as the core data repository using structured XML stored in the browser's local storage.
    - Keeping all the parts (students, attendance records, assignments, schedules and video links) in check, making sure they are always available as well as robust.

5.8.2. Functional Flow

- **Input Phase:** Admin enters or edits data related to student (assignments, attendance, time table and videos)
- **Processing Phase:** The storage mechanism is the XML based,it stores and maintains structured records for all classes,students

- **Output Stage:** Students explore their individual dashboards to see and play with the data, while the chatbot responds immediately or offers help in learning.

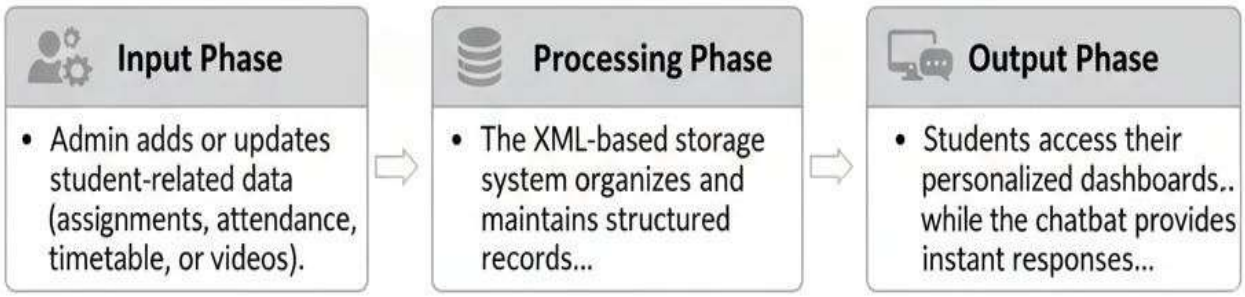


Fig 5.8.1 Functional View

5.9 Operational View

This chapter discusses the operation of the presidency portal in real time as users interact with the system **fig 5.9.1** below shows how internally to respond to user interaction here are a number of financial and administrative interactions available through this system.

1. User Roles & Access

- **Students**
    - Login via web interface
    - Link features: attendance, timetable, results, chatbot
- **Administrators/Faculty**
    - Manage student records
    - Update courses, announcements
    - Train chatbot modules

2. Authentication Flow

- Users authenticate through login page
- Credentials verified against centralized storage
- On successful login feature access based on roles is get provided

3. User Interactions & System Responses

- **Data retrieval**
- **Data update**
- **Validation checks** (e.g., login, data input)

Real-time feedback is provided via the user interface

4. Backend Operations

- Ensures data consistency and synchronization
- Manages business logic and data communication
- Communicates with XML-based storage/localStorage

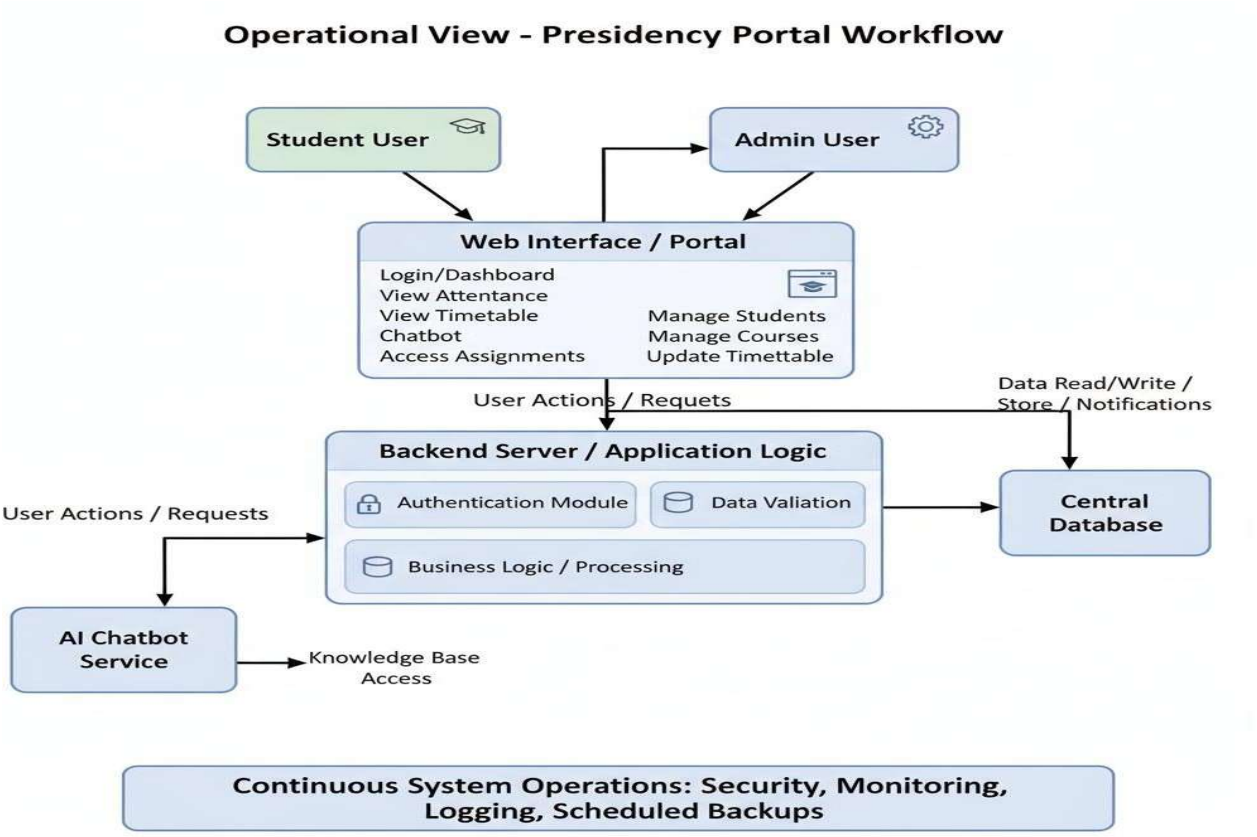


Fig 5.9.1 Operational View

5.10 Other Design

Additional design aspects for improved usability, performance and future expandability of the Presidency Portal are presented in details in **Fig 5.10.1.**

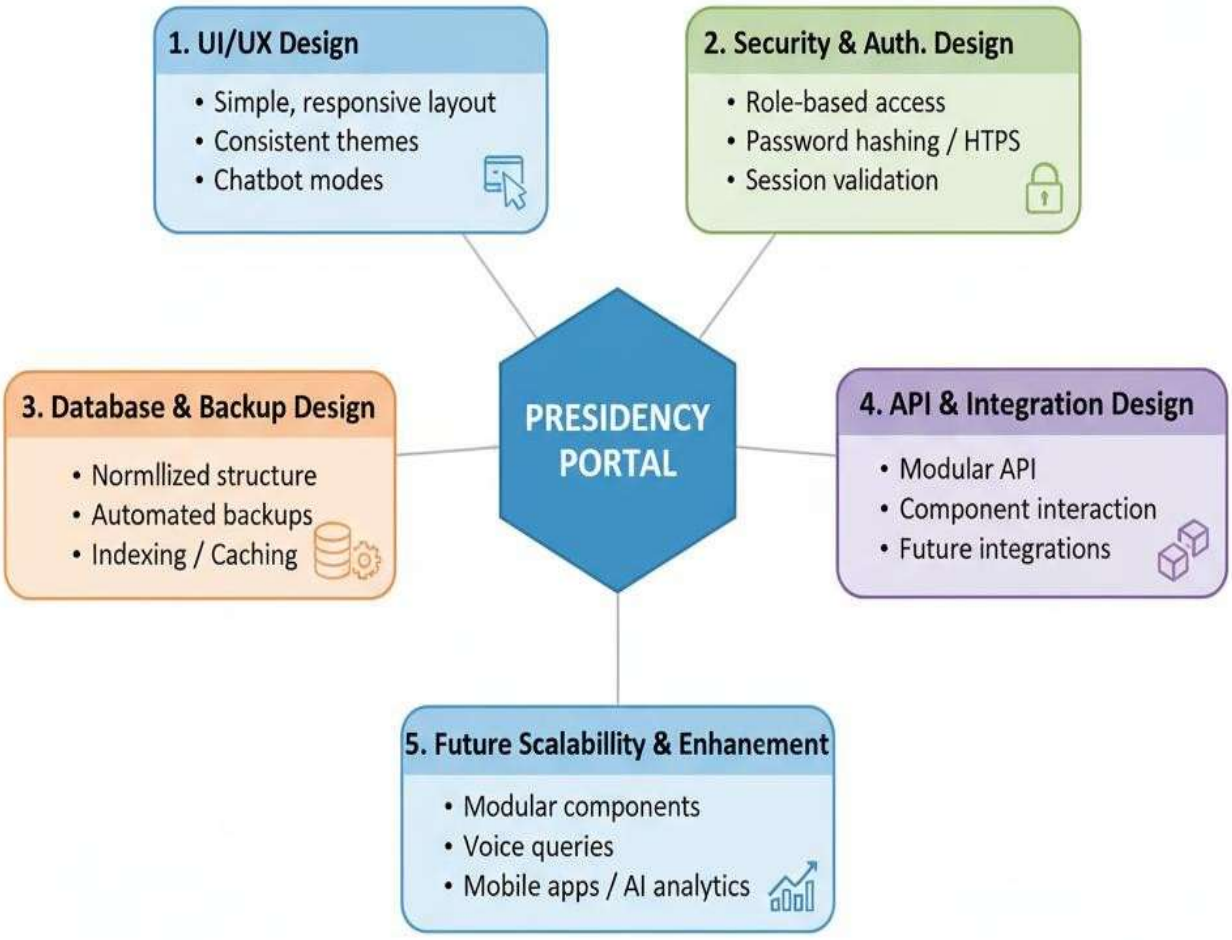


Fig 5.10.1 Other Design

Chapter 6

Hardware And Software

This chapter presents a complete description of the hardware and software environment when developing the **Interactive Software Tool for Primary School Students (PRESIDENCY PORTAL).** Definition of the simulation process that has been applied in order to validate functionality, check data integrity and assure system's reliability prior to its implementation is provided as well.

6.1 Hardware Requirements

As shown in the **Table 6.1,** the project is primarily **web-based**, it was necessary to adhere to some **hardware requirements** for smooth development, testing and use of features such as video playback and integration with chatbots.

6.1.1 Development Environment

Table 6.1.1: Development Environment

Component	Specification	Purpose
Processor	Intel Core i5 or higher	Fast compilation, smooth multitasking during development
RAM	8 GB or higher	Sufficient memory for running development servers, browser tabs, and testing tools
Storage	20 MB free space	For localStorage and temporary data caching
Graphics	Integrated or dedicated GPU	Required for smooth UI rendering and video playback in the Portal
Display	Full HD 1920x1080	Accurate layout testing for responsive design
Peripherals	Keyboard, Mouse	Basic input devices for development and testing

6.2 Software Development Tools

The **webdev** technologies enable to develop a strong and scalable portal which is also user-friendly. These tools facilitated modular design, testing & seamless integration of interactive

features shown in **Table fig 6.2.1**

6.2.1 Frontend Tools

Table 6.2.1 Frontend Tools

Tool/Technology	**Version/Details**	**Purpose**
React	18.3.1	Component-based UI development, reactivity for dynamic dashboards
TypeScript	5.x	Type safety for reducing runtime errors
Vite	Latest	Fast development server and build tool
Tailwind CSS	Latest	Utility-first framework for responsive and modern design
Radix UI + shadow	Latest	Accessible and reusable UI components

6.2.2 State Management & Data Handling

Table 6.2.2: State Management and Data Handling

Tool/Technology	**Purpose**
React Hooks	Local state management (useState, useEffect)
React Query	Server-state simulation and caching
LocalStorage / XML Storage	Persistent offline storage for student records, attendance, assignments, and videos

6.2.3 Form Handling and Validation

Table 6.2.3: Form Handling and Validation

Tool/Technology	**Purpose**
React Hook Form	Efficient form management for assignment submission and attendance marking
Zod	Schema-based validation to ensure accurate user inputs

6.2.4 User Interaction and Feedback

Table 6.2.4: User Interaction and Feedback

Tool/Technology	Purpose
Sonner + Radix Toast	Real-time notifications for successful actions (e.g., assignment submission)
Lucide React	Educational and intuitive iconography for better navigation
Custom Animations	Smooth transitions and user-friendly interactions

6.2.5 Development Workflow

- **Version Control**: Git was used to track code changes and maintain history.
- **Testing Tools**: Browser developer tools for debugging and inspecting elements.
- **Build & Deployment**: Vite for fast builds and live server previews.
- **Documentation Tools**: Markdown and flipchart tools for flowcharts and ERDs. This soft was

environment provided efficient development, maintainability and high performance.

6.3 Representation

Simulation is important for **verifying system functionality, testing data handling** and **performance** under realistic scenarios. Because the system stores **XML offline,** simulation essentially revolved around data quality, module interactions and user's interactivity. The complete overview is shown in **Table 6.3.1**

6.3.1 Data Representation

- **Student Data**: 140 students pre-populated with systematic IDs.
- **Attendance Data**: Presence records – you can test reporting features on this: Randomized entries with present/absent.

- **Assignments & Submissions**: Mock your uploads and submit results to test dashboards.
- **Timetable & Video Information**: Preprocessed weekly timetables and video items of each course.

Table 6.3.1 : Representation Of Attendance Data

Student	Class	Date	Status
20250101	1	2025-10-01	Present
20250102	1	2025-10-01	Present
20250103	1	2025-10-01	Present

6.3.2 Module Representation

1. **Admin Module**:
    - Added, updated, and deleted student records.
    - Marked attendance and verified dashboard statistics.
    - Uploaded assignments and monitored submission status.
    - Added video lectures and tested playback.
2. **Student Module**:
    - Viewed personal dashboard and statistics.
    - Submitted assignments (PDF) and checked deadlines.
    - Monitored personal attendance history and timetable.

6.3.3 Testing Workflows

- **Workflow 1 – Attendance:** Admin takes attendance → System is up to date with it in dashboard → Student can see the latest attendance → Consistent verified data.
- **Workflow 2 - Assignment Start:** Admin uploads assignment → Student submits PDF → System
    - verifies submission → Dashboard reflects completion.

- **Workflow 3 – Chatbot Interaction**: Student asks about timetable or Learn Mode → System gives the true correct answer in context → Feedback written for improvement.

These simulations check the robustness of the system, data integrity and accuracy to interaction prior to real usage.

6.3.4 Performance Evaluation

- Tested **portal loading times** for 140+ student records.
- Verified **responsiveness** on different screen sizes (desktop, tablet, mobile).
- Assessed **video playback** speed and chatbot response times.
- Ensured **offline functionality** via localStorage even without internet (except for YouTube videos).

Chapter 7
Evaluation and Results

Evaluation and testing Evaluation and testing is need to carried out for the **PRESIDENCY PORTAL** to fulfil its **functional, usability** and **performance requirements**. Test points, principles of testing and results will be provided in the chapter validating the system as to its **reliability, accuracy** and **user friendly-ness.**

7.1 Test Points

The testing covered the following priority areas to ensure the portal operates effectively and stably:

1. Student Management

- Adding, updating, and deleting student records
- Proper generation of student IDs (format: YYYYMMDD-XXXXXX)
- Accurate class allocation and roll number assignment

2. Attendance Module

- Daily marking of attendance by class
- Calculation of attendance percentages
- Display of color-coded attendance indicators on dashboards

3. Assignment Module

- Assignment creation with deadlines
- PDF submission validation
- Submission status tracking and reporting

4. Timetable Module

- Creation and update of weekly class schedules
- Accurate display of daily schedules for students

- Subject and teacher allocation verification

5. Video Lecture Module

- Lecture Module o uploading and organizing of YouTube links
- E Student portal and video embed and playback
- Class and subject-specific filtering

6. Chatbot Functionality

- Academic Query Mode: answering assignments, timetable, attendance questions
- Learn Mode: answering general educational queries across subjects
- Mode switching and enable/disable functionality
- Response accuracy and contextual relevance

7. Data Persistence & Offline Functionality

- Verification of XML storage for student data, attendance, assignments, and videos
- Testing portal behaviour during page refresh or browser restart
- Ensuring offline functionality where possible

8. User Interface & Responsiveness

- Navigation across modules
- Layout consistency on desktop, tablet, and mobile
- Visual feedback (toasts, progress bars, status indicators)

7.2 Test Plan

The test design was created with the goal to perform **functional, usability** and **performance tests** that addresses both administrators' and students' requirements.

7.2.1 Testing Strategy

- **Functional Testing:** Check whether all the modules working properly and correctly
- **Performance test:** how much time it takr to load a page Responsiveness Offline functionality

- **Integration Testing:** All integrated modules run perfectly, from the chatbot operations

7.2.2 Test Methodology

1. **Module Testing:** Each module was tested separately for the individual functionality.
2. **Testing workflow end-to-end:** tracking admin actions (attendance, uploading assignments) to their completion on the student dashboards.
3. **Mock User Interactions:** Random student logins were emulated to verify chatbot replies, assignment submissions and timetable visibility.
4. **Data Persistent Testing**: Changes were tested between logouts and page refreshing to ensure offline

7.3 Test Results

Observations produced from the testing of all modules are summarized and presented in tables below.

7.3.1 Student Management

Table 7.3.1: Student Module

Test Case	Expected Result	Observed Result	Status
Add new student	Student record adds with unique ID	Successfully added 140+ students	Pass
Update student info	Updated details reflected n Portal	Verified for 100 records	Pass
Delete student	Student removed from system	Deleted records removed from Dashboards	Pass

Observation: Student ID generation was consistent and sequential; class and roll assignments were accurate.

7.3.2 Attendance Module

Table 7.3.2: Attendance Module

Test Case	Expected Result	Observed Result	Status
Mark attendance	Status saved and dashboard updated	Confirmed accurate for all students	**Pass**
Calculate %	Correct percentage displayed	Verified with test data	**Pass**
Color coding	Green for >75%, Yellow 50–75%, Red <50%	Functioned correctly	**Pass**

Observation: Attendance dashboard is intuitive, with obvious visual response

7.3.3 Assignment Module

Table 7.3.3: Assignment Module

Test Case	Expected Result	Observed Result	Status
Upload assignment	Visible to assigned class	Successfully uploaded	Pass
Submit PDF	Submission accepted and recorded	Verified for multiple students	Pass
Submission status	Pending/completed reflected	Correctly updated in dashboards	Pass

Observation: Deadlines were strictly enforced and status tracking went effortlessly

7.3.4 Timetable Schedule & Video Modules

- **Schedule:** Students were able to check their schedules for the week and see daily highlights; teacher information displayed properly.
- **Videos:** Class-specific videos streamed smoothly; embedded You Tube clips.

7.3.5 Chatbot Module

Table 7.3.4: Chatbot Module

Mode	Test Case	Observed Result	Status
Academic	Ask about assignments	Correct assignment info provided	**Pass**
Academic	Ask timetable	Accurate class schedule returned	**Pass**
Learn Mode	Ask general questions	Relevant educationa response Returned	**Pass**
Mode switching	Enable/Disable mode	Mode changed and responses adjusted	**Pass**

Observation: Contextual responses were observed in Chatbot and mode-switching was successfully. performed without any problem.

7.3.6 Data Persistence

Table 7.3.5: Data Persistence

Test Case	Expected Result	Observed Result	Status
Logout/Login	Data persisted	Verified for all modules	Pass
Page refresh	Data maintained	LocalStorage worked as intended	Pass

Observation: XML-backed localStorage successfully specialised a database at offline allowing it.

7.3.7 User Interface and Responsiveness

- All modules were also cross-tested on several devices.
- There were some visually pleasing design and layout.
- Moving from dashboards to modules to chatbot was easy and seamless.

7.4 Insights and Analysis

Key Findings from the PRESIDENCY PORTAL There were multiple important learnings drawn from evaluation of the PRESIDENCY PORTAL:

- **System Usability:** Modules worked as designed and with low errors; data integrity was consistent across sessions and refreshes.
- **User Experience:** Clear Dashboards, colour coded signalling, user friendly Use of Space in the screens have helped to make things simple clear for the students being a driving engagement form them.
- **Interactive Learning:** The chatbot can accept both academic and non-academic question, which proved again the benefit of dual-mode learning.
- **Offline and LocalStorage encapsulation:** since the user could have internet connection issues, site are persistence using LocalStorage.
- **Scalable:** The modular architecture and organized data storage will help incorporate features of parent portal, mobile apps or more analytics at a later time.
- **Performance**: Load times reduced, both dashboards and videos loaded well under 140 student records

.

Chapter 8
Social, Legal, Ethical, Sustainability, and Safety Aspects

It is not only technology driven, but it takes into account the impact on society, regulation of laws and ethics to sustainability and security which are reflected when we use the **PRESIDENCY PORTAL**. This chapter brings these into focus in order to make the portal consistent with contemporary educational and societal logic.

8.1 Social Aspects

The mission of the **PRESIDENCY PORTAL** is to make desired impact in basic education that will promote

access, learning and **performance**. Key social considerations include:

1. **Enhanced Accessibility**
    - Creates a level playing field for student accessibility to assignments, attendance information and parent contact information.
    - Contributes to closing the digital gap thanks to the possibility of making use of data off-line and storage in XML.
2. **Student Engagement and Motivation**
    - Interactive dashboards and video learning increase student participation.
    - Chatbot Learn Mode encourages curiosity and self-directed learning, improving overall learning outcomes.
3. **Teacher and Parent Support**
    - Admin modules simplify administrative tasks, allowing teachers to focus more on pedagogy.
    - The portal's structured design can be extended for parent access, promoting community involvement.
4. **Inclusivity and Usability**
    - Simple and intuitive interface ensures that children from age 6–12 can use it effectively.

- Visual indicators and color-coded dashboards assist students with varying literacy & cognitive skills.

Observation: There exists a more inclusive and participative environment in the educational program by offering a central and easy-access platform like the portal.

8.2 Legal Aspects

When building an e-learning platform, Law is no exception as you wouldn't want to infringe on data privacy,

copyright and information distribution.

1. **Data Privacy**
   - Student records, attendance, and assignment submissions are sensitive data.
   - Use of **localStorage/XML storage** ensures minimal exposure to the internet, reducing legal liability.
   - Future deployment with a backend database is recommended to comply with local education data protection laws
2. **Intellectual Property (IP)**
   - Videos used in the portal are embedded YouTube content, respecting copyright.
   - Assignments, resources, and other educational content developed internally are properly attributed to avoid infringement.
3. **Regulatory Compliance**
   - Gives complies with digital learning standards and government requirements for primary education.
   - Additionally, any future integration with cloud storage or through an external server will have to maintain compliance with data protection law.

8.3 Ethical Aspects

An ethical implementation would change to make it certain the portal is reliable, fair and valuable for all users.

Ethical considerations include:

1. **Student Welfare**
    - Data collection is minimal and only used for academic purposes.
    - No profiling or sharing of personal information occurs without consent.
2. **Transparency**
    - Clear communication about chatbot capabilities, Learn Mode, and data usage is provided.
    - Students and teachers are informed of how the portal operates and what data is stored.
3. **Academic Integrity**
    - Assignments and submission tracking ensure **authentic evaluation** of student work.
    - Chatbot Learn Mode encourages learning but does not directly provide answers to graded assignments unless permitted.

Observation: Ethical behaviour leads to trust, fairness and good for training.

8.4 Sustainability Aspects

Sustainability is about ensuring **long-term usability, low resource usage** and **environmentally-decent construction.**

1. **Resource Efficiency**
    - Lightweight portal design reduces system requirements and energy consumption.
    - LocalStorage reduces the need for server communication, lowering electricity.
2. **Future-Proof Design**
    - Modular architecture allows future upgrades without major redevelopment.
    - Reusable components, TypeScript type safety, and documented code enhance maintainability.

3. **Digital Literacy and Green Practices**
    - Encourages **digital learning** rather than paper-based resources, reducing paper usage.
    - Provides an eco-friendly alternative for schools transitioning.

Observation: The portal promotes **educational sustainability** and **responsible technology** use by focusing on efficiency and reusability.

8.5 Safety Aspects

Anyone making kids' software knows that safety is a vital consideration. The portal has features and protocols to support a safe online experience.

1. **Cyber Safety**
    - No personal student data is uploaded online, reducing the risk of cyber threats.
    - Chatbot interactions are limited to educational content, avoiding exposure to inappropriate material.
2. **Secure File Uploads**
    - Only PDF files are allowed for assignment submissions.
    - Input validation prevents malicious code execution or script injection.
3. **User Authentication and Session Management**
    - Secure login credentials for admin and students.
    - Sessions are managed through localStorage/sessionStorage to prevent unauthorized access.

Chapter 9
Conclusion

The Interactive Software Tool for Primary School Students (PRESIDENCY PORTAL) is an Integrated Interactive Software Package for primary school, which is an all-encompassing modern and scalable solution for academics in primary schools. The platform effectively combines student **information management, attendance monitoring, assignment submission, timetable scheduling** along with **video learni**ng and **automated chatbot** support for interactive education.

Through this project, several objectives were achieved:

- **Streamlined Administration:** Teachers and administrators can efficiently manage student data, monitor attendance, and organize class resources.
- **Ease of Admin:** From student data, attendance tracking and one place for class assets, teachers & admins can do it all easily.
- **Improved Student Learning:** Personalized dashboards, real-time assignment tracking and student videos empower learners to take charge of their own learning and become self-motivated.
- **Interactive Chatbot Integration:** The chatbot in two modes facilitates both academic and generic learning, encouraging curiosity and self-study.
- **Data persistence and robustness:** LocalStorage powered (using xml.js) we keep critical academic data across sessions/page refreshes (your paper doesn't just vanish while marking roster), with some implementations supporting offline where internet access is a premium.
- **Pro UI/UX Work:** Modern/Organic/Responsive design to hit children of any age with familiarity and clarity.

References

[1] H. Al-Fraihat, M. Joy, R. Masa'deh, and J. Sinclair, "Evaluating e-learning systems success: An empirical study," Computers in Human Behavior, vol. 102, pp. 67–86, 2020.

[2] S. Al-Sharhan, "A proposed framework for a web-based learning management system using modern web technologies," Education and Information Technologies, vol. 25, no. 4, pp. 2979–2997, 2020.

[3] F. Martin, M. Parker, and D. Deale, "Examining student behavior in online learning: Student perceptions of the LMS and learning," Online Learning Journal, vol. 22, no. 3, pp. 1–21, 2018.

[4] N. Islam, "Investigating e-learning system usage outcomes in the university context," Computers & Education, vol. 69, pp. 387–399, 2013.

[5] P. Sun, R. Tsai, G. Finger, Y. Chen, and D. Yeh, "What drives a successful e-learning? An empirical investigation of the critical factors influencing learner satisfaction," Computers & Education, vol. 50, no. 4, pp. 1183–1202, 2008.

[6] K. Hirsh-Pasek et al., "Putting education in 'educational' apps: Lessons from the science of learning," Psychological Science in the Public Interest, vol. 16, no. 1, pp. 3–34, 2015.

[7] N. Zaranis and V. Oikonomidis, "Digital tools in primary mathematics education: A study of their educational effectiveness," Education and Information Technologies, vol. 23, no. 6, pp. 2493–2515, 2018.

[8] E. Fokides, "Digital storytelling in primary education: The effect of student characteristics on learning outcomes," Journal of Educational Computing Research, vol. 54, no. 4, pp. 483–517, 2017.

[9] S. Papert, The Children's Machine: Rethinking School in the Age of the Computer. New York, NY, USA: Basic Books, 1993.

[10] Google Developers, Progressive Web Apps: The Complete Guide, Google, Mountain View, CA, USA, 2021. [Online]. Available: https://developers.google.com

[11] A. Russell, "Service workers: An introduction," W3C Web Fundamentals, Google/Chrome Dev Summit, 2016. [Online]. Available: https://developers.google.com/web

[12] Mozilla Developer Network (MDN), IndexedDB API Documentation, Mozilla Foundation, 2023. [Online]. Available: https://developer.mozilla.org

[13] Microsoft, TypeScript Language Specification: Handbook and Documentation, Microsoft Corp., 2021. [Online]. Available: https://www.typescriptlang.org

[14] Meta/Facebook, React: Official Documentation, Meta Platforms Inc., 2024. [Online]. Available: https://react.dev

[15] R. Winkler and M. Söllner, "Unleashing the potential of chatbots in education: A state-of-the-art analysis," in Proc. 38th Int. Conf. Information Systems (ICIS), 2018, pp. 1–17.

[16] A. Kerly, R. Ellis, and S. Bull, "Conversational agents in learning: A review of the field," International Journal of Artificial Intelligence in Education, vol. 18, no. 1, pp. 1–25, 2008.

[17] F. Ahmad, J. Malik, and B. Jan, "A review of intelligent tutoring systems using machine learning," International Journal of Advanced Computer Science and Applications, vol. 11, no. 2, pp. 541–548, 2020.

[18] L. Fryer, H. Nakao, and G. Thompson, "Chatbot learning partners: Connecting social presence and cognitive load in children's interactions," Journal of Computer Assisted Learning, vol. 35, no. 5, pp. 689–702, 2019

Base Paper

1. Google Developers (PWA Guide) — Offline First Architecture

Reference :

[10] Google Developers, *Progressive Web Apps: The Complete Guide*, Google, 2021. This reference is highly relevant because the proposed LMS was intentionally designed to be lightweight, fast, and capable of operating in low-connectivity school environments. The Google Developers PWA Guide provides an authoritative explanation of offline-first architecture, service workers, client-side caching, and background sync, all of which directly influence the architectural decisions in our system. Many primary schools—especially government and rural schools—do not have reliable high-speed internet. By following PWA principles outlined in this reference, the LMS ensures that attendance management, timetable viewing, assignment submission, and video access remain functional even when internet connectivity is poor. This supports the core project requirement of building an accessible and device-agnostic LMS. The reference therefore justifies why technologies like IndexedDB, LocalStorage, and asynchronous event handling were selected for our implementation.

2. Hirsh-Pasek et al. — Child-Centred Design & UI Justification

Reference:

[6] K. Hirsh-Pasek et al., "Putting education in 'educational' apps: Lessons from the science of learning,"

Psychological Science in the Public Interest, 2015. This paper provides foundational evidence for designing digital tools targeted at children aged 6–12. It explains that young learners require interfaces that minimize cognitive load through **flat navigation**, **large touch targets**, **visual cues**, **minimal text**, and **immediate feedback**. These principles directly influenced the design of the LMS's student dashboard, which uses large color-coded icons, simplified menu structures, and animated feedback to maintain engagement. The reference strengthens the academic justification behind building an LMS tailored specifically for primary school students rather than reusing higher-education methods. By applying the guidelines from this paper, the project ensures that the LMS is developmentally appropriate, easier to use, and more engaging for

children—improving both usability and learning outcomes.

3. Winkler & Sollner — AI Chatbot Integration Justification Reference:

[15] R. Winkler and M. Söllner, “Unleashing the potential of chatbots in education: A state-of-the-art analysis,”2017 -2018

This reference is essential for grounding the dual-mode AI chatbot component of the LMS in established educational research. Winkler & Söllner analyze how chatbots support learning environments by reducing teacher workload, providing instant feedback, and increasing student engagement. Their work highlights that chatbots can improve both administrative efficiency (e.g., answering routine queries) and pedagogical support (e.g., offering simplified explanations). These findings directly justify the inclusion of two distinct chatbot modes—**Portal Mode** and **Study Mode**—in our system. Portal Mode follows their recommended approach for automating repetitive tasks like attendance queries and deadline reminders, while Study Mode provides age-appropriate academic explanations. This reference therefore provides strong academic backing for using conversational AI in a primary-school LMS and supports the claim that the chatbot enhances student autonomy and participation.

Appendix

Appendix A: System Screens and Interface Description

A.1 Login Page

- Dual login interface for both Administrator and Student.
- Validates credentials using localStorage-based authentication.

A.2 Dashboard

- Displays total number of students, attendance summary, pending assignments, and uploaded videos.

A.3 Student Dashboard

- shows personalized welcome message, attendance percentage, and number of assignments.

A.4 Attendance Module

- Class-wise attendance marking interface.
- Present/Absent toggle with auto-save functionality.

A.5 Assignment Upload Page

- Admin can upload assignment title, description, class, and deadline.
- Displays real-time upload confirmation.

A.6 Assignment Submission Page (Student)

- Allows PDF upload.
- Shows submission status and due date.

A.7 Timetable Module

- Weekly schedule mapped class-wise.
- Editable by admin; auto-view for students.

Appendix B: System Screenshots

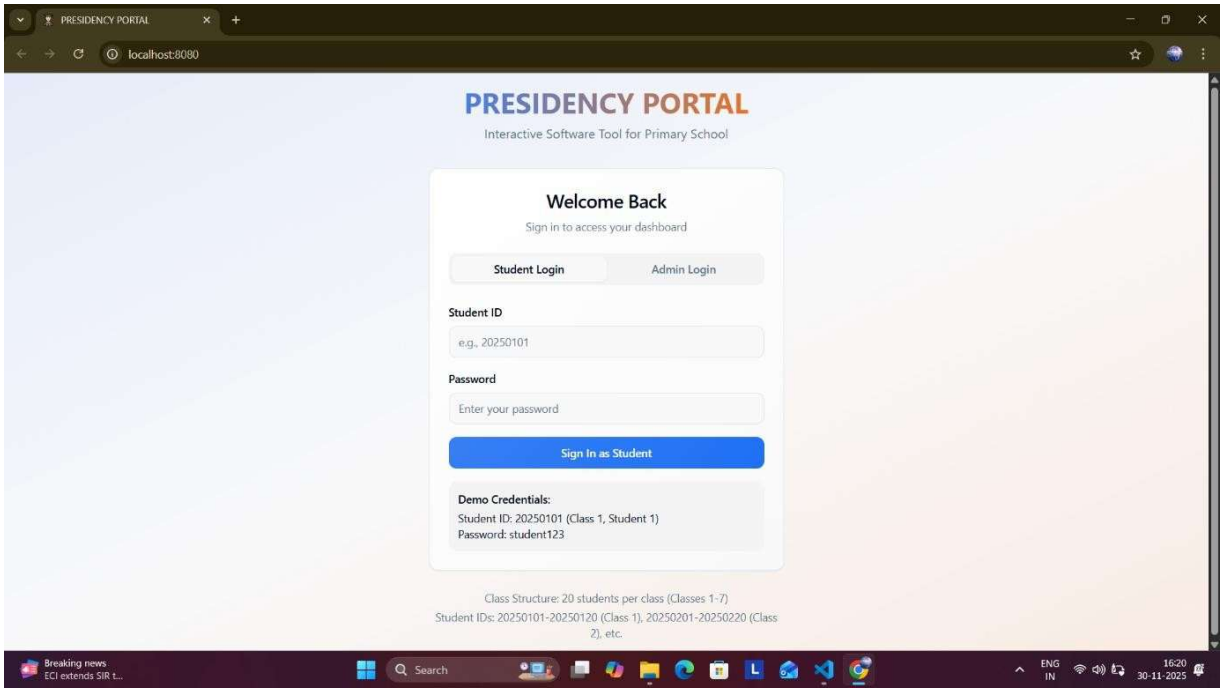


Fig B.1 Login Page

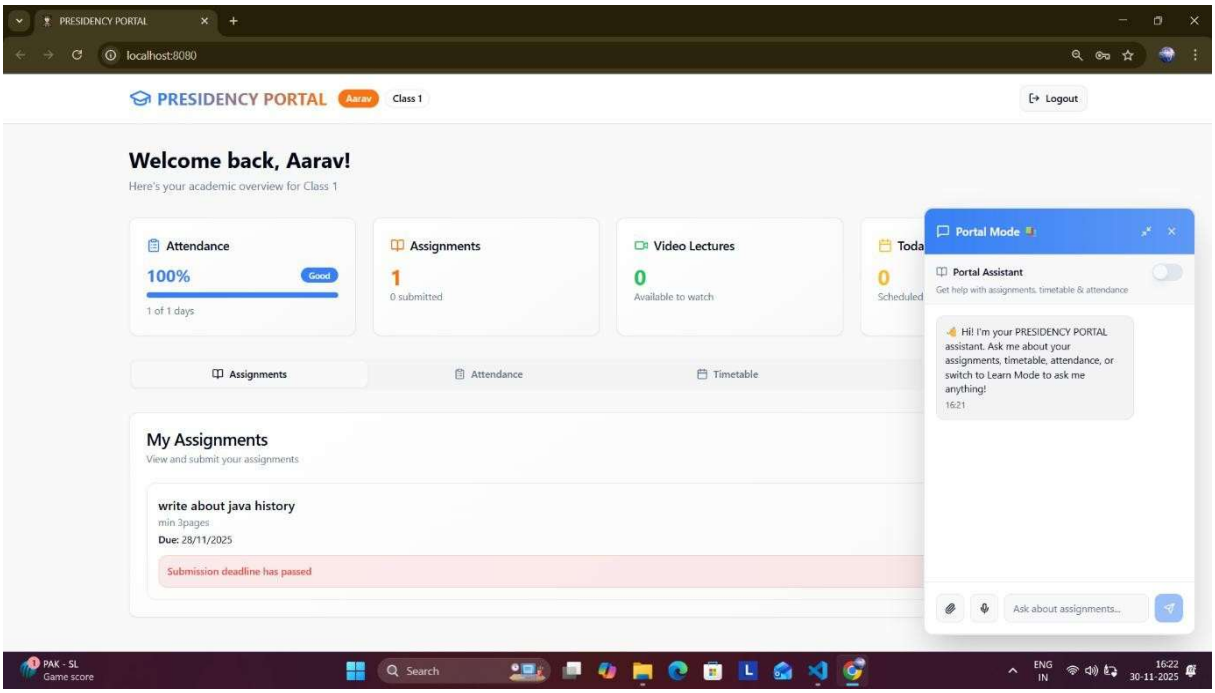


Fig B.2 Student Dashboard

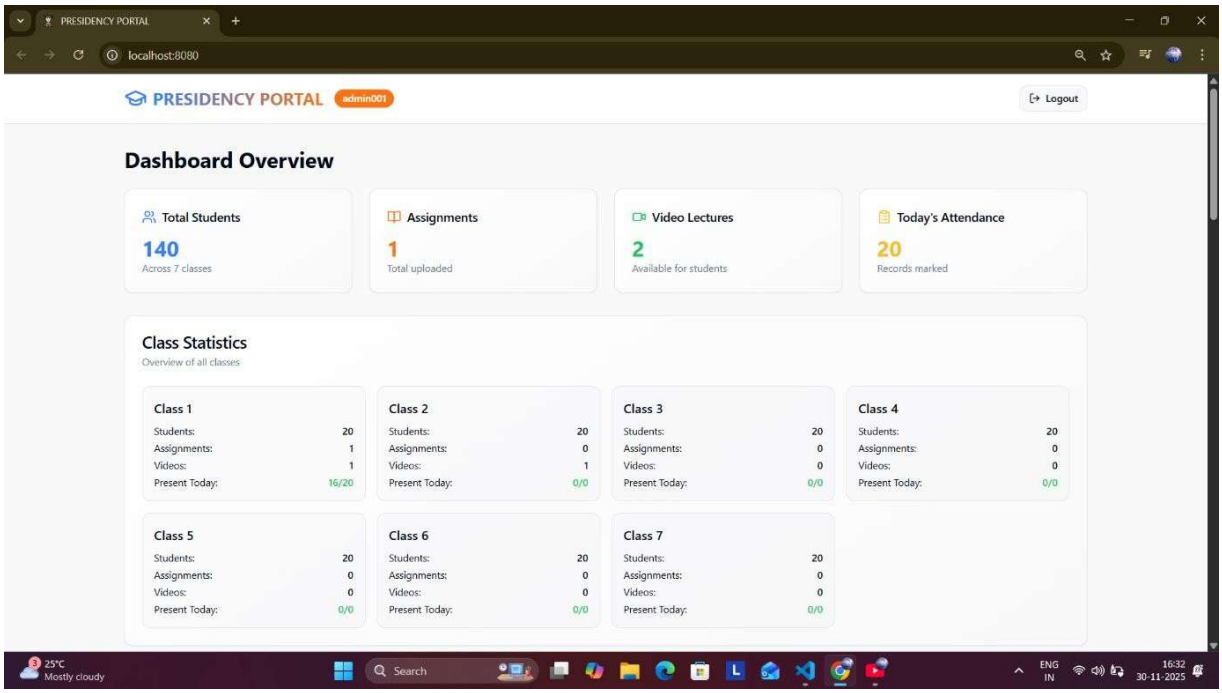


Fig B.3 Admin Dashboard

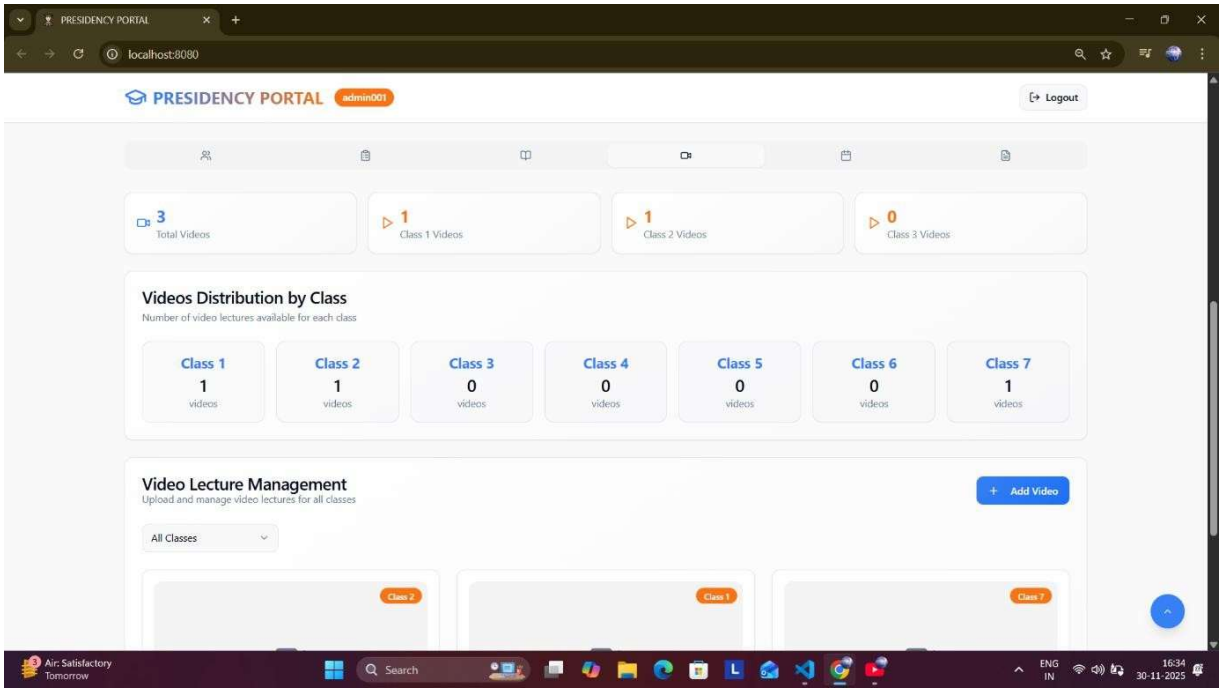


Fig B.4 Admin Video Lectures

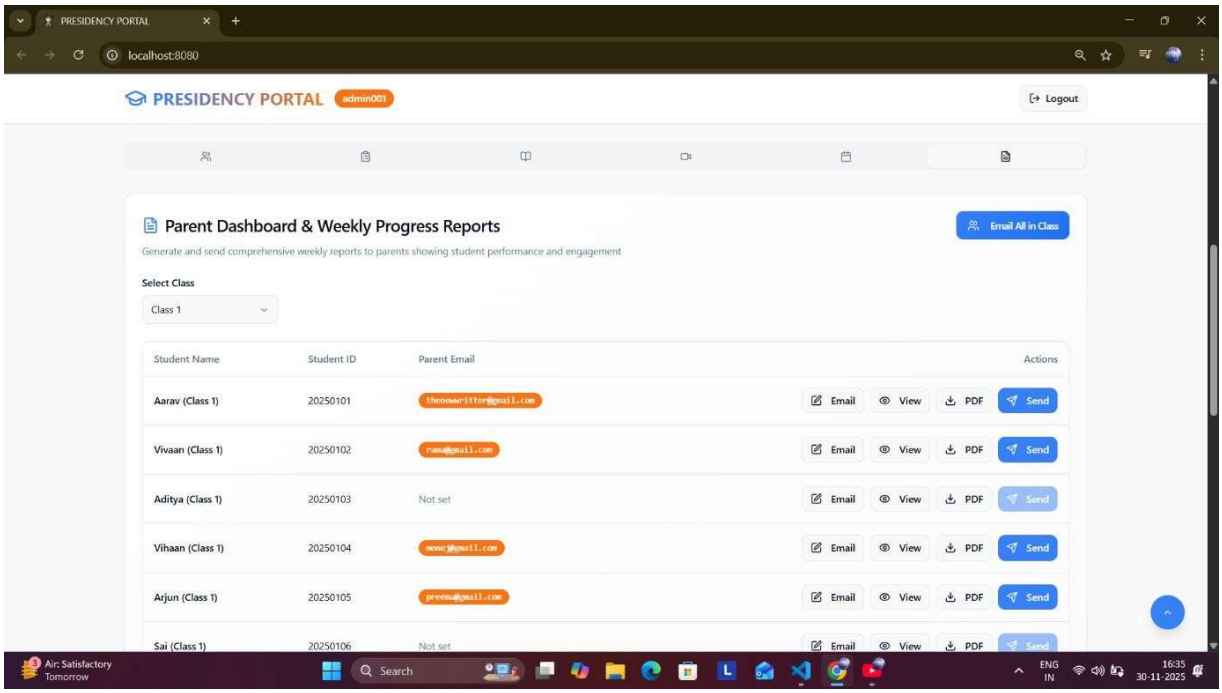


Fig B.5 Parental Report

Appendix C: Code Snippets

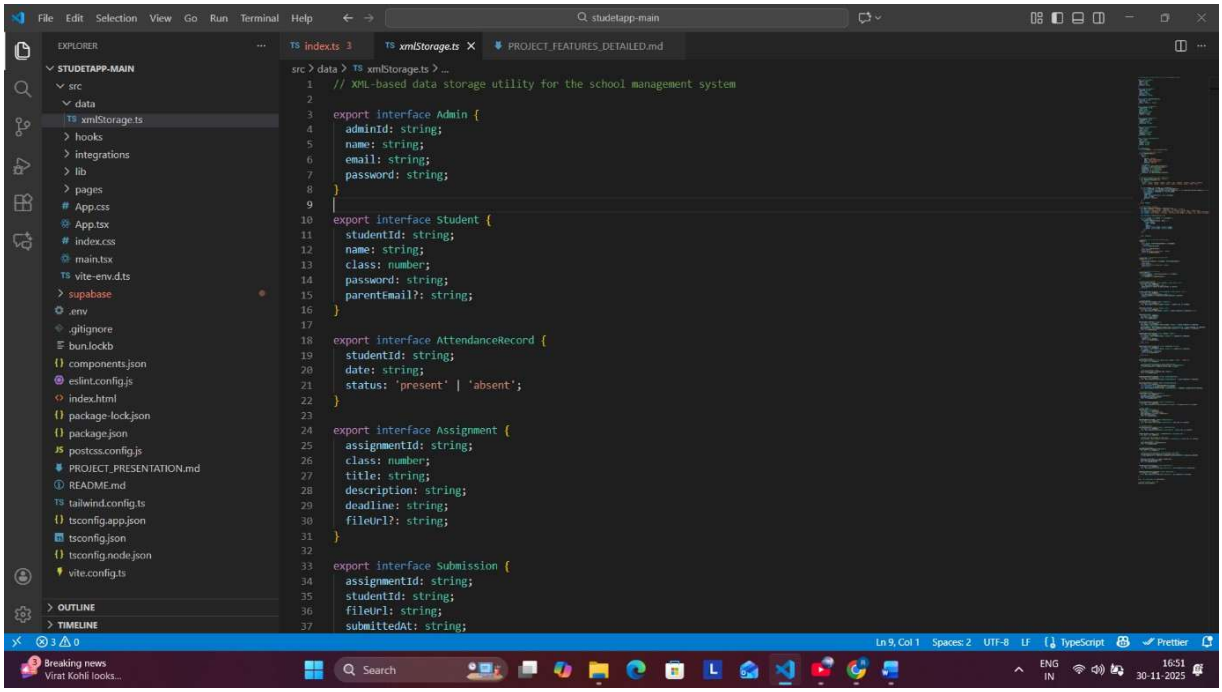


Fig C.1 XML Storage Layer (Data Persistence System)

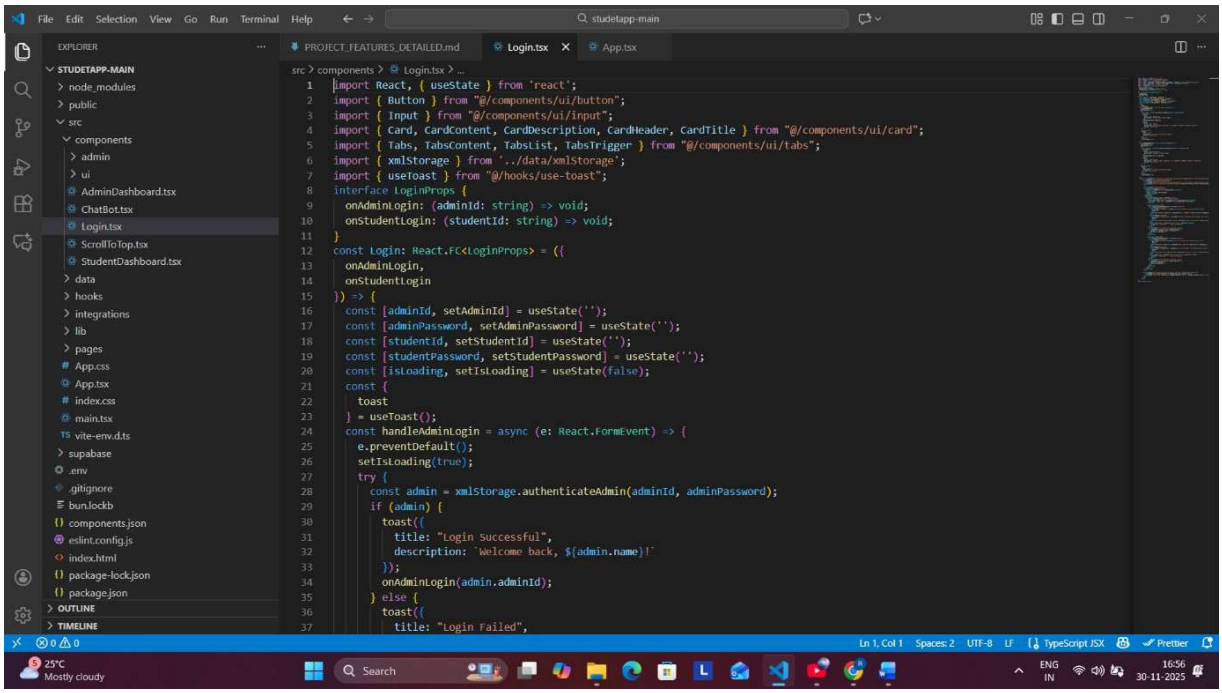


Fig C.2 : Dual Authentication System

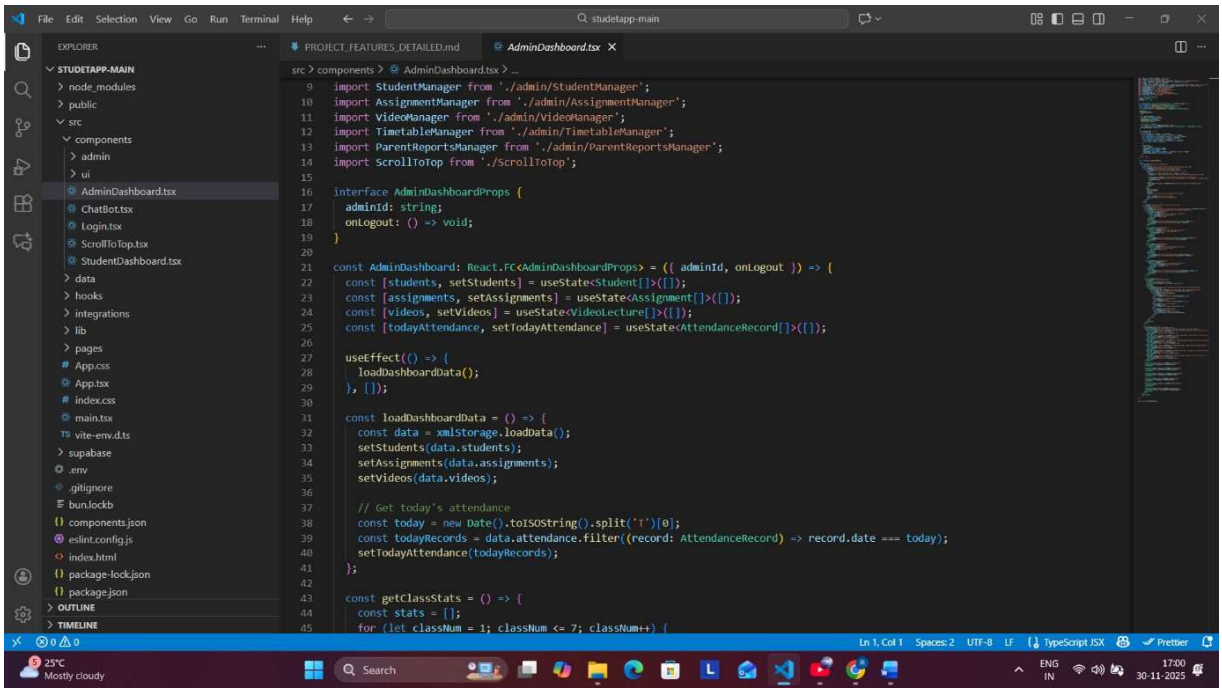


Fig C.3 Admin Dashboard Code (Management Hub)

Appendix D: GitHub Repository

GitHub Link : https://github.com/ShashidharReddyD/4th_Year_Major_Project

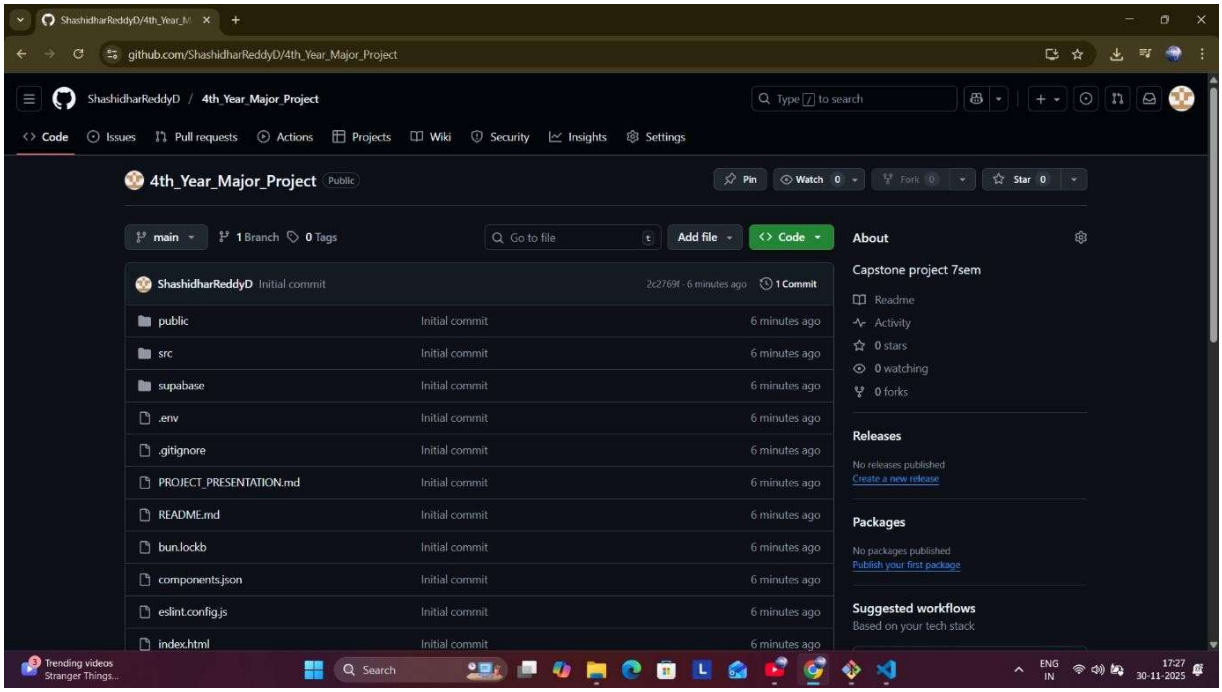


Fig D.1 GitHub Repository

Appendix E: Similarity And AI Report

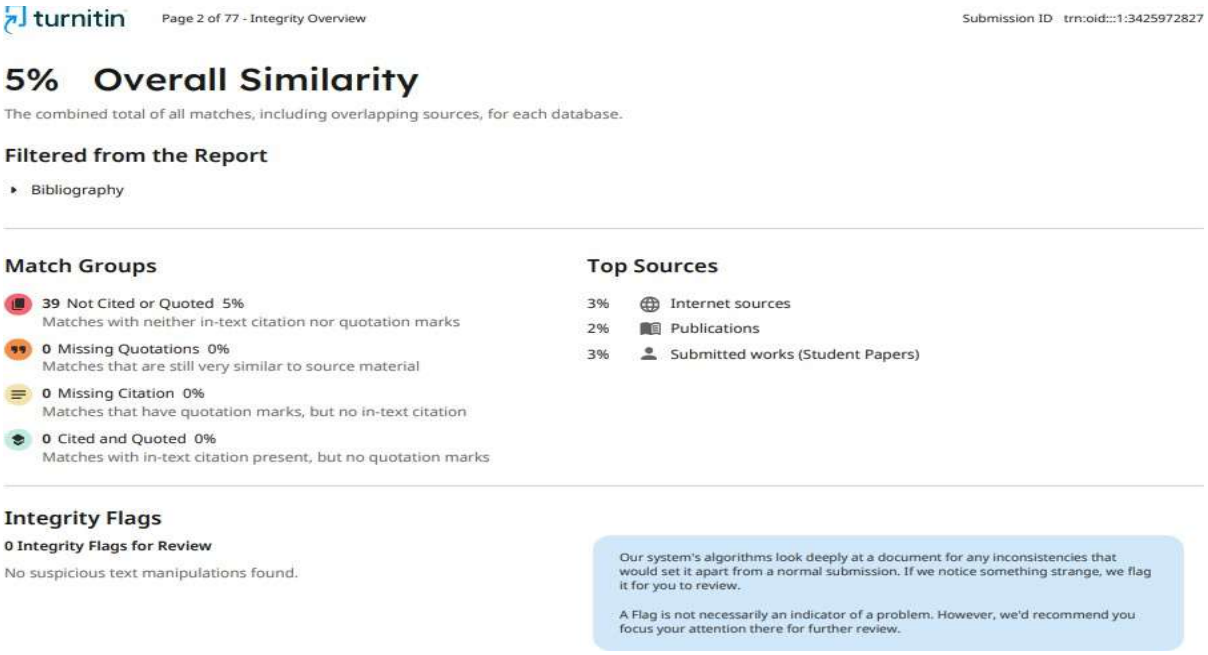


Fig E.1: Report Similarity

turnitin Page 2 of 74 - AI Writing Overview Submission ID trn:oid:::1:3425972827

*% 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.

Disclaimer
Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (i.e., our AI models may produce either false positive results or false negative results), so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

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.

Fig E.2: AI Report