A FIELD PROJECT REPORT

on

“Project Management System using MERN”

Submitted

By

221FA04099
M. Chandana

221FA04104
Sk. Sameena

221FA04256
A. Mokshagna

221FA04709
Rajeev

Under the guidance of

Dr. D. Yakobu

Associate Professor

VIGNAN’S
FOUNDATION FOR SCIENCE, TECHNOLOGY & RESEARCH
(Deemed to be University) - Estd. u/s 3 of UGC Act 1956

SCHOOL OF COMPUTING & INFORMATICS

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

VIGNAN'S FOUNDATION FOR SCIENCE, TECHNOLOGY AND RESEARCH Deemed to be UNIVERSITY

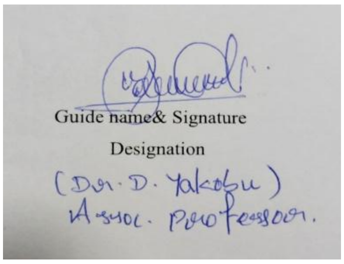
Vadlamudi, Guntur.

ANDHRA PRADESH, INDIA, PIN-522213.

VIGNAN'S
FOUNDATION FOR SCIENCE, TECHNOLOGY & RESEARCH
(Deemed to be University) - Estd. u/s 3 of UGC Act 1956

CERTIFICATE

This is to certify that the Field Project entitled **"Project Management System using MERN"** that is being submitted by 221FA04099 (Chandana), 221FA04104(Sameena), 221FA04256(Mokshagna) and 221FA04709(Rajeev) for partial fulfilment of Field Project is a bonafide work carried out under the supervision of Dr. D. Yakobu, Assistant Professor, Department of CSE.



Guide name& Signature
Designation
(Dr. D. Yakobu)
Assoc. Professor.

Dr. S. V. Phani Kumar
HOD,CSE

VIGNAN'S
FOUNDATION FOR SCIENCE, TECHNOLOGY & RESEARCH
(Deemed to be University) - Estd. u/s 3 of UGC Act 1956

DECLARATION

We hereby declare that the Field Project entitled "**Project Management System using MERN"** that is being submitted by 221FA04099 (Chandana), 221FA04104(Sameena), 221FA04256(Mokshagna) and 221FA04709(Rajeev) in partial fulfilment of Field Project course work. This is our original work, and this project has not formed the basis for the award of any degree. We have worked under the supervision of Dr. D. Yakobu., Assistant Professor, Department of CSE.

By

221FA04099 (Chandana),
221FA04104(Sameena),
221FA04256(Mokshagna),
221FA04709(Rajeev)

ABSTRACT

The CSE department currently lacks a centralized, automated platform to manage and track the lifecycle of student projects, resulting in inefficiencies and significant gaps in oversight, transparency, and communication. As it stands, the department faces the following challenges: Project guides are not assigned systematically based on faculty expertise. This often leads to mismatched assignments where faculty may be given projects outside their areas of expertise, negatively impacting project outcomes. Faculty members are often assigned to guide students in areas where they have minimal experience or knowledge, leading to suboptimal support for students and inefficient project execution. The current process does not provide a comprehensive mechanism for tracking the individual performance of students working on a project. This lack of visibility makes it difficult to gauge progress, identify areas of improvement, or address issues proactively. PRC members, project guides, and even students themselves do not have an easily accessible system to monitor the project's progress or the contributions made by each student. Marks assigned to students are currently done manually by the PRC, which may result in delays, errors, and inconsistencies. There is no centralized platform for reviewing and updating student marks in real-time, leading to a lack of transparency and difficulty in managing changes in grades. Additionally, this manual process increases the chances of discrepancies, affecting the overall integrity of student performance assessments. The absence of collaborative tools makes it difficult for students and faculty to interact seamlessly on the platform. Faculty cannot efficiently communicate feedback, give timely suggestions, or provide constructive criticism, leaving students without adequate guidance. Similarly, students may not be able to communicate effectively among themselves, leading to delays in project completion. There is currently no centralized system where all project-related data, including documents, code, research materials, and progress reports, can be stored and accessed by relevant stakeholders. This lack of a repository increases the risk of losing valuable project data and makes it difficult for PRC members or faculty to review or evaluate the work of students systematically. Assigning project guides to batches is currently done without an automated system, based purely on availability, leading to suboptimal decisions. There is no consideration for the specific experience or expertise required for each project. Without a system to match faculty members' specialization with project needs, the department misses the opportunity to leverage the most suitable faculty for each project, thus impacting the quality of guidance provided to students. The lack of

real-time access to project data and student performance updates hinders the ability of faculty, students, and PRC members to make informed decisions. Faculty and students may not be able to track their progress and performance continuously, while PRC members are unable to make real-time evaluations or intervene when necessary. This can delay feedback loops and impact project timelines. Data such as project status, student marks, and faculty assignments are often spread across multiple disconnected sources, leading to inefficiencies when trying to access or update the information. Stakeholders must manually search for updates, wasting valuable time and risking data loss or inconsistency across different systems. These challenges hinder the overall management of student projects, leading to delays in project completion, poor faculty-student engagement, and a lack of proper monitoring and evaluation. The absence of an integrated, automated, and responsive system also results in communication gaps between the Project Review Committee (PRC), faculty, and students, making it difficult to maintain transparency, ensure accountability, and provide continuous support. Therefore, the need arises for a centralized project management system that automates key processes, facilitates seamless communication between stakeholders, provides real-time tracking of student performance, and simplifies project guide assignment based on expertise. This system will allow: The automatic and efficient assignment of faculty members to student batches based on specialization. Continuous tracking of individual and batch performance.Real-time access to project data for all stakeholders.Automated grade management, evaluation, and reporting. Such a system will not only streamline the project management process but also foster collaboration, ensure transparency, and enhance the overall quality of the projects produced by students in the CSE department.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREIVATIONS

ADS	: ANDROID DATA SYNCHRONIZATION
SDK	: SOFTWARE DEVELOPMENT KIT
ADT	: ANDROID DEVELOPMENT KIT
API	: APPLICATION PROGRAMMING INTERFACE
AOSP	: ANDROID OPEN SOURCE PROJECT
AVD	: ANDROID VIRTUAL DEVICE
CRM	: CUSTOMER RELATIONSHIP MANAGEMENT
UI	: USER INTERFACE
JSON	: JAVA SCRIPT OBJECT NOTATION
XAMPP	: WINDOWS/LINUX APACHE MYSQL PERL PHP
PHP	: HYPERTEXT PREPROCESSOR
IDE	: INTEGRATED DEVELOPMENT ENVIRONMENT
HTML	: HYPER TEXT MARKUP LANGUAGE
JDK	: JAVA DEVELOPMENT KIT
XML	: EXTENSIBLE MARKUP LANGUAGE
WIFI	: WIRELESS FIDELITY
AVD	: ANDROID VIRTUAL DEVICES
ADB	: ANDROID DEBUG BRIDGE

CHAPTER - 1

INTRODUCTION

1. INTRODUCTION

1. Introduction

The Project Management Portal is an innovative solution designed to streamline the management, tracking, and evaluation of student projects within the CSE department. It aims to centralize all project project-released data, allowing students, faculty, and the Project Review Committee (PRC) to efficiently collaborate, monitor progress, and ensure academic success.

With features such as automated faculty assignment based on expertise, real-time performance tracking, and transparent mark evaluations, the portal will improve communication, reduce administrative overhead, and enhance the overall project experience for all stakeholders involved.

1. **Student:**

This user signs up to the system as a batch (minimum-1 and maximum-3) and all the students in a batch have same password. Students must select a domain while registering to the system. Students can log in using his registration number and batch password.

2. **Faculty:**

This user signs in to the system and can upload marks for a batch and each student and can post comments on student performance. He can track the student's progress as well as the whole batch progress that are assigned to him.

3. Deo:

This user can sign in to our system and uploads the promoted list of students who are eligible to apply for a project in their final year. The list uploaded should be in the form of .csv or excel. The list contains all the students irrespective of their status (promoted or not).

4.Project Co-Ordinator:

This user can sign in to the system and verify the promoted list uploaded by the Deo.

He / She will assign the faculty or guides to the batches who registered for project, based on their selected domain.

1. Literature Survey

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, then next steps are to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration are taken into account for developing the proposed system.

- As the project is under the "**Agile Development Model**," the "**Agile for Beginners WBT**" course has been done.
- Took the "**Prodigious Git FP**" course as extra learning to know about the Version control system.

1. Project Background

The CSE department currently lacks a centralized, automated platform to manage and track the lifecycle of student projects, resulting in inefficiencies and significant gaps in oversight, transparency, and communication.

2. Objective

To create a portal or management system that houses and maintains a record of every project created by our CSE department's students. A responsive and easy-to-use website that shows the performance of developers working on each project in their fields of interest. A function that can automatically assign a project guide to each batch according to faculty experience and the area in which they are experts. The PRC oversees and manages each project. The marks assigned to each student individually and to batch can be viewed, downloaded, and edited by everyone in the PRC.

CHAPTER - 2

SOFTWARE REQUIREMENT SPECIFICATION

5. SOFTWARE REQUIREMENTS SPECIFICATION

1. Requirement Analysis

For easy access and portability, we proposed developing a web-based system with windows or ubuntu as a platform because they are mostly used operating systems on the market. As a part of this system, we are going to develop web-based software with which data can be accessed and retrieved easily. The required documents for these processes are as follows.

1. Problem statement
2. Data flow diagrams
3. Use case diagram
4. Other UML diagrams.

The above-mentioned documents give us a diagrammatical view of the system what we are going to develop.

1. Problem Statement

The CSE department currently lacks a centralized, automated platform to manage and track the lifecycle of student projects, resulting in inefficiencies and significant gaps in oversight, transparency, and communication. As it stands, the department faces the following challenges:

Project guides are not assigned systematically based on faculty expertise. This often leads to mismatched assignments where faculty may be given projects outside their areas of expertise, negatively impacting project outcomes. Faculty members are often assigned to guide students in areas where they have minimal experience or knowledge, leading to suboptimal support for students and inefficient project execution.

The current process does not provide a comprehensive mechanism for tracking the individual performance of students working on a project. This lack of visibility makes it difficult to gauge progress, identify areas of improvement, or address issues proactively. PRC members, project guides, and even students themselves do not have an easily accessible system to monitor the project’s progress or the contributions made by each student.

Marks assigned to students are currently done manually by the PRC, which may result in delays, errors, and inconsistencies. There is no centralized platform for reviewing and updating student marks in real-time, leading to a lack of transparency and difficulty in managing changes in grades. Additionally, this manual process increases the chances of discrepancies, affecting the overall integrity of student performance assessments.

The absence of collaborative tools makes it difficult for students and faculty to interact seamlessly on the platform. Faculty cannot efficiently communicate feedback, give timely suggestions, or provide constructive criticism, leaving students without adequate guidance. Similarly, students may not be able to communicate effectively among themselves, leading to delays in project completion.

There is currently no centralized system where all project-related data, including documents, code, research materials, and progress reports, can be stored and accessed by relevant stakeholders. This lack of a repository increases the risk of losing valuable project data and makes it difficult for PRC members or faculty to review or evaluate the work of students systematically.

Assigning project guides to batches is currently done without an automated system, based purely on availability, leading to suboptimal decisions. There is no consideration for the specific experience or expertise required for each project. Without a system to match faculty members' specialization with project needs, the department misses the opportunity to leverage the most suitable faculty for each project, thus impacting the quality of guidance provided to students.

The lack of real-time access to project data and student performance updates hinders the ability of faculty, students, and PRC members to make informed decisions. Faculty and students may not be able to track their progress and performance continuously, while PRC members are unable to make real-time evaluations or intervene when necessary. This can delay feedback loops and impact project timelines.

Data such as project status, student marks, and faculty assignments are often spread across multiple disconnected sources, leading to inefficiencies when trying to access or update the information. Stakeholders must manually search for updates, wasting valuable time and risking data loss or inconsistency across different systems.

These challenges hinder the overall management of student projects, leading to delays in project completion, poor faculty-student engagement, and a lack of proper monitoring and evaluation. The absence of an integrated, automated, and responsive system also results in communication gaps between the Project Review Committee (PRC), faculty, and students, making it difficult to maintain transparency, ensure accountability, and provide continuous support.

Therefore, the need arises for a centralized project management system that automates key processes, facilitates seamless communication between stakeholders, provides real-time tracking of student performance, and simplifies project guide assignment based on expertise. This system will allow:

The automatic and efficient assignment of faculty members to student batches based on specialization.

Continuous tracking of individual and batch performance. Real-time access to project data for all stakeholders. Automated grade management, evaluation, and reporting.

Such a system will not only streamline the project management process but also foster collaboration, ensure transparency, and enhance the overall quality of the projects produced by students in the CSE department.

2. Functional Requirements

1. **Admin:**
    1. Manage User Roles and Permissions
    2. Manage Faculty-Student Assignments
    3. Configure System Settings
    4. Monitor System Activity
    5. Generate System-Wide Reports
    6. Maintain Data Security
    7.
2. **PRC Members:**
    1. Review All Projects
    2. Assign and Manage Marks
    3. Generate Reports
    4. Approve Project Guide Assignments
    5. Track Project Completion
    6. Review Faculty Feedback
3. **Students**

1. Create and Submit Projects
2. Track Individual Performance
3. View Marks and Feedback
4. Collaboration with Team Members
5. Download Reports
6. View Assigned Faculty/Guide.

4. **Faculty/Guides**
   1. Assign and Manage Projects
   2. Provide Feedback to Students
   3. Evaluate Student Performance
   4. Access Project Data
   5. Monitor Batch Performance
   6. Provide Recommendations for Marks

7. Software Requirement Specification

The project is developed in Java Programming Language by using the visual studio. We use the Azure Extension which includes a variety of custom tools that help us to deploy web applications on the azure cloud platform. At the Server-side Apache Tomcat Server is used. Apache Web Server, MySQL is used at the backend and angular is used for the front end.

1. Purpose

The purpose of this document is to present a detailed description of **“Project Management System”** application. It will explain the purpose and features of the system that it will provide, constraints under which it must operate and how the system will react. The document also describes the non functional requirements of the system.

2. Scope of the project

In the project we are going to create a website which consist of various kinds of Users they are Students, Faculty/Guides, PRC members, DEO.

3. Technologies Used

[1] HTML

Hypertext Markup Language is the main markup language for displaying web pages and other information that can be displayed in a web browser.

[2] JAVASCRIPT

JavaScript is a scripting language commonly implemented as part of a web browser in order to create enhanced user interfaces and dynamic websites.

[3] REACT

React is a free and open-source front-end JavaScript library that aims to make building user interfaces based on components more "seamless". It is maintained by Meta and a community of individual developers and companies.

[4] RESTFULSERVIES

REST has become one of the most important technologies for Web applications.

REST stands for **Representational State Transfer**, which is an architectural style for networked hypermedia applications, it is primarily used to build Web services that are lightweight, maintainable, and scalable. A service based on REST is called a RESTful service. REST is not dependent on any protocol, but almost every RESTful service uses HTTP as its underlying protocol.

[5] Mongo

Mongo is one of the most popular relational database management systems on the web. Mongo is used for the internet applications as it provides good speed and is very secure. Mongo was developed to manage large volumes of data at very high speed to overcome the problems of existing solutions.

4. Overview

The application is based on both windows and ubuntu platform an application which communicates with the server allowing the collection of sales visit data offline and then syncs the data with that of the corporate server when there is access to the internet.

8. Software Requirements

The software interface is the operating system, and application programming interface used for the development of the software.

1. Operating System	-	Windows XP
2. Stack	-	Node
3. HTTP version	-	1.1
4. Back End	-	MongoDB
5. Cloud Platform	-	AMAZON WEB SERVICES
6. Technologies	-	Node, React, Rest.

1. Hardware Requirements

CLIENT			
OPERATING SYSTEM	SOFTWARE	DISK SPACE	RAM
Any operating system	Any Web Browser	Minimum 250 MB	256 MB

Table 2.1 Client Requirements

SERVER				
OPERATING SYSTEM	SOFTWARE	PROCESSOR	RAM	DISK SPACE
Any operating system	Apache 2.2.22 MongoDB	Intel Xeon I CPU E31220	256Mb	Minimum 250Mb

Table 2.2 Server Requirements

2. Non-Functional Requirements

1. Backup and Recovery:

The system should have a reliable backup and recovery process in place to prevent data loss in case of failures.

2. Interoperability:

The system should be able to integrate with other software tools used within the department (e.g., gradebook systems, email systems).

3. Standards Compliance:

There shall be consistency in variable names within the system. The graphical user interface shall have a consistent look and feel.

4. Availability:

The system shall be available during normal Canteen operating hours.

5. Portability:

The Canteen Management System shall run in any Microsoft Windows environment that contains browser and be able to available to all users all the time while they have the internet.

6. Reliability:

Since the application is being developed through java, the most famous, efficient and reliable language, so it is reliable in every aspect until and unless there is an error in the programming side. Thus the application can be a compatible and reliable one.

3. External Interface Requirements

1. User Interface

A critical aspect of this project was examining how the app would look and its usability. The layout of web applications is defined in a HTML and JSP files.

When the App is launched, it displays the First Screen- an activity that contains the layout displays buttons for Owner activity and Customer activity. The application has a user-friendly interface.

4. Feasibility study

A key part of the preliminary investigation reviews anticipated costs and benefits and recommends a course of action based on operational, technical, economic, and time factors. The purpose of the study is to determine if the systems request should proceed further.

1. Organizational Feasibility

The application would contribute to the overall objectives of the organization. It would provide a quick, error-free and cost-effective solution to the current process of CRM marketing. It would provide a solution to many issues in the current system. As the new system is flexible and scalable it can also be upgraded and extended to meet other complex requirements which may be raised in the future. However, it is up to the organization to upgrade or extend it.

2. Economic Feasibility

The project is economically feasible as it only requires a computer with Any operating system. The application is free to download once released into the market. The users should be able to connect to internet through computer and this would be the only cost incurred on the project.

3. Technical Feasibility

To develop this application, a high-speed internet connection, a database server, a web server and software are required. The current project is technically feasible as the application was successfully deployed on aws

4. Behavioral Feasibility

The application is behaviorally feasible since it requires no technical guidance, all the modules are user friendly and executed in a manner they were designed to.

CHAPTER - 3

ANALYSIS & DESIGN

7. ANALYSIS & DESIGN

1. Introduction

1. Purpose

In this section the purpose of the document and the project is described.

1. Document Purpose

An SDD is a representation of a software system that is used as a medium for communicating software design information.

2. Project Purpose

The prime purpose of this “Project management system application” is to create a fully-fledged web application which would communicate with the remote server to send and retrieve data as per requirement. This application works when there is internet connectivity or in local lan connection in the scenario of a company canteen Organizations with large numbers of employees cannot handle a canteen with manual processes. They need a centralized canteen management system that promotes efficient operations to cover a large organizational workforce.

2. Scope

In this section the scope of the document and the project is explained in brief.

1. Document Scope

This document contains a thorough description of the high level architecture that will be used in developing the system. Communicating at a purposefully high level, it will only form the basis for the Software Detailed Design and implementation. However, the SDD itself will not be in sufficient detail to implement the code. It will convey the overall system design of the system, the user interface design and higher level module design (including android development tools) and the architecture of the Linux kernel and the working of the Dalvik Virtual Machine. Design details that will not be included in the SDD are:

1. Low level classes that will be used in the implementation. The full description of the implementation of each module is not needed, but the public modules that will be interfaced will be described.

2. Exact detailed description of interactions within each module.

2. System Overview

Cloud deployment framework uses certain development tools which are as follows:

1. Visual Studio Code

Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity).

3. System Architecture

1. Architectural Design

Web application architecture is a framework connecting different elements to enable a web experience. It is the backbone of our daily internet browsing: typing in a URL and viewing and interacting with the website while the browser communicates with the server is one of the ways to describe what web application architecture is.

Attributes of a well-built web application architecture:

1. Solving business problems
2. Supports visual aesthetic
3. Enables A/B testing and analytics
4. Ensures fast user experience
5. Provides security
6. Sustainable and self-regulating
7. Scales out and logs errors in an easy way
8. Guarantees a high level of automation

Components of Web Application Architecture

Web application architecture consists of application components, middleware systems, and databases. They can be divided into two groups:

9. UI/UX components
10. Structural components

UI/UX components include dashboards, statistical data, notification elements, layouts, activity tracking, and other elements. These components create the visuals of a web page and lay the foundation for user experience.

Meanwhile, **structural components** include the web application server and the database server. knowledge of HTML, JavaScript, and CSS, as well as Node.js are required to create them.

When it comes to building the components, there are several models to choose from:

11. 1 web server and 1 database
12. 2 web servers and 2 databases
13. More than 2 web servers and databases

One web server with one database is the simplest model. With this web server architecture, the successful operation of an application depends on server stability. In other words, if there is a problem with the server, the app will not work. Still, the model is sufficient for testing and private sessions.

Using **one database for two web servers** is a more reliable model, as there is a backup server. On the other hand, ensuring the database is secure and always running is important.

Having **more than two databases and web servers** is the most dependable option. Due to its ability to manage and process large amounts of data, this model is a solid the basis for an enterprise web application architecture.

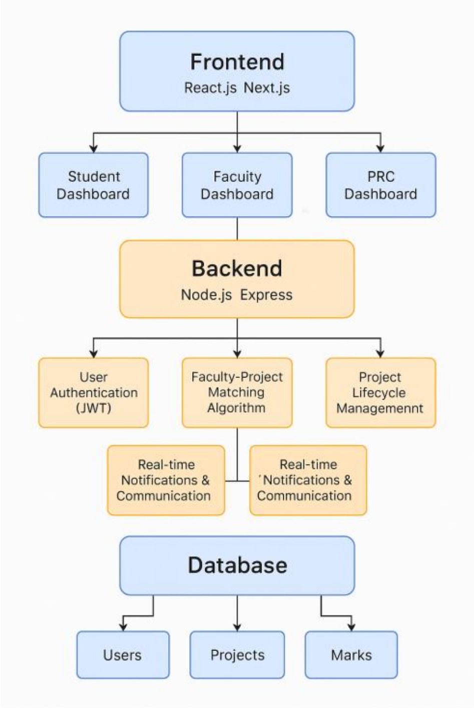

Frontend
React.js Next.js
Student Dashboard
Facuity Dashboard
PRC Dashboard
Backend
Node.js Express
User Authentication (JWT)
Faculty-Project Matching Algorithm
Project Lifecycle Managemennt
Real-time Notifications & Communication
Real-time 'Notifications & Communication
Database
Users
Projects
Marks

Figure 3-1 **Web Application Architecture Diagram**

CHAPTER - 4

MODELING

14. MODELING

1. Design

Requirements gathering followed by careful analysis leads to a systematic Object Oriented Design (OOAD). Various activities have been identified and are represented using Unified Modeling Language (UML) diagrams. UML is used to specify, visualize, modify, construct and document the artifacts of an object-oriented software-intensive system under development.

4.1.1. Use Case Diagram

In the Unified Modeling Language (UML), the use case diagram is a type of behavioral diagram defined by and created from a use-case analysis. It represents a graphical overview of the functionality of the system in terms of actors, which are persons, organizations or external system that plays a role in one or more interaction with the system. These are drawn as stick figures. The goals of these actors are represented as use cases, which describe a sequence of actions that provide something of measurable value to an actor and any dependencies between those use cases.

In this application there is only actor – soldier and below is the use case diagram of this application.

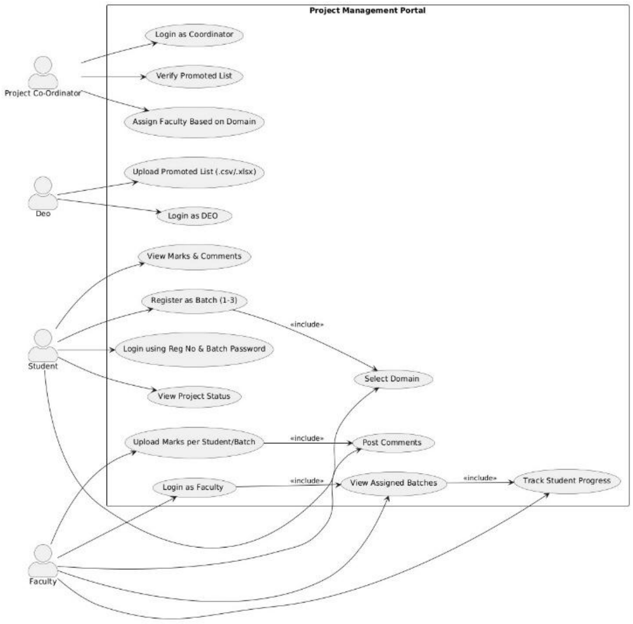


Figure 4-1 Use Case Diagram for System

1. Sequence Diagram

UML sequence diagrams are used to show how objects interact in a given situation. An important characteristic of a sequence diagram is that time passes from top to bottom: the interaction starts near the top of the diagram and ends at the bottom (i.e. Lower equals later).

A popular use for them is to document the dynamics in an object-oriented system. For each key, collaboration diagrams are created that show how objects interact in various representative scenarios for that collaboration.

Sequence diagram is the most common kind of interaction diagram, which focuses on the message interchange between a numbers of lifelines.

The following nodes and edges are typically drawn in a UML sequence diagram: lifeline, execution specification, message, combined fragment, interaction use, state invariant, continuation, destruction occurrence.

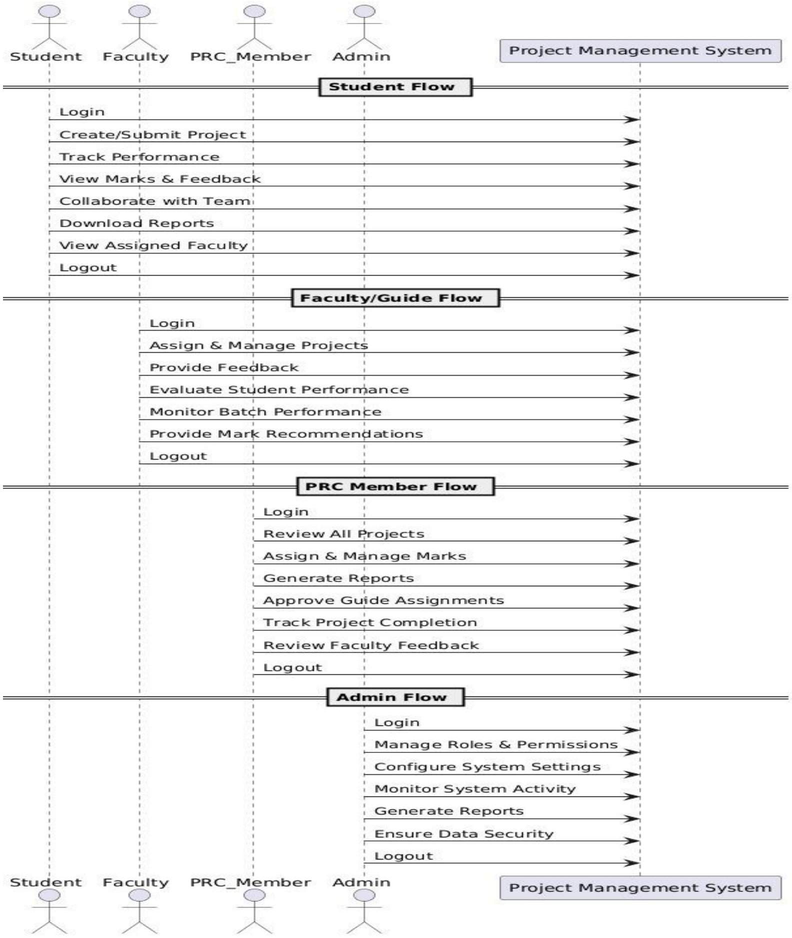


Figure 4-2 Sequence Diagram for Project Management System

2. Activity Diagram

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. Activity diagram is basically a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent. Activity diagrams deals with all type of flow control by using different elements like fork, join etc. Activity is a particular operation of the system.

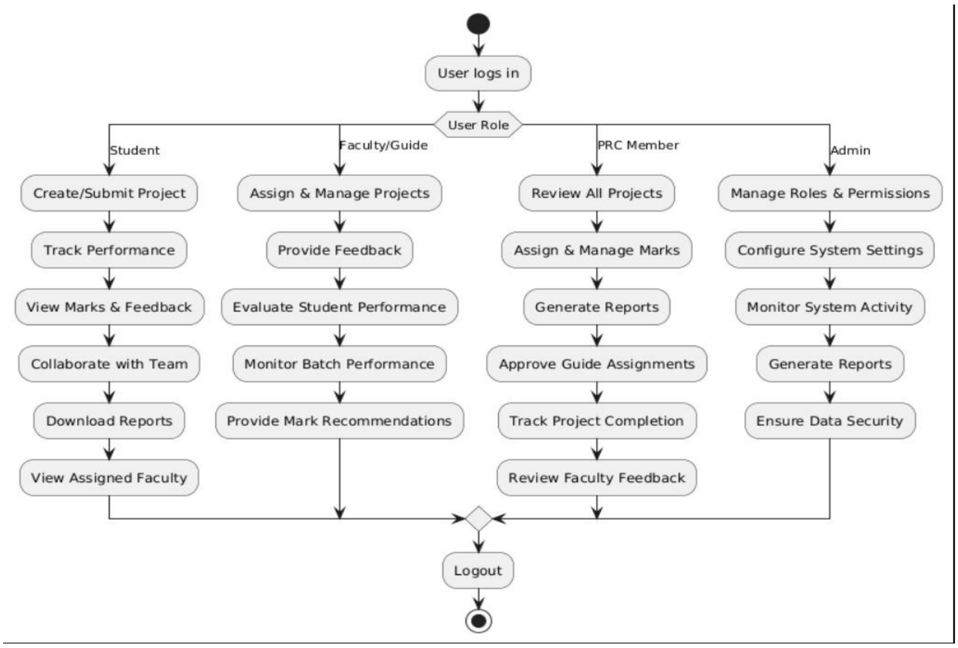


Figure 4-3 Activity Diagram for Project Management System

3. Class Diagram

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system’s classes, their attributes, operations (or methods), and the relationships among the classes.

The class diagram is the main building block of object-oriented Modelling. It is used both for general conceptual modelling of the application, and for detailed modelling translating the models into programming code. Class diagrams can also be used for data modelling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed.

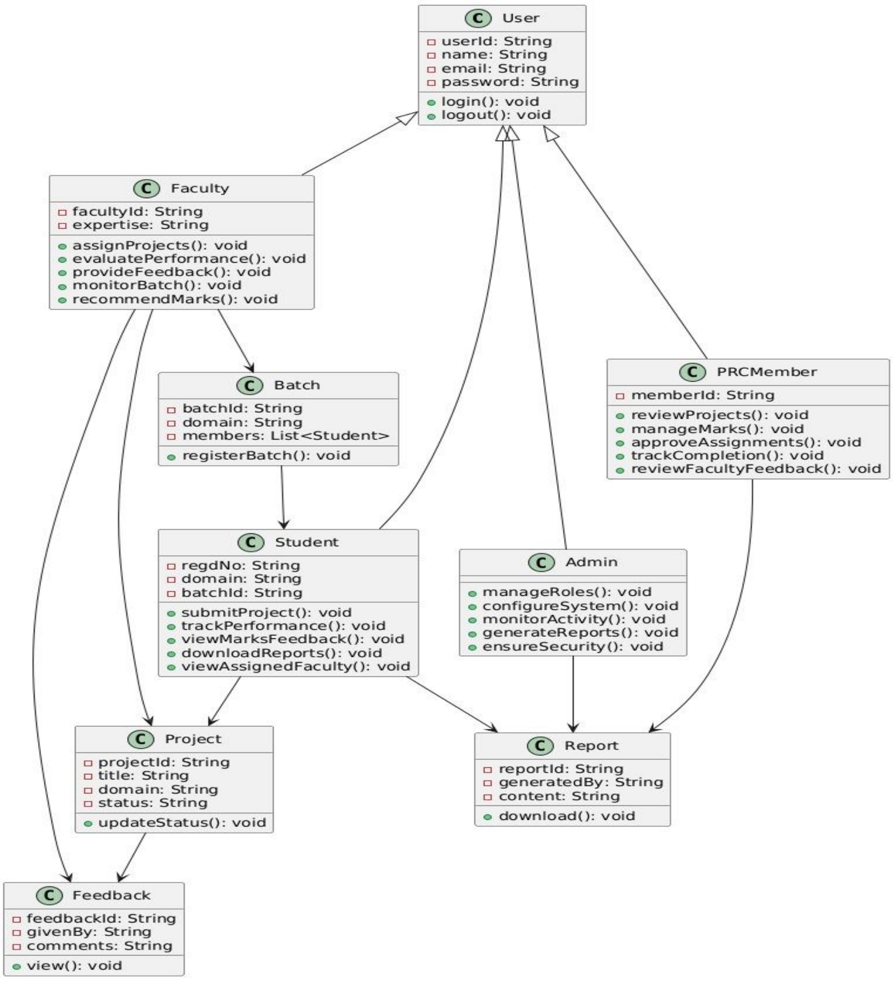


Figure 4-4 Class Diagram for PMS

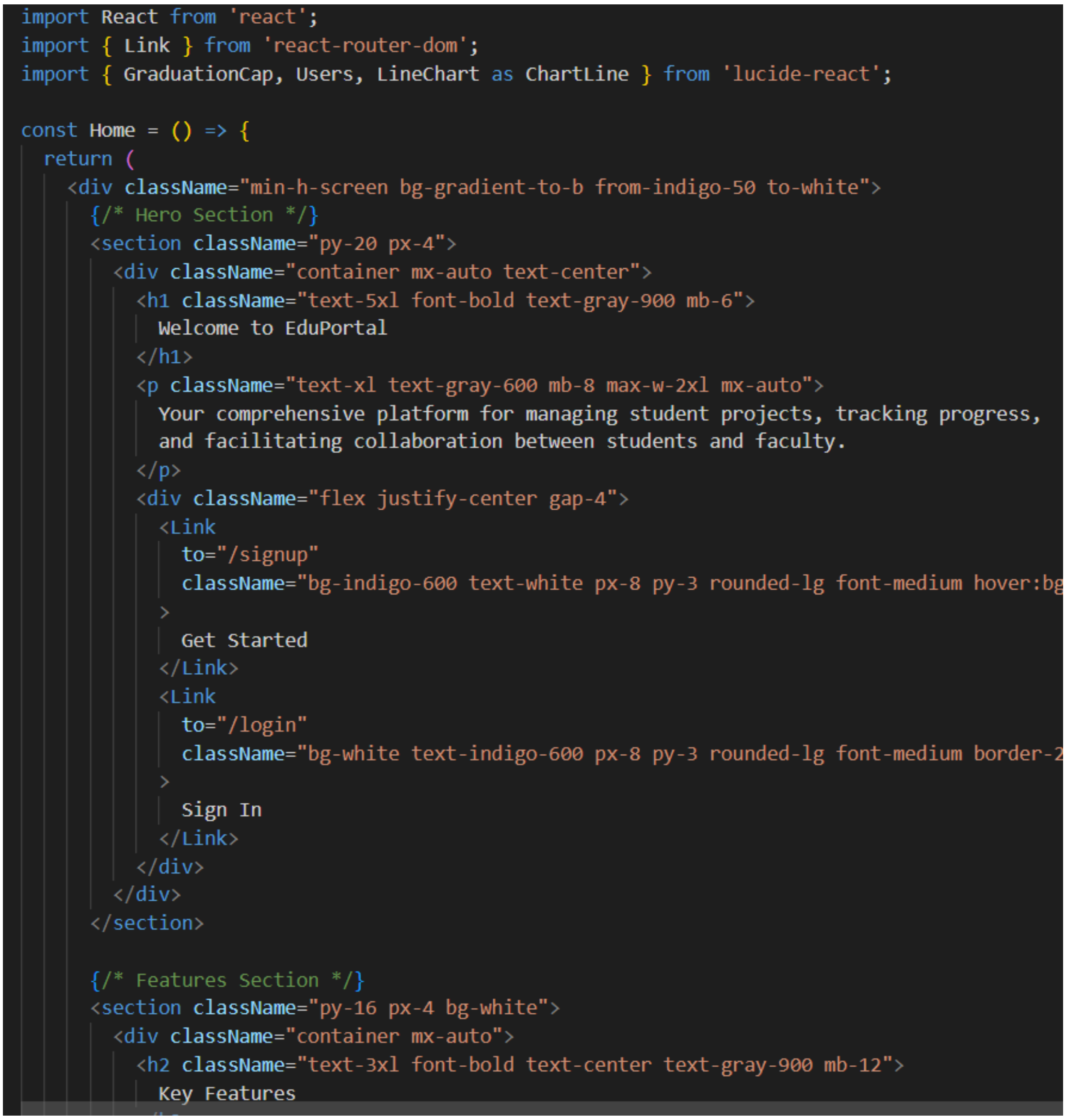
CHAPTER - 5

IMPLEMENTATION

15. IMPLEMENTATION

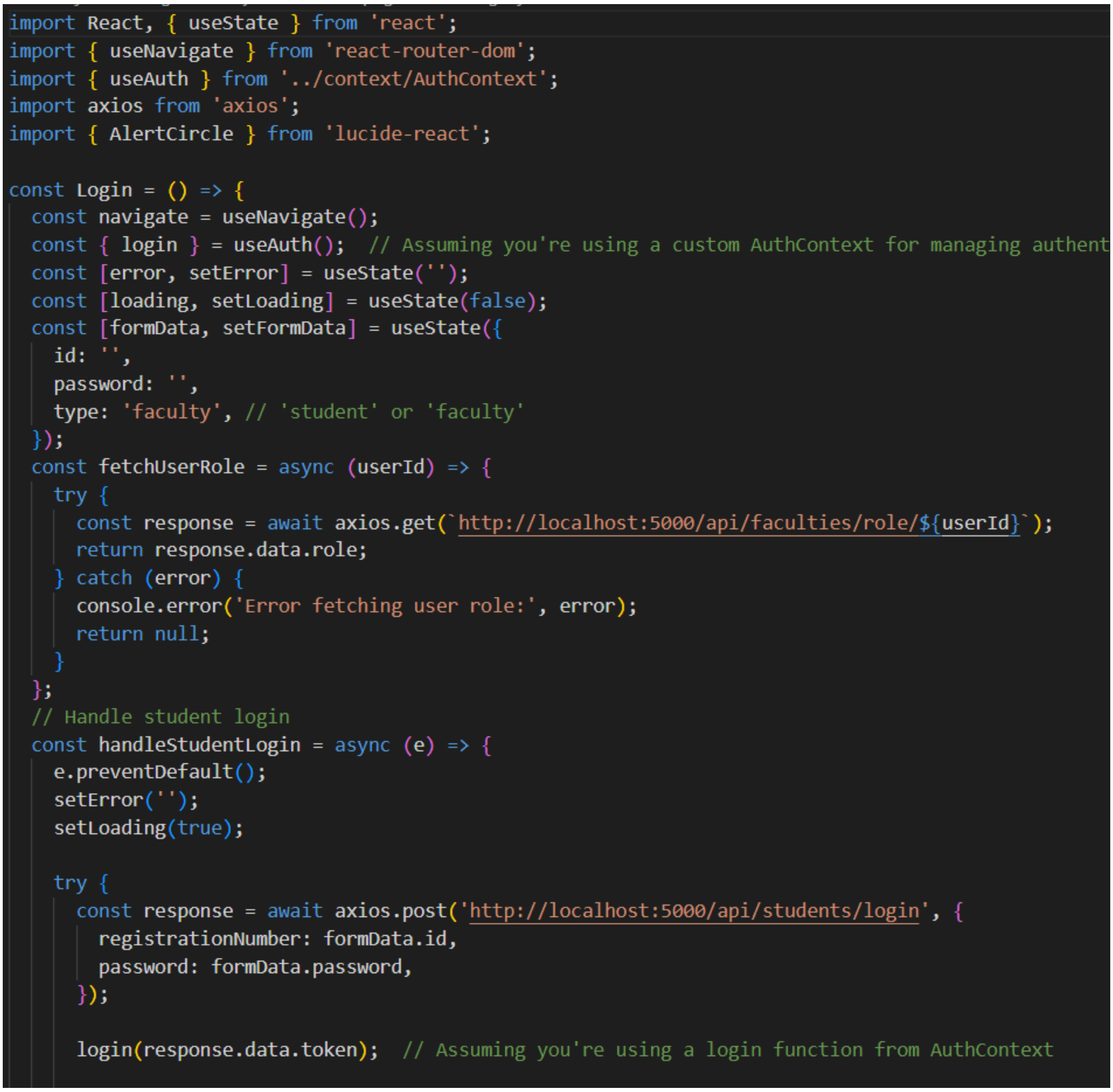
1. Sample Code

1. Code for Home page



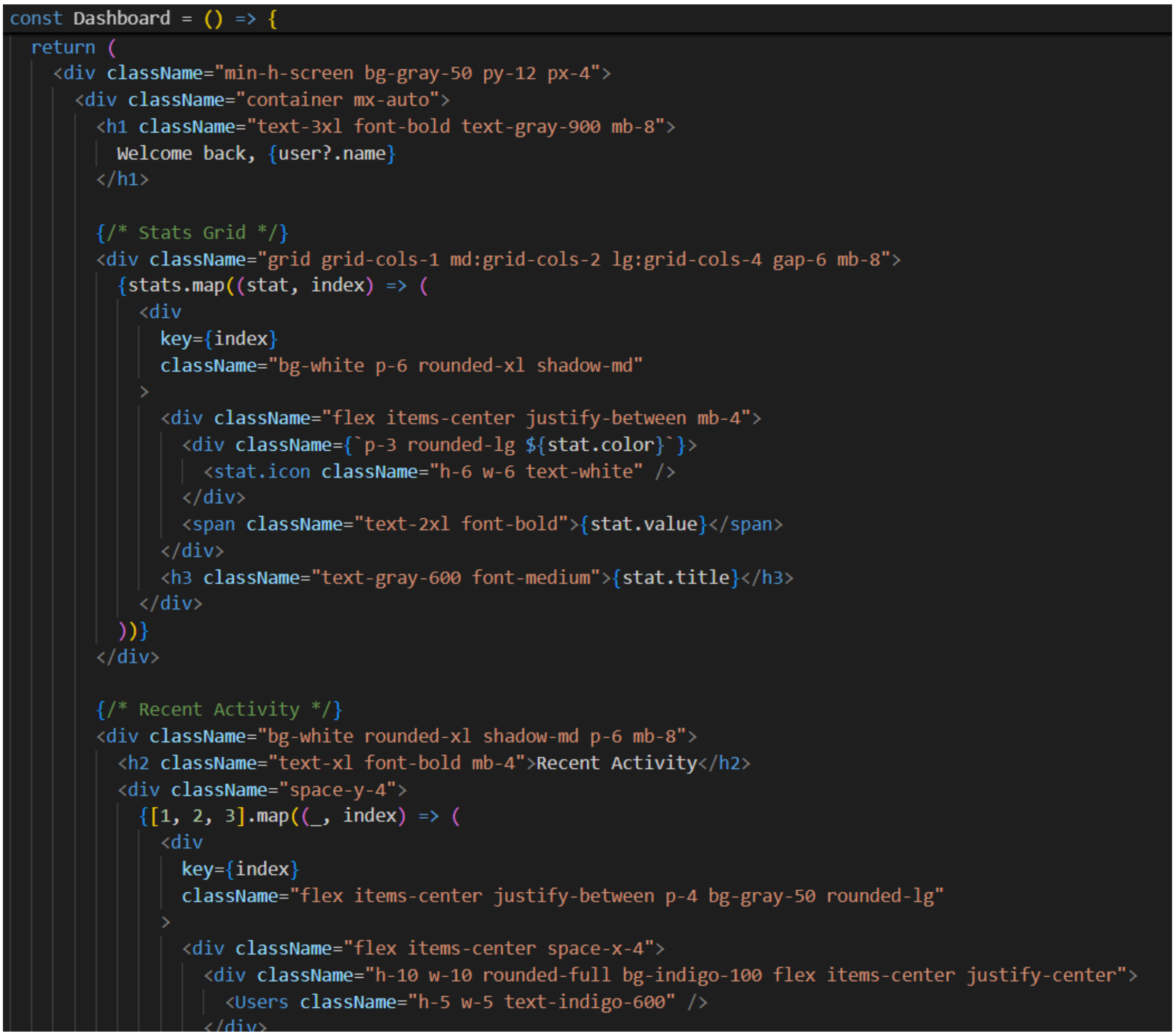
```
import React from 'react';
import { Link } from 'react-router-dom';
import { GraduationCap, Users, LineChart as ChartLine } from 'lucide-react';

const Home = () => {
  return (
    <div className="min-h-screen bg-gradient-to-b from-indigo-50 to-white">
      {/* Hero Section */}
      <section className="py-20 px-4">
        <div className="container mx-auto text-center">
          <h1 className="text-5xl font-bold text-gray-900 mb-6">
            Welcome to EduPortal
          </h1>
          <p className="text-xl text-gray-600 mb-8 max-w-2xl mx-auto">
            Your comprehensive platform for managing student projects, tracking progress,
            and facilitating collaboration between students and faculty.
          </p>
          <div className="flex justify-center gap-4">
            <Link
              to="/signup"
              className="bg-indigo-600 text-white px-8 py-3 rounded-lg font-medium hover:bg
            >
              Get Started
            </Link>
            <Link
              to="/login"
              className="bg-white text-indigo-600 px-8 py-3 rounded-lg font-medium border-2
            >
              Sign In
            </Link>
          </div>
        </div>
      </section>

      {/* Features Section */}
      <section className="py-16 px-4 bg-white">
        <div className="container mx-auto">
          <h2 className="text-3xl font-bold text-center text-gray-900 mb-12">
            Key Features
```

2. Code for User login



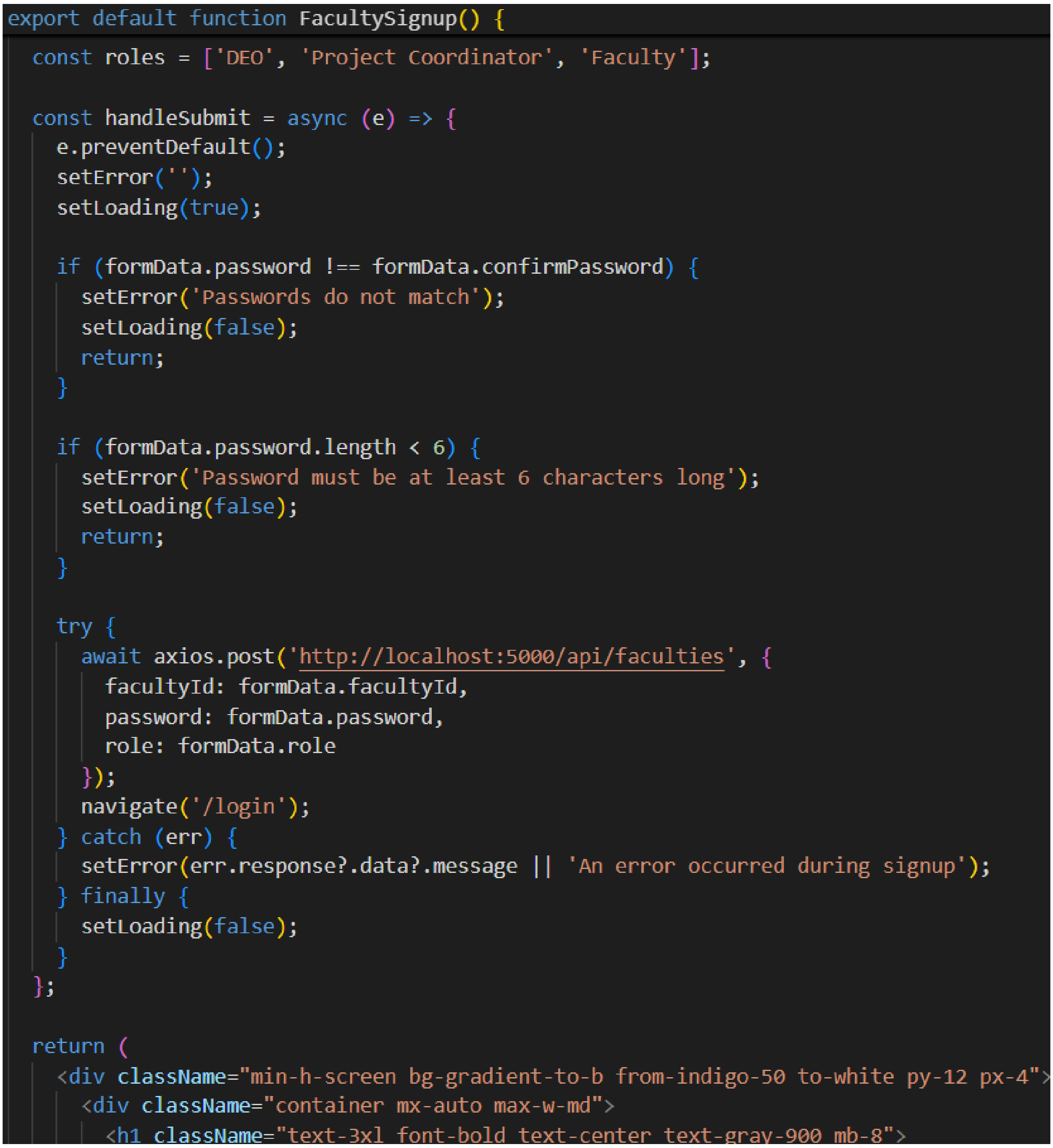
```
import React, { useState } from 'react';
import { useNavigate } from 'react-router-dom';
import { useAuth } from '../context/AuthContext';
import axios from 'axios';
import { AlertCircle } from 'lucide-react';

const Login = () => {
  const navigate = useNavigate();
  const { login } = useAuth();  // Assuming you're using a custom AuthContext for managing authent
  const [error, setError] = useState('');
  const [loading, setLoading] = useState(false);
  const [formData, setFormData] = useState({
    id: '',
    password: '',
    type: 'faculty', // 'student' or 'faculty'
  });
  const fetchUserRole = async (userId) => {
    try {
      const response = await axios.get(`http://localhost:5000/api/faculties/role/${userId}`);
      return response.data.role;
    } catch (error) {
      console.error('Error fetching user role:', error);
      return null;
    }
  };
  // Handle student login
  const handleStudentLogin = async (e) => {
    e.preventDefault();
    setError('');
    setLoading(true);

    try {
      const response = await axios.post('http://localhost:5000/api/students/login', {
        registrationNumber: formData.id,
        password: formData.password,
      });

      login(response.data.token);  // Assuming you're using a login function from AuthContext
```

3. Code for Dashboard



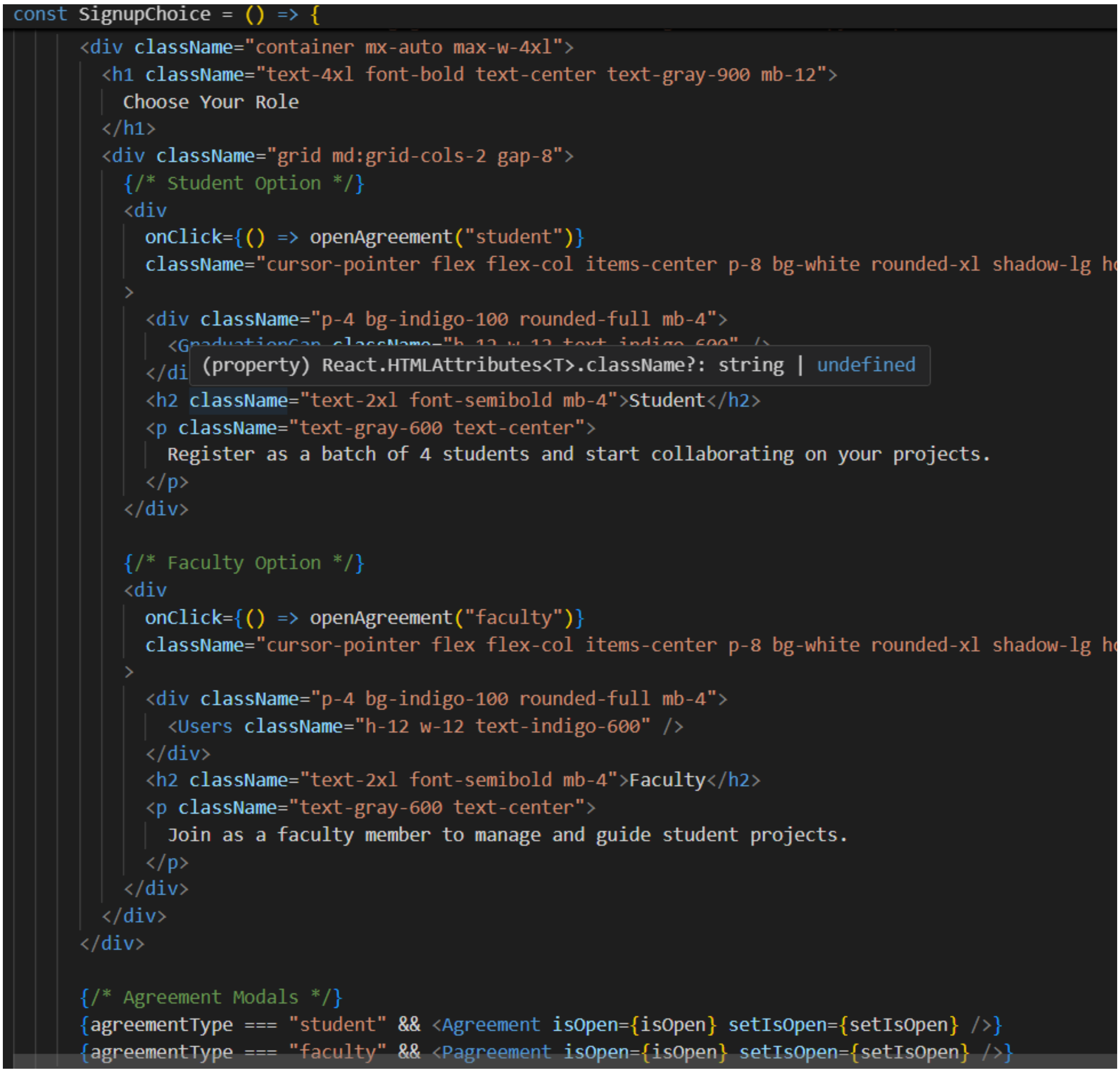
```
const Dashboard = () => {
  return (
    <div className="min-h-screen bg-gray-50 py-12 px-4">
      <div className="container mx-auto">
        <h1 className="text-3xl font-bold text-gray-900 mb-8">
          Welcome back, {user?.name}
        </h1>

        {/* Stats Grid */}
        <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-4 gap-6 mb-8">
          {stats.map((stat, index) => (
            <div
              key={index}
              className="bg-white p-6 rounded-xl shadow-md"
            >
              <div className="flex items-center justify-between mb-4">
                <div className={`p-3 rounded-lg ${stat.color}`}>
                  <stat.icon className="h-6 w-6 text-white" />
                </div>
                <span className="text-2xl font-bold">{stat.value}</span>
              </div>
              <h3 className="text-gray-600 font-medium">{stat.title}</h3>
            </div>
          ))}
        </div>

        {/* Recent Activity */}
        <div className="bg-white rounded-xl shadow-md p-6 mb-8">
          <h2 className="text-xl font-bold mb-4">Recent Activity</h2>
          <div className="space-y-4">
            {[1, 2, 3].map((_, index) => (
              <div
                key={index}
                className="flex items-center justify-between p-4 bg-gray-50 rounded-lg"
              >
                <div className="flex items-center space-x-4">
                  <div className="h-10 w-10 rounded-full bg-indigo-100 flex items-center justify-center">
                    <Users className="h-5 w-5 text-indigo-600" />
                  </div>
```

4. Code for FacultySignUp



```
export default function FacultySignup() {
  const roles = ['DEO', 'Project Coordinator', 'Faculty'];

  const handleSubmit = async (e) => {
    e.preventDefault();
    setError('');
    setLoading(true);

    if (formData.password !== formData.confirmPassword) {
      setError('Passwords do not match');
      setLoading(false);
      return;
    }

    if (formData.password.length < 6) {
      setError('Password must be at least 6 characters long');
      setLoading(false);
      return;
    }

    try {
      await axios.post('http://localhost:5000/api/faculties', {
        facultyId: formData.facultyId,
        password: formData.password,
        role: formData.role
      });
      navigate('/login');
    } catch (err) {
      setError(err.response?.data?.message || 'An error occurred during signup');
    } finally {
      setLoading(false);
    }
  };

  return (
    <div className="min-h-screen bg-gradient-to-b from-indigo-50 to-white py-12 px-4">
      <div className="container mx-auto max-w-md">
        <h1 className="text-3xl font-bold text-center text-gray-900 mb-8">
```

5. Code for Student Signup

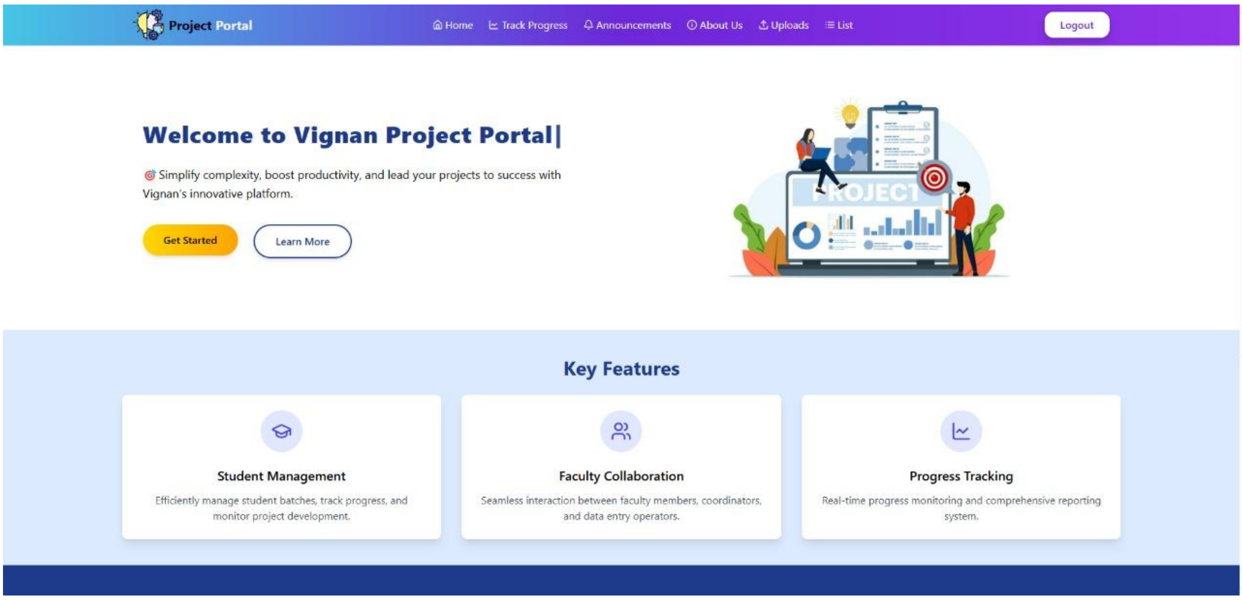


```
const SignupChoice = () => {
      <div className="container mx-auto max-w-4xl">
        <h1 className="text-4xl font-bold text-center text-gray-900 mb-12">
          Choose Your Role
        </h1>
        <div className="grid md:grid-cols-2 gap-8">
          {/* Student Option */}
          <div
            onClick={() => openAgreement("student")}
            className="cursor-pointer flex flex-col items-center p-8 bg-white rounded-xl shadow-lg h
          >
            <div className="p-4 bg-indigo-100 rounded-full mb-4">
              <GraduationCap className="h-12 w-12 text-indigo-600" />
            </di (property) React.HTMLAttributes<T>.className?: string | undefined
            <h2 className="text-2xl font-semibold mb-4">Student</h2>
            <p className="text-gray-600 text-center">
              Register as a batch of 4 students and start collaborating on your projects.
            </p>
          </div>

          {/* Faculty Option */}
          <div
            onClick={() => openAgreement("faculty")}
            className="cursor-pointer flex flex-col items-center p-8 bg-white rounded-xl shadow-lg h
          >
            <div className="p-4 bg-indigo-100 rounded-full mb-4">
              <Users className="h-12 w-12 text-indigo-600" />
            </div>
            <h2 className="text-2xl font-semibold mb-4">Faculty</h2>
            <p className="text-gray-600 text-center">
              Join as a faculty member to manage and guide student projects.
            </p>
          </div>
        </div>
      </div>

      {/* Agreement Modals */}
      {agreementType === "student" && <Agreement isOpen={isOpen} setIsOpen={setIsOpen} />}
      {agreementType === "faculty" && <Pagreement isOpen={isOpen} setIsOpen={setIsOpen} />}
```

2. Screen Captures

1. home screen

Figure 5-1 home screen Activity



Welcome to Vignan Project Portal|

🎯 Simplify complexity, boost productivity, and lead your projects to success with Vignan's innovative platform.

Key Features

Student Management

Efficiently manage student batches, track progress, and monitor project development.

Faculty Collaboration

Seamless interaction between faculty members, coordinators, and data entry operators.

Progress Tracking

Real-time progress monitoring and comprehensive reporting system.

Description: After launching the application a home screen will come into view.

2. Login Screen

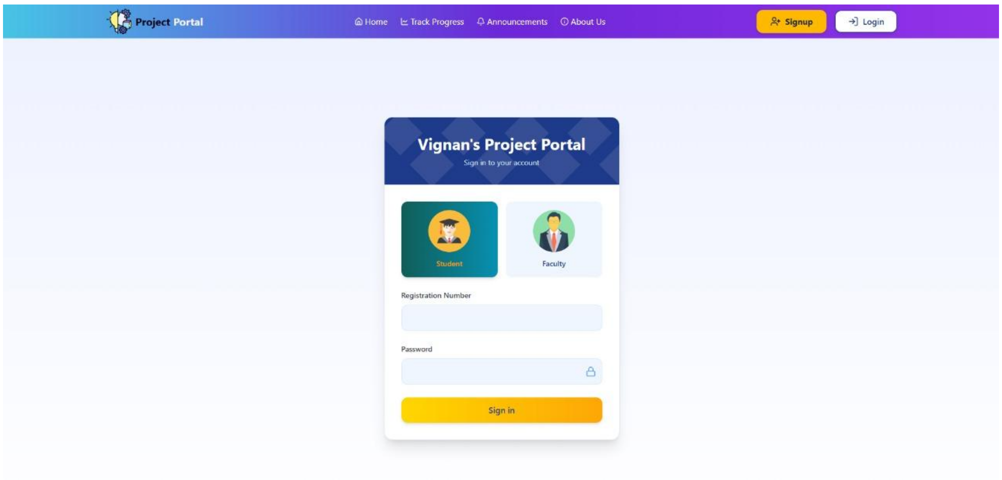


Figure 5-2 Login Activity

3. Faculty Signup Screen

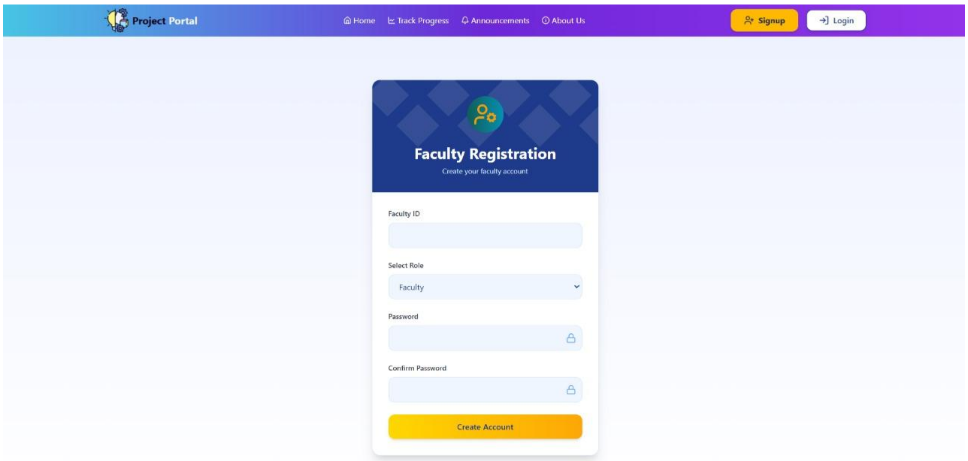
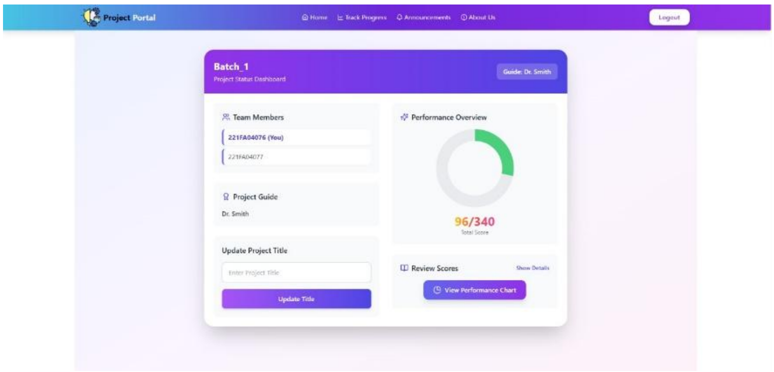


Figure 5-3 Faculty Signup Activity

4. Student dashboard Screen



Batch_1
Project Status Dashboard
Guide: Dr. Smith

Team Members
221FA04076 (You)
221FA04077

Project Guide
Dr. Smith

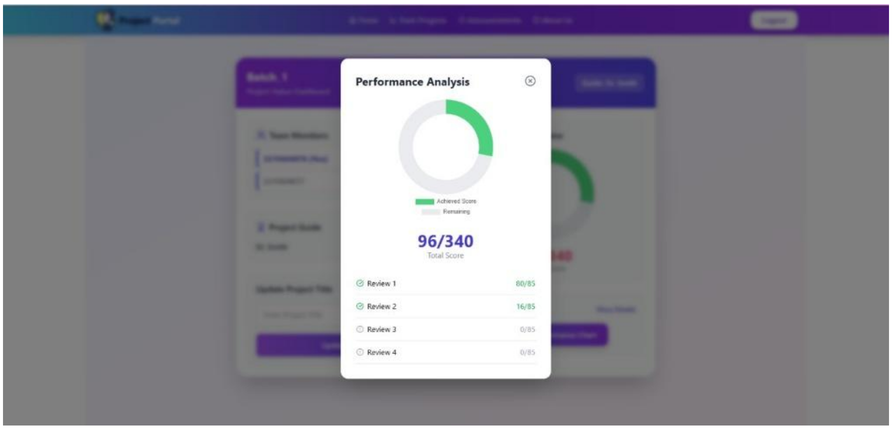
Update Project Title
Enter Project Title

Performance Overview
96/340
Total Score

Review Scores

Figure 5-4 Student dash board Screen Activity

5. Student Performance Screen



Performance Analysis

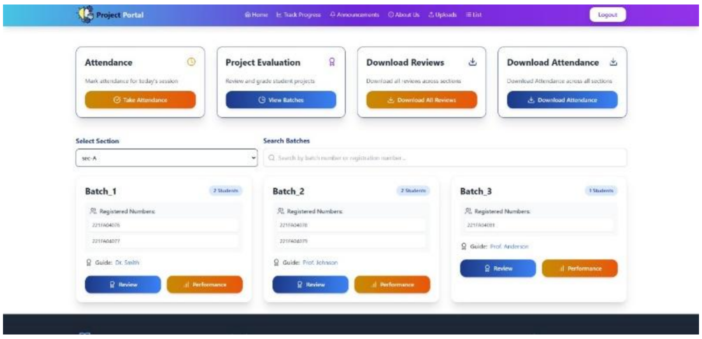
96/340
Total Score

Review 1	80/85
Review 2	16/85
Review 3	0/85
Review 4	0/85

Figure 5-5 Student Performance screen Activity

Description: vendor details screen will shows the details about the vendor which consist of columns like name, id, email, phone number and address of the vendor

6. PC view Activity



Attendance
Mark attendance for today's session

Project Evaluation
Review and grade student projects

Download Reviews
Download all reviews across sections

Download Attendance
Download Attendance across all sections

Select Section: sec-A

Search Batches: Search by batch number or registration number...

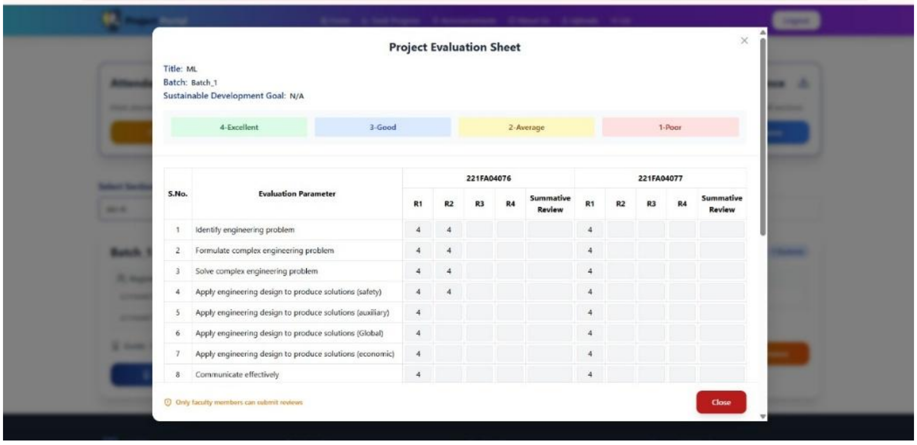
Batch_1 — 2 Students
Registered Numbers:
221FA04076
221FA04077
Guide: Dr. Smith

Batch_2 — 2 Students
Registered Numbers:
221FA04078
221FA04079
Guide: Prof. Johnson

Batch_3 — 1 Students
Registered Numbers:
221FA04081
Guide: Prof. Anderson

Figure 5-6 List of Batches

1. Evaluation



Project Evaluation Sheet

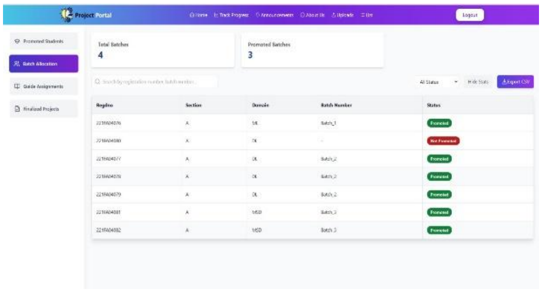
Title: ML
Batch: Batch_1
Sustainable Development Goal: N/A

S.No.	Evaluation Parameter	221FA04076					221FA04077				
		R1	R2	R3	R4	Summative Review	R1	R2	R3	R4	Summative Review
1	Identify engineering problem	4	4				4				
2	Formulate complex engineering problem	4	4				4				
3	Solve complex engineering problem	4	4				4				
4	Apply engineering design to produce solutions (safety)	4	4				4				
5	Apply engineering design to produce solutions (auxiliary)	4					4				
6	Apply engineering design to produce solutions (Global)	4					4				
7	Apply engineering design to produce solutions (economic)	4					4				
8	Communicate effectively	4					4				

Only faculty members can submit reviews

Figure 5-7 Evaluation

2. Batch Allotments



Total Batches
4

Promoted Batches
3

Regdno	Section	Domain	Batch Number	Status
221FA04076	A	ML	Batch_1	Promoted
221FA04080	A	DL	-	Not Promoted
221FA04077	A	DL	Batch_2	Promoted
221FA04078	A	DL	Batch_2	Promoted
221FA04079	A	DL	Batch_2	Promoted
221FA04081	A	MSD	Batch_3	Promoted
221FA04082	A	MSD	Batch 3	Promoted

Figure 5-8 Batch Allotments Activity Flow

CHAPTER - 6

TESTING

.

16. TESTING

1. Software Testing

Software testing is the process of validating and verifying that a software application meets the technical requirements which are involved in its design and development. It is also used to uncover any defects/bugs that exist in the application. It assures the quality of the software. There are many types of testing software viz., manual testing, unit testing, black box testing, performance testing, stress testing, regression testing, white box testing etc. Among these performance testing and load testing are the most important ones for an android application and next sections deal with some of these types.

2. Black box Testing

Black box testing treats the software as a "black box"—without any knowledge of internal implementation. Black box testing methods include equivalence partitioning, boundary value analysis, all-pairs testing, fuzz testing, model-based testing, traceability matrix, exploratory testing and specification-based testing.

3. White box Testing

White box testing is when the tester has access to the internal data structures and algorithms including the code that implements these.

4. Performance Testing

Performance testing is executed to determine how fast a system or sub-system performs under a particular workload. It can also serve to validate and verify other quality attributes of the system such as scalability, reliability and resource usage.

5. Load Testing

Load testing is primarily concerned with testing that can continue to operate under specific load, whether that is large quantities of data or a large number of users.

6. Manual Testing

Manual Testing is the process of manually testing software for defects. The functionality of this application is manually tested to ensure the correctness. A few examples of test case for Manual Testing are discussed later in this chapter.

CHAPTER - 7

RESULTS &CHALLENGES

17. RESULTS AND CHALLENGES

1. Results

The implementation of the centralized project management system significantly streamlined the student project lifecycle in the CSE department. Faculty were assigned to projects based on expertise, improving the quality of mentorship and project outcomes. Students and faculty gained real-time access to project data, enhancing transparency and communication.

Automated grade entry reduced manual errors and ensured timely evaluations. The system enabled continuous performance tracking for both individuals and batches, allowing early identification of issues. Centralized document storage improved data accessibility and reduced the risk of loss.

2. Challenges

1. Ensuring accurate mapping of faculty expertise with project domains.
2. Managing real-time data synchronization and access control.
3. Training users (students, faculty, PRC) to adapt to the new system.
4. Maintaining data security and privacy for sensitive academic records.

CHAPTER - 8

CONCLUSIONS

5. CONCLUSION

1. Conclusions

In conclusion, the Project Management Portal will significantly improve the efficiency and effectiveness of project management within the CSE department. By centralizing project data, automating faculty assignments, and providing real-time tracking of student performance, the portal will foster better communication, collaboration, and transparency among students, faculty, and PRC members. The ability to easily monitor progress, provide feedback, and manage evaluations will ensure that projects are completed successfully and that students receive the guidance they need. Ultimately, this portal will enhance the overall academic experience, streamline administrative tasks, and contribute to the continuous growth and success of the department.

BIBLIOGRAPHY

[1] *Pro MERN Stack: Full Stack Web App Development with Mongo, Express*, *React, and Node*, Apress, 2019.

[2] Florian Dedov, *MERN Stack Development: Build and Deploy a Full-Stack App from Scratch*, Independently Published, 2022.

[3] MongoDB Documentation: https://www.mongodb.com/docs/

[4] Express.js Documentation: https://expressjs.com/

[5] React.js Documentation: https://react.dev/

[6] Node.js Documentation: https://nodejs.org/en/docs/