
SILVER OAK UNIVERSITY
EDUCATION TO INNOVATION

**SCHOOL OF TECHNOLOGY, DESIGN**

**AND COMPUTER APPLICATION**

**SILVER OAK COLLEGE OF ENGINEERING & TECHNOLOGY**

**DEPARTMENT OF INFORMATION TECHNOLOGY**

A Report on

# ‘Smart Campus : Enhancing University Productivity and Engagement ’

Under Subject of

**PROBLEM BASED LEARNING - II**

Semester - VI

***Submitted by***

**Jil Upadhyay : 2201031800044**

**Jeel Rajpara : 2201030400113**

**Tejus Prasad : 2201031830013**

**Nishit Chaudhary : 2201030400013**

**Vishwa Panchal : 2201030400195**

**Faculty Name**

Mr. Digant Parmar

**HOD Name**

Mr. Mayuresh Kulkarni

# Silver Oak College of Engineering & Technology

Department of Information Technology

# CERTIFICATE

This is to certify that the project entitled “Project Title” has been carried out by **“Jil Upadhyay”, “Jeel Rajpara” , “Tejus Prasad” , “Nishit Chaudhary” , “Vishwa Panchal”** under my guidance in fulfillment of the Problem Based Learning-II (1010003392) Subject of Bachelor of Engineering in **Computer Engineering** – 6th Semester of Silver Oak University, Ahmedabad during the academic year 2024- 2025.

**Faculty Name**
Mr. Digant Parmar

**HOD Name**
Mr. Mayuresh Kulkarni

# ACKNOWLEDGEMENT

We are heartily thankful to our supervisor ,**Mr.Digant Parmar,** whose encouragement, supervision and support from the preliminary to the concluding level enabled me to develop an understanding of the subject. At the end, we offer my regards and blessings to all of those who supported us in any respect during the completion of the project and to our college for providing resources and materials.

We would like to extend my gratitude to **HOD Mr. Mayuresh Kulkarni** head of Computer Engineering Department, Silver Oak college of Engineering and Technology, Ahmedabad, for his continuous encouragement and motivation.

Last but not the least we would like to mention here that we are greatly indebted to each and everybody who has been associated with our project at any stage but whose name does not find a place in this acknowledgement.

Yours Sincerely

**Jil Upadhyay**

**Jeel Rajpara**

**Vishwa Panchal**

**Tejus Prasad**

**Nishit Chaudhary**

# ABSTRACT

This project seeks to address these challenges by developing a comprehensive calendar application specifically designed for university students, integrating critical features that cater to their unique academic lifestyles.

The proposed system aims to unify and streamline academic planning through a responsive and intuitive interface that consolidates exam schedules, event calendars, academic announcements, and learning material sharing into a single platform. With the rise of collaborative learning and digital-first education environments, this application serves as a smart assistant for students, promoting improved time management, proactive planning, and increased involvement in university activities.

At its core, the application is powered by a robust backend built with Go (Golang)—chosen for its high performance and scalability—and a modern, highly responsive frontend developed using Next.js, ensuring cross-platform compatibility and smooth user experiences across both desktop and mobile devices. The technology stack not only guarantees performance but also provides a scalable architecture for future expansion and feature enhancement.

Key functionalities include:

- Dynamic exam and event scheduling, allowing students to view, manage, and plan according to real-time academic calendars.
- Customizable event reminders and smart notifications that alert users of upcoming classes, deadlines, and events.
- Collaborative event planning, enabling students and faculty to co-create and manage group activities or meetings.
- Centralized sharing of academic materials and announcements, helping students stay up-to-date without the need for scattered platforms or communication channels.

- Advanced search and filtering capabilities, allowing users to quickly

find relevant information, documents, or events using intelligent keyword and category-based searching.

The overarching goal of this project is to create a smart, student-centric platform that not only serves as a digital calendar but also as a productivity hub—improving academic outcomes, fostering community engagement, and supporting the holistic development of university students. By leveraging modern web technologies and user-centric design, this application represents a step forward in the digital transformation of campus life.

# LIST OF FIGURES

**Figure No.** **Figure Name** **Page No.**

# TABLE OF CONTENT

## Introduction

University life is a dynamic and multifaceted experience that demands effective time management and organizational skills from students. Amidst lectures, assignments, examinations, club meetings, workshops, and campus events, students often struggle to keep track of their obligations. The typical student juggles various responsibilities and deadlines, often relying on multiple disconnected tools or, in some cases, none at all. This lack of a centralized platform frequently results in confusion, overlooked assignments, missed events, and decreased productivity.

In most academic institutions, the tools available to students for managing their academic and extracurricular schedules are either too generic or not tailored to the specific requirements of campus life. Traditional calendar applications and reminders offer basic scheduling functionalities but lack integration with university systems such as course schedules, examination timetables, assignment deadlines, event announcements, and faculty notifications. As a result, students must manually manage their time, which often leads to inconsistencies and inefficiencies.

Moreover, campus events—whether academic seminars, sports competitions, cultural fests, or departmental gatherings—often suffer from poor student turnout due to inadequate promotion and scheduling conflicts. Without a streamlined platform to consolidate these events and alert students in real time, opportunities for engagement and community-building are frequently lost.

The absence of a unified and smart system not only hinders individual performance but also affects the overall student experience on campus. It limits students' ability to actively participate in academic and extracurricular opportunities, negatively impacting their development and well-being. As higher education continues to evolve with increasing digital integration, there is a growing need for smart solutions that align with the demands of modern student life.

## Purpose of the Smart Campus Application

The primary purpose of the **Smart Campus** application is to provide university students with a centralized, intelligent, and user-friendly platform that integrates all aspects of their academic and extracurricular schedules. This digital solution aims to simplify time management and enhance student engagement by offering a single interface where users can:

- **Access academic calendars** including class schedules, examination dates, assignment deadlines, and semester plans.
- **Receive automated notifications and reminders** for upcoming tasks and events.
- **Stay informed about campus activities** such as club meetings, guest lectures, sports events, and cultural festivals.
- **Customize personal schedules** by adding personal tasks or events alongside academic obligations.
- **Enable real-time updates** from university administration or faculty regarding schedule changes or important announcements.

By combining these features into one cohesive system, Smart Campus eliminates the fragmentation of information and provides students with a streamlined way to organize their daily lives. The application not only boosts individual productivity but also fosters better communication, planning, and participation within the student community.

In essence, Smart Campus is more than just a calendar app—it is a comprehensive productivity and engagement tool designed to meet the real-world needs of university students. It supports academic success, promotes involvement in campus life, and contributes to a more organized and fulfilling university experience.

# Objective

The primary objective of this project is to develop a comprehensive calendar application that caters to the unique needs of university students. The application will:

- Integrate Exam Schedules: Provide a centralized platform to display all exam dates, allowing students to plan their study schedules efficiently.
- Event Scheduling: Include academic and extracurricular event schedules, ensuring students are informed about upcoming activities and can participate accordingly.
- Customizable Reminders: Offer customizable reminders for exams, assignments, events, and other important deadlines, helping students stay on top of their commitments.
- Centralized Announcements: Serve as a hub for all academic and administrative announcements, ensuring students receive timely and relevant information.
- Enhanced Productivity: Aim to improve students' overall productivity by streamlining the management of their academic and extracurricular schedules.
- Academic Performance Improvement: Contribute to better academic performance by helping students manage their time effectively and meet their deadlines.

# Methodology

The development of the Smart Campus application involves several key steps:

1. Requirements Analysis: Identify and document the specific needs and requirements of the target users (university students and faculty).
2. System Design: Develop a detailed design for the application, including the database schema, user interface wireframes, and system architecture.
3. Hardware Implementation: Determine and set up the necessary hardware infrastructure for the application.
4. Software Development: Develop the backend and frontend components of the application using GO and Next.js, respectively.
5. Integration and Testing: Integrate all components of the application and conduct thorough testing to ensure functionality and performance.
6. Iterative Improvement: Continuously improve the application based on user feedback and performance metrics.

# Requirements Analysis

## Functional Requirements

- Integration of Exam Schedules: Centralized display of exam dates.
- Integration of Event Schedules: Include academic and extracurricular events.
- Announcements Integration: Consolidated platform for important announcements.
- Customizable Reminders: Personalized alerts for events, exams, and deadlines.
- Collaborative Event Planning: Tools for group event organization and voting.
- Centralized Academic Notifications: Single hub for academic updates and notices.
- User Account Management: Secure login and profiles for users.
- Notification System: Push notifications for announcements.
- Calendar Syncing: Sync with other calendars like Google Calendar.
- Search and Filter Options: Easy search and filtering of events.
- Fee : Manage paying for fees on time avoiding penalties.

## Non-functional Requirements

- User-friendly interface.
- Responsive design for compatibility with various devices.
- Secure data handling and storage.

# System Design

## Backend Architecture

The backend architecture will be designed to efficiently handle the various functionalities of the Smart Campus application. The key components include:

### Go Server

- API Server: A single server handling all the API endpoints and business logic, simplifying deployment and maintenance.
- Custom Authentication: Implement custom authentication mechanisms to ensure secure access to the application.
- JSON RESTful Endpoints: Use RESTful endpoints for communication between the client and server, ensuring a standardized interface for data exchange.

### Database

- Postgresql : A NoSQL database used to store user data, event details, schedules, and other application data. MongoDB is chosen for its flexibility and scalability.

## Frontend Architecture

The frontend will be designed to provide a responsive and intuitive user interface, leveraging modern web technologies.

### Next.js Web Application

- Nextt.js: A powerful JavaScript library for building user interfaces, chosen for its component-based architecture and efficient rendering.
- Responsive Design: Ensure the web application is accessible on various devices, including desktops, tablets, and smartphones.

### Notification System

- Firebase: Utilize Firebase Cloud Messaging (FCM) to send push notifications to users. This will be used for real-time alerts about new announcements, upcoming events, and reminders.

# Hardware Implementation

## Containerization with Docker

To ensure a scalable, portable, and efficient deployment environment, the Smart Campus application will utilize containerization technology with Podman. All software components, including the API server, databases, caching system, will be managed within containers. This approach provides consistency across different development, testing, and production environments.

Deployment of Components on Docker

- Go Server: The API server, handling all API endpoints and business logic, will be deployed in a Podman container.
- Postgresql : The NoSQL database for storing user data, event details, and other application data will be deployed in a separate Podman container.

## IoT Features Integration

To support real-time physical access and interaction on the Smart Campus, various IoT components are integrated with backend systems.

Components Used

- **ESP32S Microcontroller**: Acts as the primary IoT controller, interfacing with various peripherals and communicating with the server via Wi-Fi.
- **RDM6300 RFID Reader (125KHz)**:Used for scanning RFID cards for access control and attendance management. Connects directly to the ESP32 via UART.
- **RFID Cards**: Unique ID cards assigned to students and staff. When scanned, the UID is sent to the server to verify identity and log events.
- **RGB LEDs**:Provide visual feedback for access (e.g., green for access granted, red for denied, blue for standby).
- **Relay Switch**: Controls physical devices such as door locks or lights. Triggered based on access verification.

Languages Used

- **Programming Language**: C/C++
- **Development Environment**: Arduino IDE
- **Communication**: Wi-Fi via ESP32, HTTP/REST API for server interaction

Functionality Flow

- **Card Scanning**: RFID card is scanned via the RDM6300.
- **UID Transmission**: ESP32 sends the UID to the API server over HTTP.
- **Verification**: Server checks the UID against the database.
- **Action Triggered**:
    1. If verified: Green LED blinks and relay unlocks the door.
    2. If not verified: Red LED blinks, access denied.

# Software Development

## Backend Development

The backend of the Smart Campus application will be developed using JavaScript to leverage its strong typing and enhanced development experience. The Go runtime is chosen for its stability, extensive ecosystem, and large community support. The following components and technologies will be utilized:

### Go Environment

- Go Runtime: Utilized for its stability, extensive ecosystem, and community support.
- JavaScript: Used for backend development to ensure type safety, improved code maintainability, and enhanced developer productivity.

### Database Management

- MongoDB: Chosen for its schema-less nature, allowing flexible and scalable data storage.
- Mongoose: An ODM (Object Data Modeling) library for MongoDB and Node.js, used to interact with MongoDB, providing schema validation and query building.

### User Authentication

- JWT (JSON Web Tokens): Used for secure user authentication and authorization, ensuring that users are who they claim to be and have the necessary permissions.

### Notifications

- Firebase Cloud Messaging (FCM): Integrated for sending push notifications to users, providing real-time alerts about new announcements, upcoming events, and reminders.

### Containerization

- Docker: Utilized for containerizing the backend services, ensuring a consistent and isolated environment for development, testing, and production deployment.

### API Development

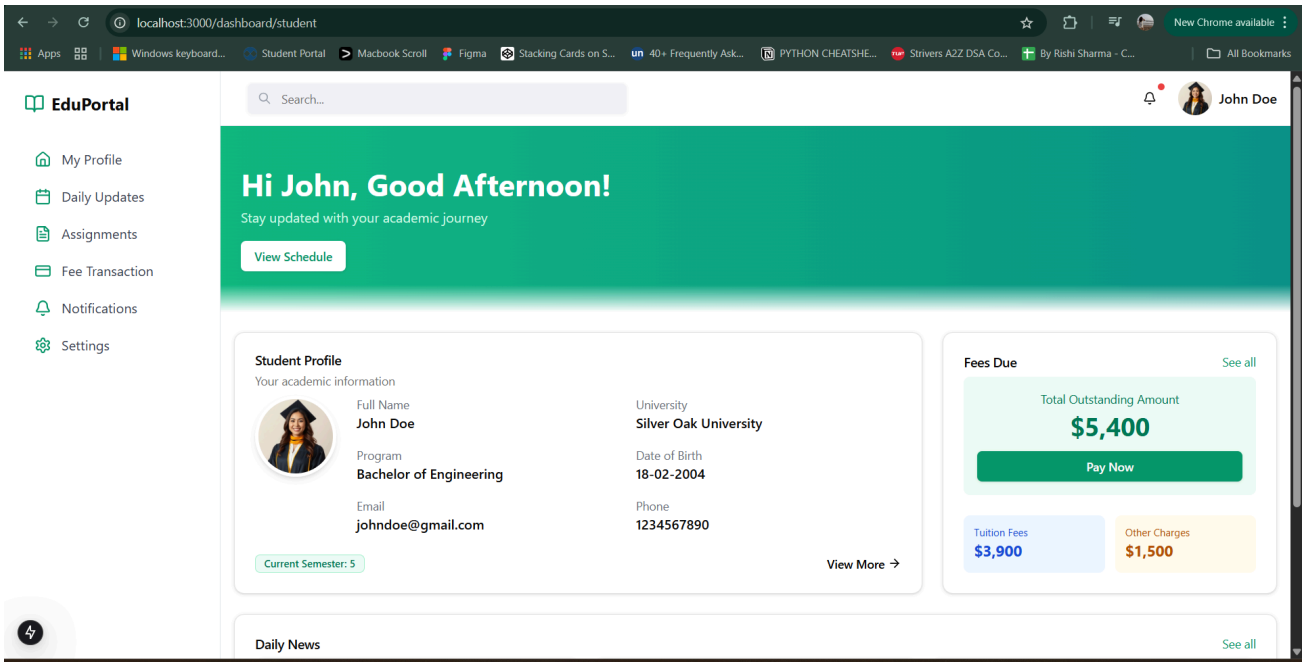
- RESTful APIs: Develop RESTful endpoints for managing user data, events, and notifications. These APIs will handle CRUD (Create, Read, Update, Delete) operations and ensure secure data transactions.
- API Documentation:  Postman to document the APIs, ensuring clear and comprehensive documentation for developers.

## Frontend Development

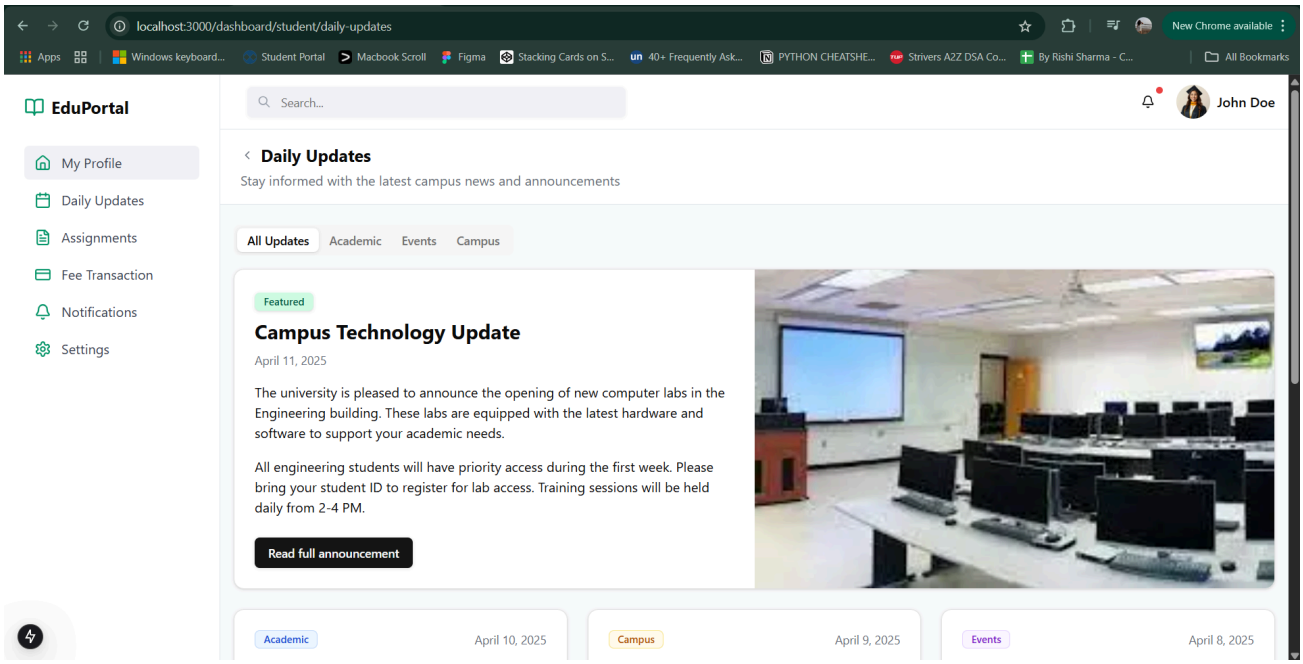
The frontend of the Smart Campus project will be developed using Next.js and Tailwind CSS, ensuring a modern and responsive user interface. Additionally, various libraries will be integrated to enhance functionality and user experience. The Shadcn/UI library will be utilized for implementing sidebars and other UI components, providing modular building blocks for streamlined design. For text editing capabilities, Quill will be integrated as a robust rich text editor. To facilitate scheduling and event management, the React-Calendar component will be employed. Together, these tools will create a dynamic and user-friendly interface for the Smart Campus platform.

# List of Figure in frontend

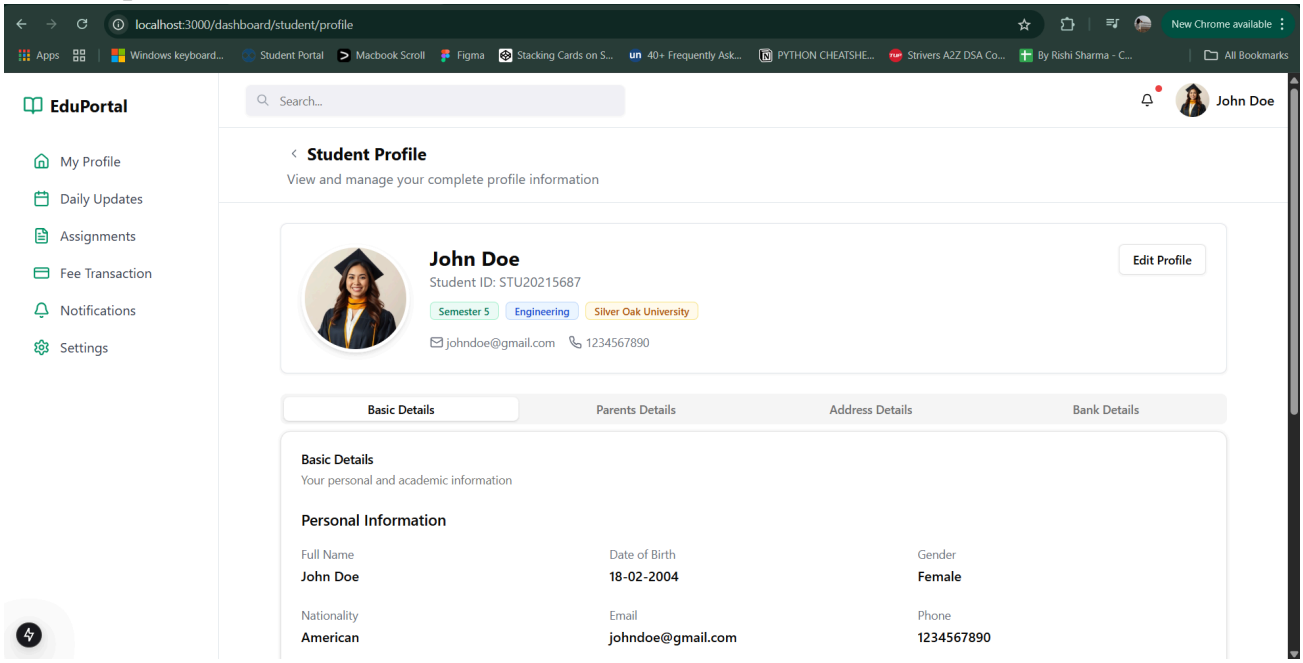
Student dashboard



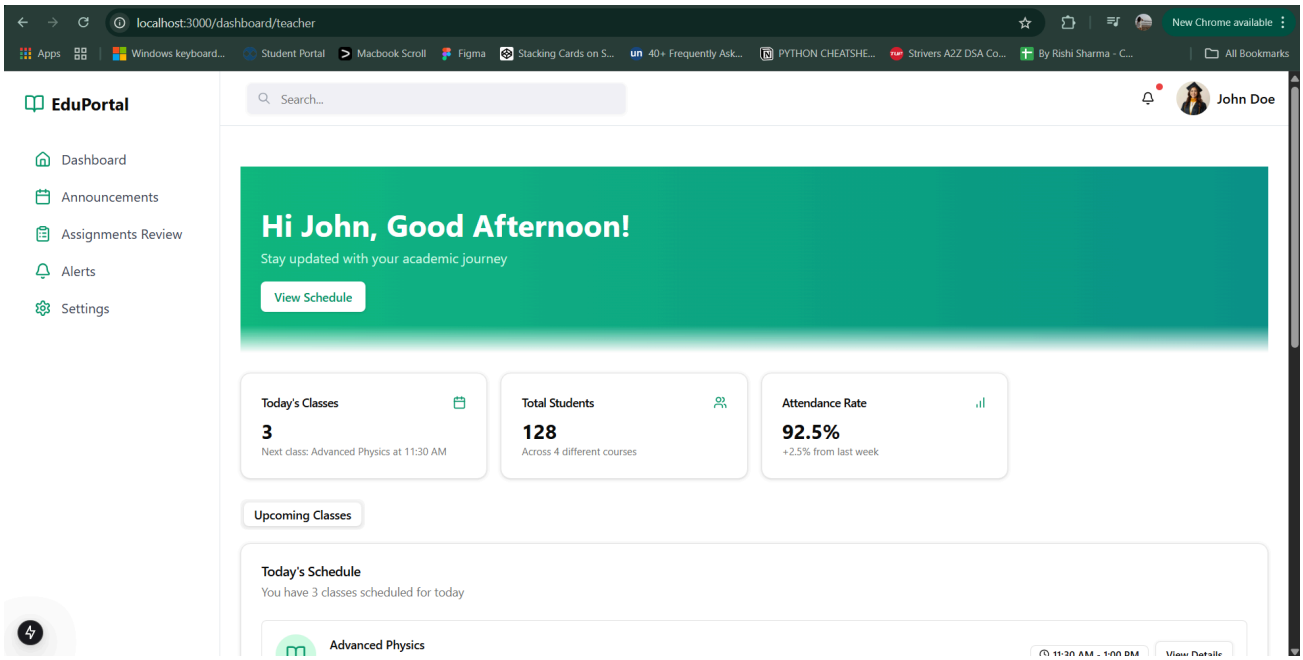

Announcement section in student dashboard



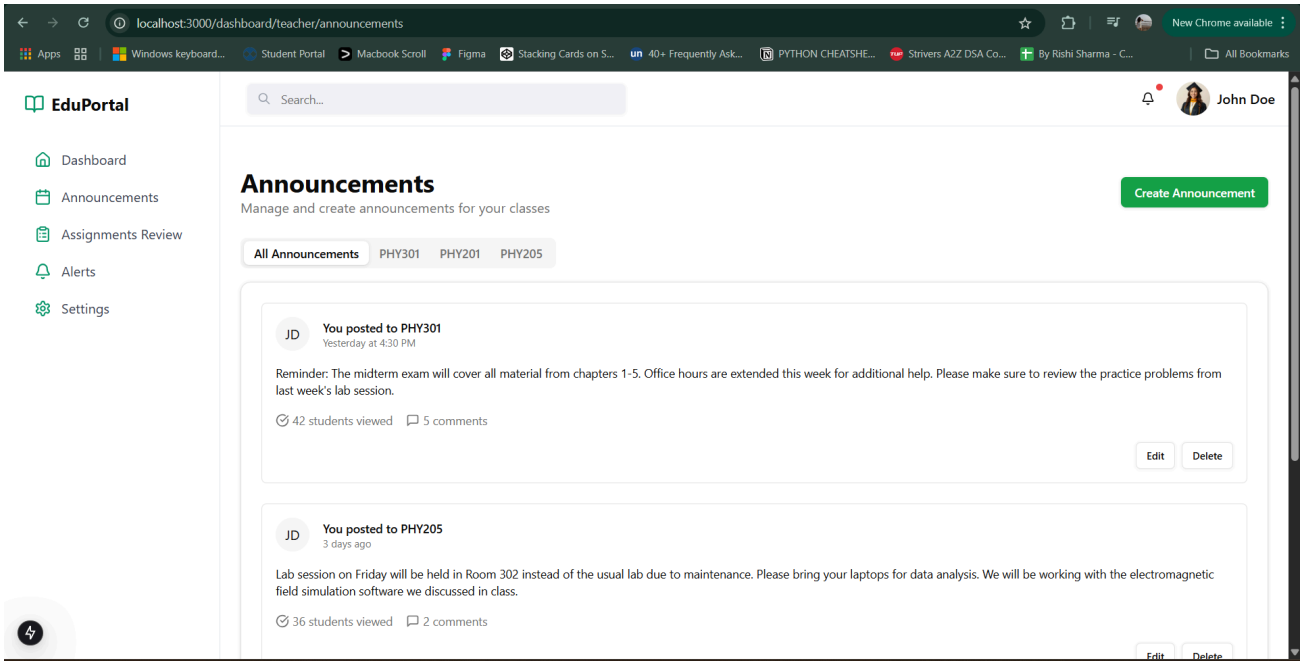
# Student profile



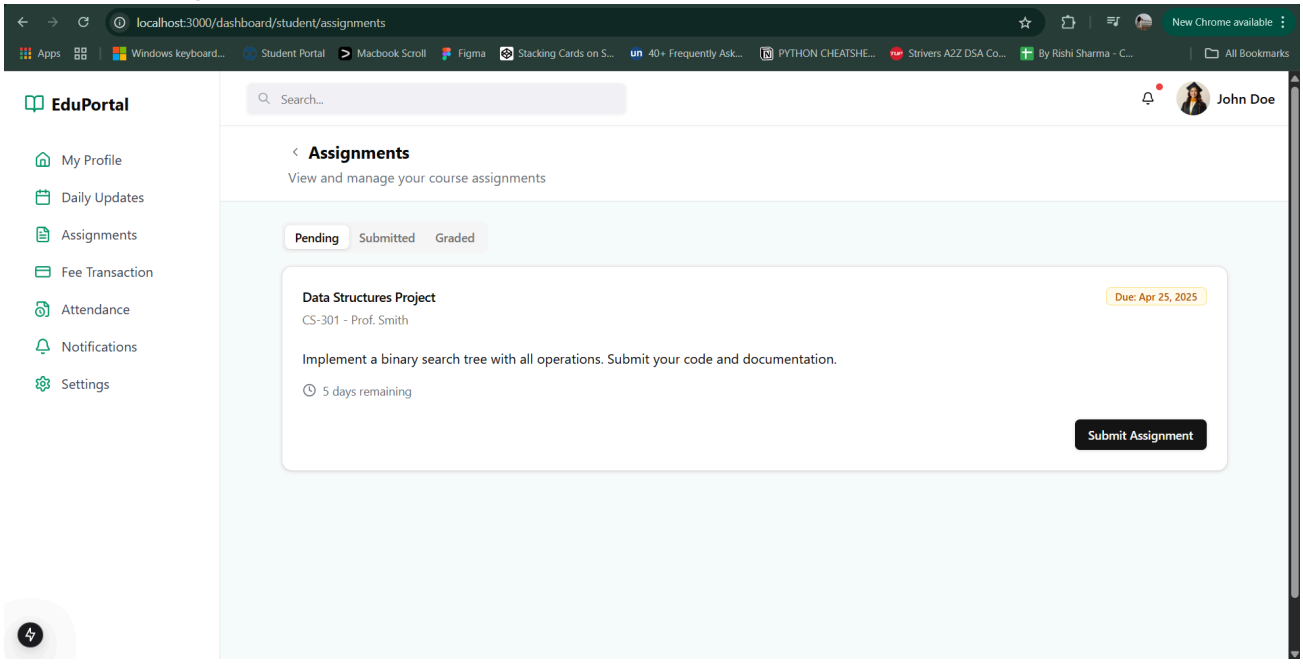

# Teacher Dashboard



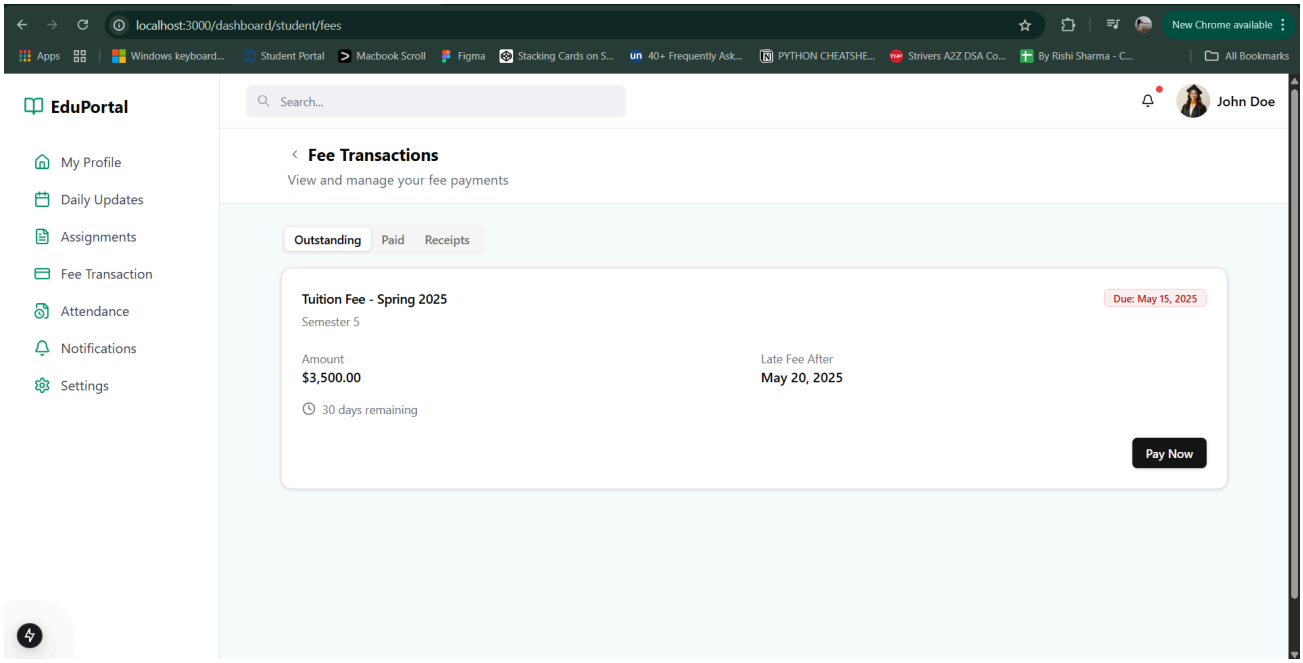
# Teacher announcement



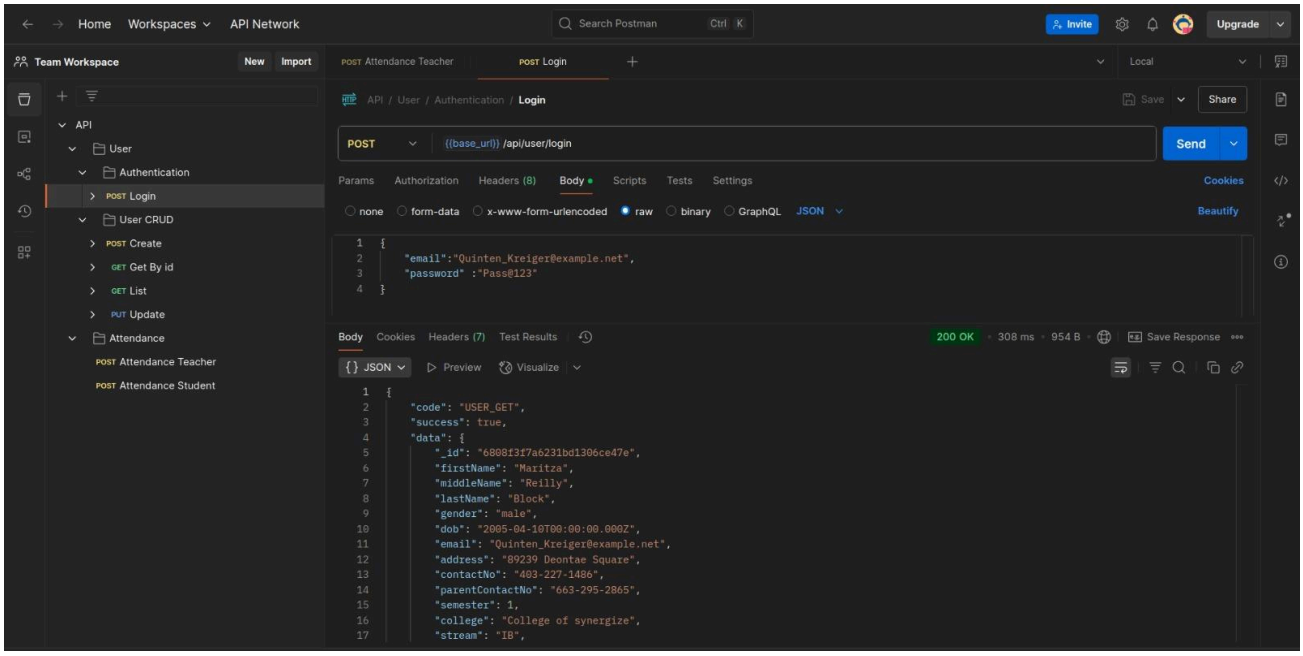

# Student Assignment section



Student Fees section




# List of figures in backend



**References:**

**GitHub repo :** https://github.com/Nebula-2003/smart-cam