|  |
| --- |
| **A**  **PROJECT REPORT ON** |
|  |
|  |
| Student Management System |
|  |
|  |
| SUBMITTED IN  PARTIAL FULFILLMENT OF  **DIPLOMA IN ADVANCED COMPUTING (PG-DAC)** |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| **BY**  **Abhishek Saptale** |
|  |
|  |
| **UNDER THE GUIDENCE OF**  **Nilesh Pawar** |
|  |
|  |
|  |
| **AT**  **SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY,**  **PUNE** |

|  |  |  |  |
| --- | --- | --- | --- |
| **SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY,**  **PUNE.** | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
| **CERTIFICATE** | | | |
|  | | | |
| This is to certify that the project | | | |
|  | | | |
| Student Management System | | | |
|  | | | |
| Has been submitted by | | | |
|  | | | |
| **ABHISHEK SAPTALE** | | | |
|  | | | |
|  | | | |
| In partial fulfillment of the requirement for the Course of **PG Diploma in Advanced Computing (PG-DAC Sep 2023)** as prescribed by The **CDAC** ACTS, PUNE. | | | |
|  | | | |
|  | | | |
| Place: Pune | | | Date: 22-Feb-2024 |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | **Nilesh Pawar** |
|  | | | **Project Guide** |
|  |

**ACKNOWLEDGEMENT**

I would like to express my special thanks of gratitude to my teacher Nilesh Pawar as well as our Director Nitin Kudale who gave me the golden opportunity to do this wonderful project on the topic Student Management System, which also helped me in doing a lot of Research and I came to know about so many new things I am really thankful to them.

Abhishek Saptale

**ABSTRACT**

Developed and maintained a student management system to streamline administrative tasks such as student enrollment, attendance tracking, and grade management. Implemented user-friendly interfaces for easy access and navigation, enhancing the overall user experience.

|  |  |  |
| --- | --- | --- |
|  | **INTRODUCTION** | 1 |
|  | 1.1 Introduction | 2 |
|  | **Product Overview and Summary** |  |
|  | 2.1 Purpose |  |
|  | 2.2 Scope |  |
|  | 2.3 User Classes and Characteristics |  |
|  | 2.4 Design and Implementation Constraints |  |
|  | **REQUIREMENTS** |  |
|  | 3.1 Functional Requirements |  |
|  | 3.1.1 Use case for Administrator. |  |
|  | 3.1.2 Use case for User. |  |
|  | 3.2 Non - Functional Requirements |  |
|  | 3.2.1 Usability Requirement |  |
|  | 3.2.2 Performance Requirement |  |
|  | 3.2.3 Reliability Requirement |  |
|  | 3.2.4 Portability Requirement |  |
|  | 3.2.5 Security Techniques |  |
|  | **PROJECT DESIGN** |  |
|  | 4.1 Data Model |  |
|  | 4.1.1 Database Design |  |
|  | 4.2 Process Model |  |
|  | 4.2.1 Functional Decomposition Diagram |  |
|  | 4.2.2 Data Flow Diagram (DFD) |  |
|  | **PROJECT RELATED STATISTICS** |  |
|  | **CONCLUSION** |  |

**INDEX**

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Section** | **Table Title** | **Page** |
| **Fig 3** | **Complete Database** |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Section** | **Figure Title** | **Page** |
| **Fig 1-2** | **User & Admin Use Case Diagram** |  |
| **Fig 3** | **Data Flow Diagram** |  |
| **Fig 4 -9** | **Dashboard Screen Shots** |  |
| **Fig 10-13** | **Mobile Application Screen Shots** |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**INTRODUCTION**

The Student Management System is a comprehensive software solution designed to streamline academic and administrative tasks within educational institutions. This project aims to enhance efficiency by automating processes such as student enrollment, attendance tracking, grading, and report generation. It provides a user-friendly interface for administrators, teachers, and students to manage information effectively. With features like performance analysis, communication tools, and database management, the system promotes better organization and communication within the academic community. By integrating modern technology into education management, the project seeks to improve overall productivity and enhance the learning experience for all stakeholders.

**The goal of this project:**   
The primary goal of the Student Management System project is to create a centralized platform that facilitates efficient management of student-related information and administrative tasks within educational institutions. This includes goals such as:

1. **Streamlining Administrative Processes:** Simplify tasks such as enrollment, attendance tracking, and grading to save time and resources for administrators and teachers.
2. **Enhancing Communication:** Provide tools for effective communication between administrators, teachers, students, and parents to improve collaboration and information sharing.
3. **Improving Data Accuracy and Accessibility:** Ensure that student records and academic data are accurate, up-to-date, and easily accessible to authorized personnel.
4. **Promoting Transparency and Accountability:** Enable stakeholders to monitor student progress, attendance, and performance transparently, fostering accountability within the educational community.
5. **Enhancing Overall Efficiency:** Increase overall productivity by automating routine tasks, reducing paperwork, and minimizing manual errors.

By achieving these goals, the Student Management System aims to create a more organized, efficient, and student-centric environment within educational institutions.

**Product Overview and Summary**

**| Purpose:** to develop student management system to streamline administrative tasks such as student enrollment, attendance tracking, and grade management. Implemented user-friendly interfaces for easy access and navigation, enhancing the overall user experience.

**| Scope**: The scope of the Student Management System project encompasses a comprehensive suite of features tailored to address the diverse needs of both students and staff within educational institutions.

For students, the system provides functionality to view their attendance records, grades, and fee payment status conveniently through a user-friendly interface. They can access their academic performance data in real-time, allowing for greater self-monitoring and awareness of their progress. Additionally, students can make fee payments online, streamlining the financial transactions process and enhancing convenience.

For staff members, the system offers administrative capabilities such as adding and managing student profiles, recording attendance, and entering grades efficiently. Staff can generate attendance reports, track student progress, and input grades accurately, thereby facilitating effective academic management. The system's administrative tools enable staff to streamline their workflow, reduce manual errors, and devote more time to teaching and student support.

Furthermore, the project scope includes features for administrators to oversee the entire system, manage user permissions, generate reports, and ensure data integrity and security. Administrators have access to comprehensive dashboards and analytics tools to monitor system performance, identify trends, and make informed decisions to optimize academic operations.

Overall, the scope of the Student Management System project encompasses a robust set of functionalities designed to enhance transparency, efficiency, and collaboration among students, staff, and administrators within educational institutions.

**User Classes and Characteristics**: As based on multiple platforms there are multiple classes and their uses but most importantly application uses Model for user that encapsulates data fetched from server like grades , attendance etc

**| Design and Implementation Constraints**

**- User Interface**

In order to effectively incorporate all focus areas of our application into the user interface without creating too much clutter, we opted to use a tabbed design. When the app is initially opened, the user is taken to the “home” tab.

**- Tab Design**:

The Student Information System is structured with several tabs to facilitate efficient navigation and access to key functionalities. The Dashboard tab serves as a central hub, offering an overview of important statistics and system notifications, along with quick links to essential features and modules. Within the Student Profile tab, users can view comprehensive details such as personal information, academic records, attendance summaries, and fee payment history. The Attendance tab allows students to track their attendance records for individual courses or classes, with options to filter by date or course. Similarly, the Grades tab provides a breakdown of grades for assignments, quizzes, and exams, along with cumulative GPA calculations. Students can manage their fee payments conveniently through the Fee Payment tab, which includes an online payment portal, statement of account, and receipt generation features.

**Functional Requirements**

**| Use Case for User**:

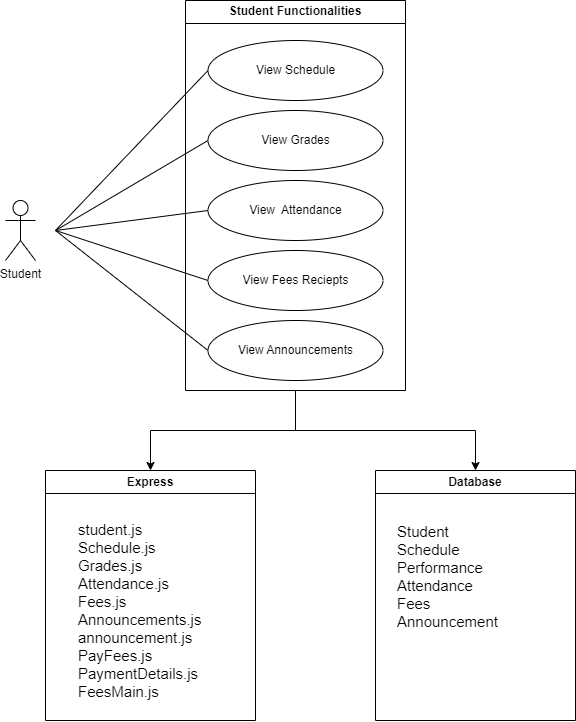


Fig. 1

**| Use Case for Staff**:

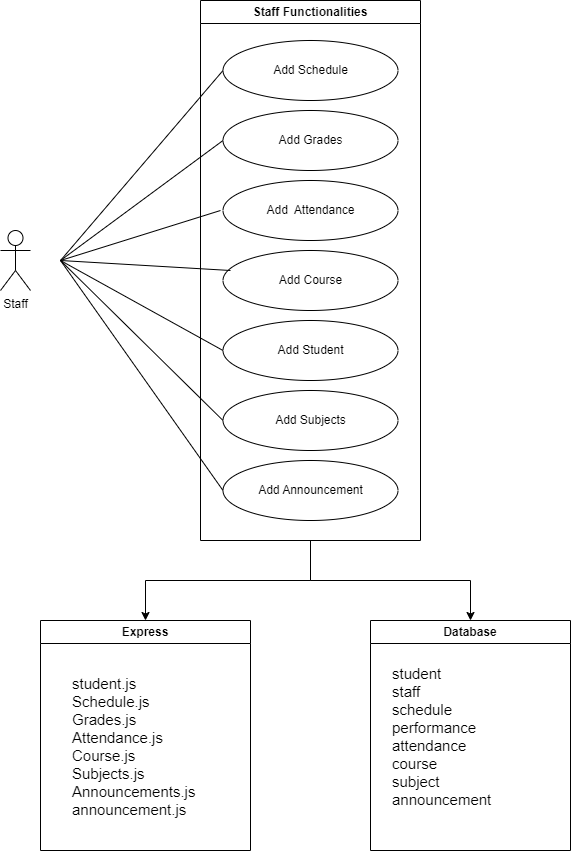


Fig. 2

**Non - Functional Requirements**

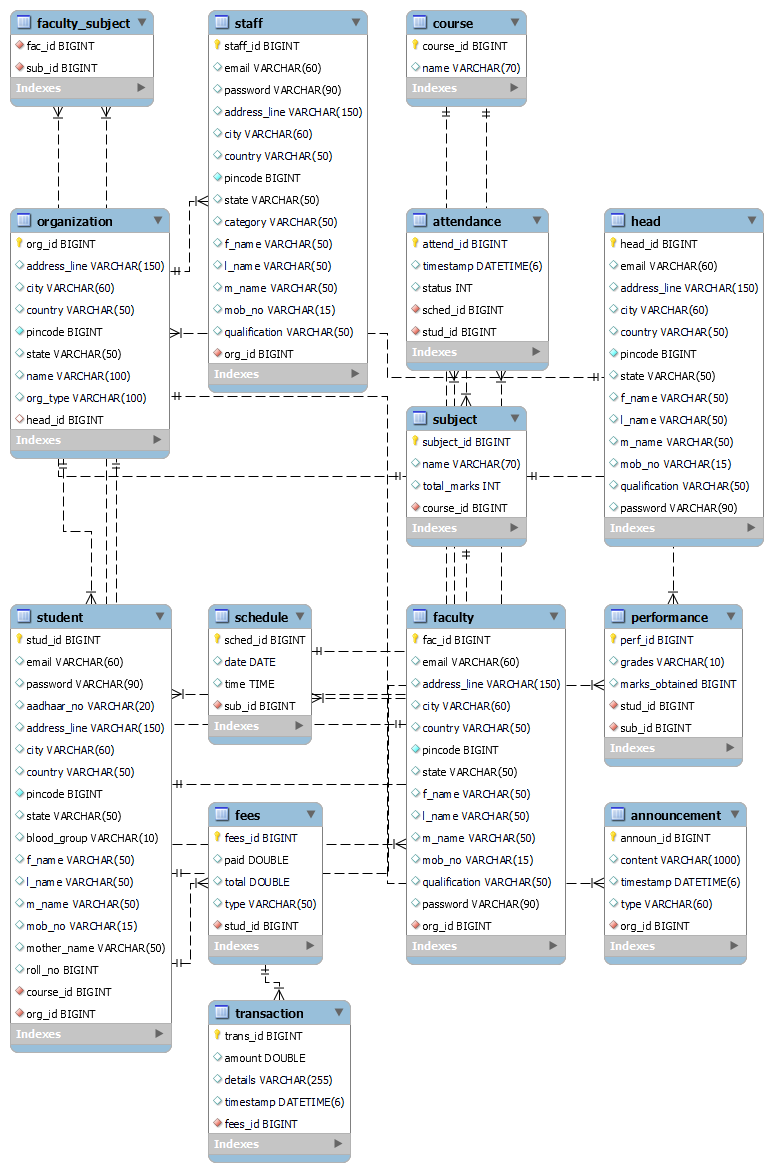
**Usability Requirement:** Application should be easy to use and provide basic user interface that can be used without any tutorial.

For this idea to be modular, several views must be used; by this, I mean the application's simplicity of use. The purpose of using a mobile application is to obtain certain features and functionalities; without consideration for usability, the application would be challenging to use. Every programme should be efficient, intelligent, and satisfying. It should also adhere to certain standards that developers should follow, such as maintaining colour and contrast integrity. The application should be designed such that users of various skill levels may effectively use the user interface.

Additionally, people with various disabilities, such blindness, limited vision, or hearing impairment, ought to be able to use the apps. As an alternative to the visual implementation, developers should also consider the app's acoustic implementation. Always try to avoid making unnecessary noises, and make sure the sounds you use to interpret text or screen elements are accurate or nearly accurate.

**Data Model**

For database in order to effectively store the user’s attendance, personal information and Grades tracking history, we decided to use a MYSQL database, which is already integrated in spring boot web Application backend by adding MySql-connector dependency. Our database has 14 tables, one separate table for every entity used in our project such as student entity, staff entity and so on. The Database Schema consists of table for every entity and functionality data. The first step in implementing the database was to create objects that could be used to store and retrieve information for each data type. More specifically, we created entity classes and required data transfer object classes. Each class has multiple variables, as well as methods for setting and getting these variables. As seen in fig 3, the entity classes are converted into tables in the database and the fields in the class are the columns in the tables. Most of these tables have a primary key which is in most of the cases autogenerated using the spring boot JPA and annotation (like @id and (generation type =identity)). We used controllers, Service, repository and entity classes at the backend to do data transfer and used restful APIs to send the data from backend to front end and vice-versa. We used react at the front-end which consisted of functional components to render the response and axios to get the data from the backend. This retrieved data is then rendered into a list, table or card format as per the requirement on the UI of the webpage.



Complete Database (fig 3)

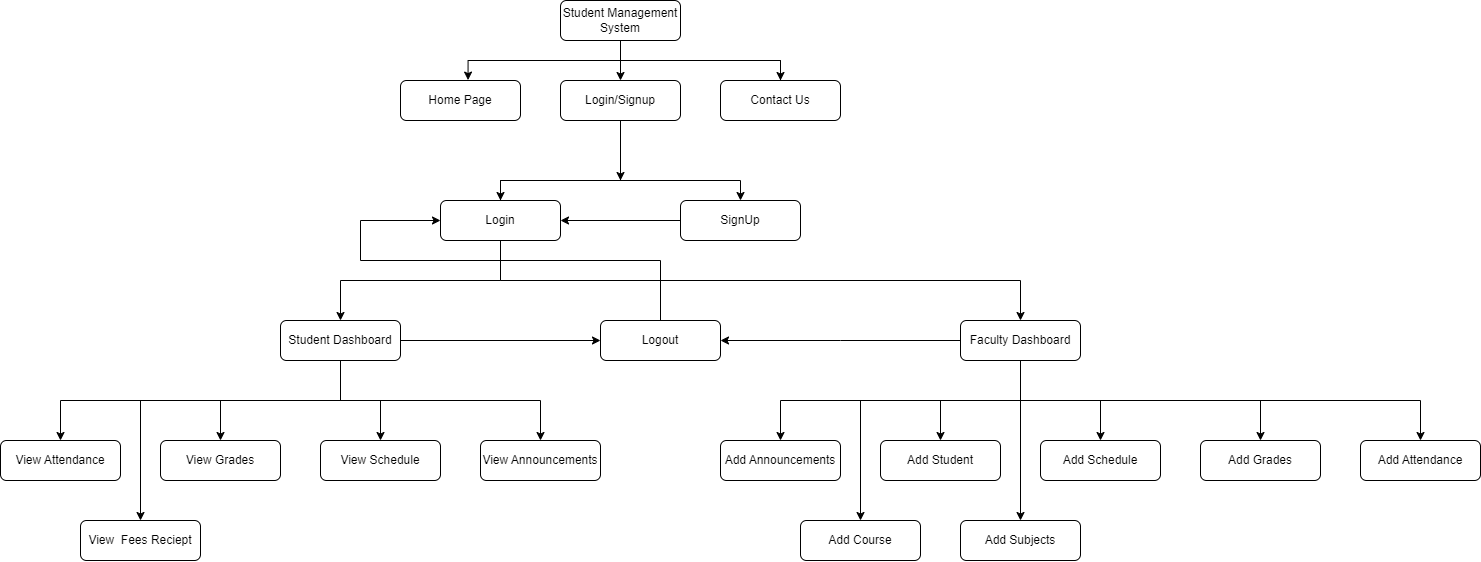


Fig. 3

**Screen Shots**

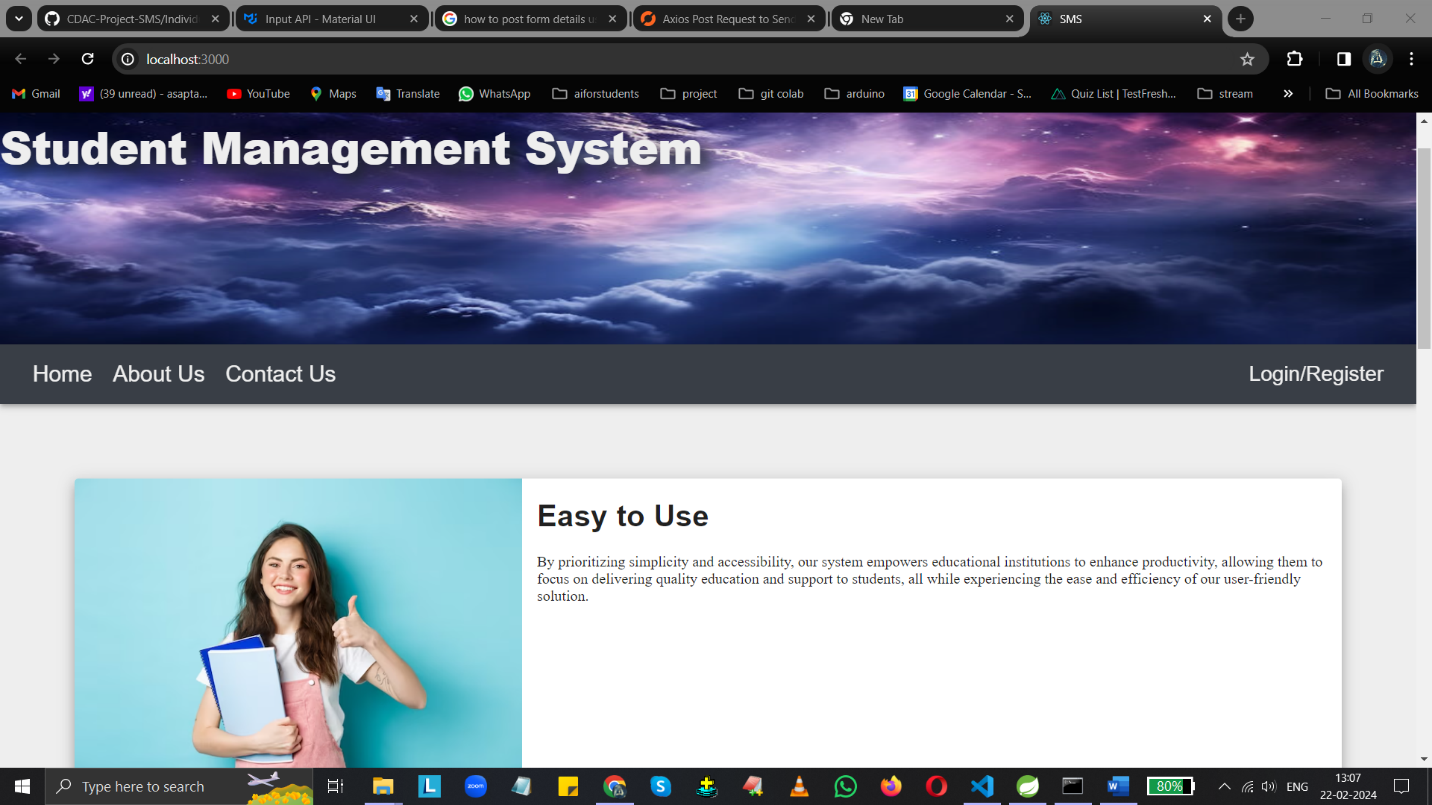


Fig. 4

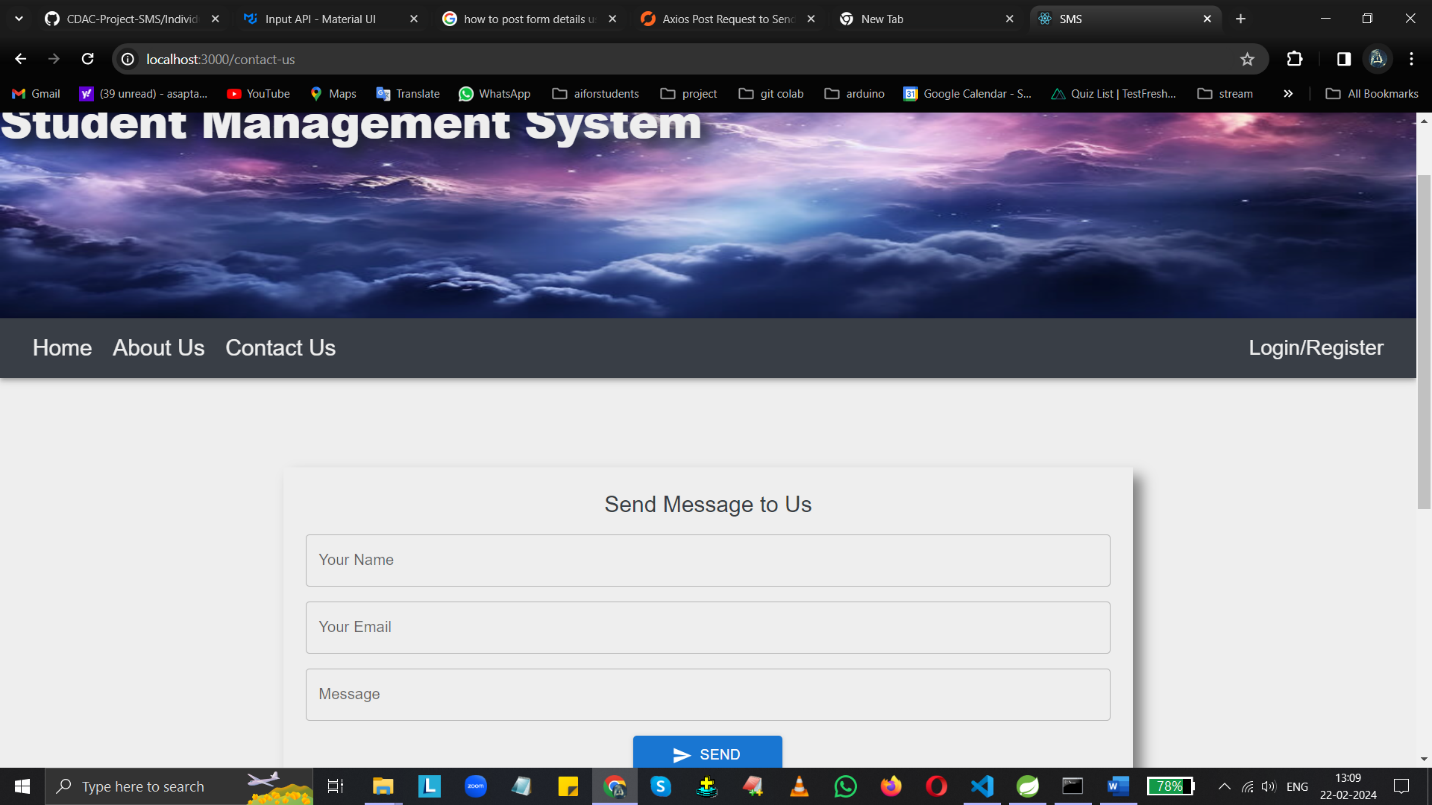


Fig. 5

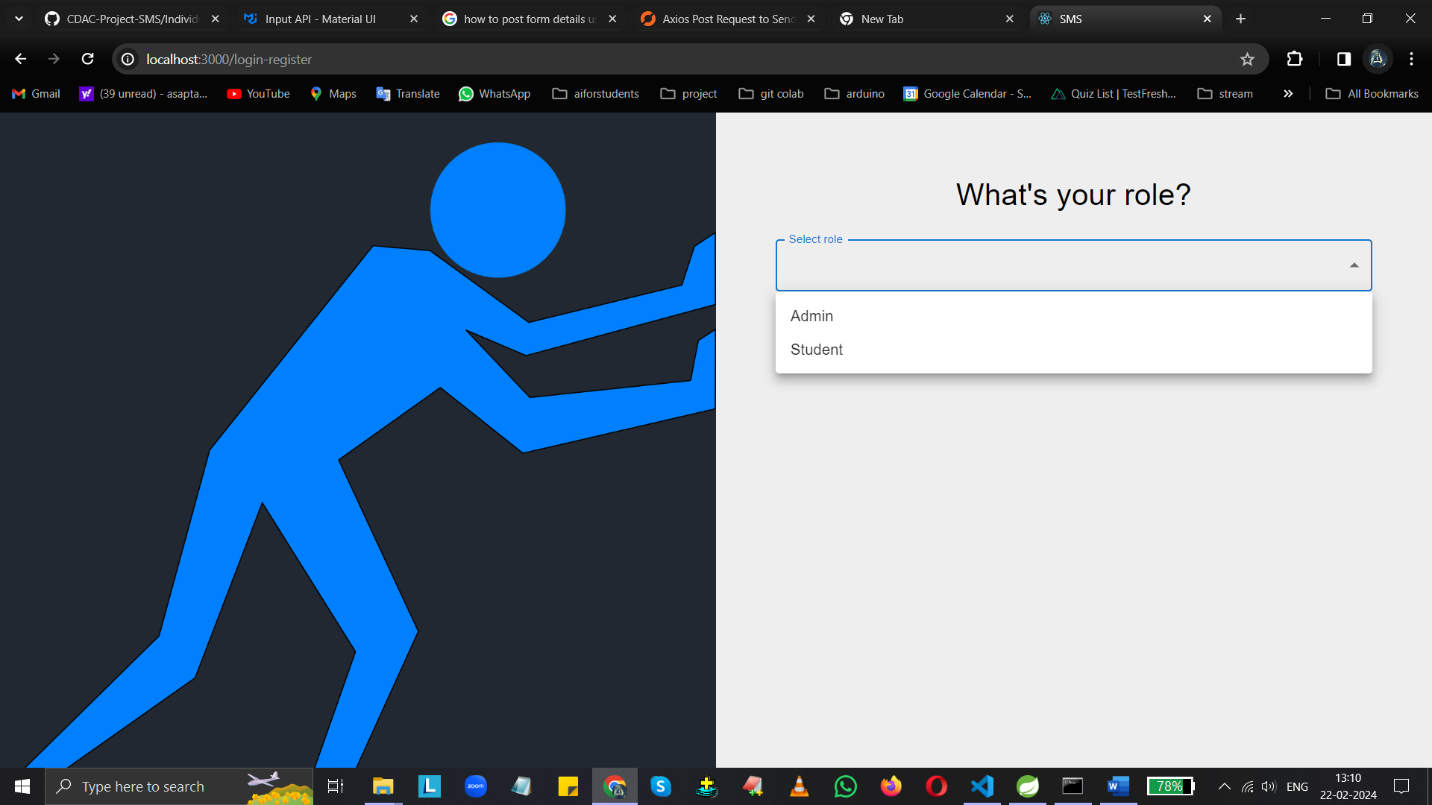


Fig. 6

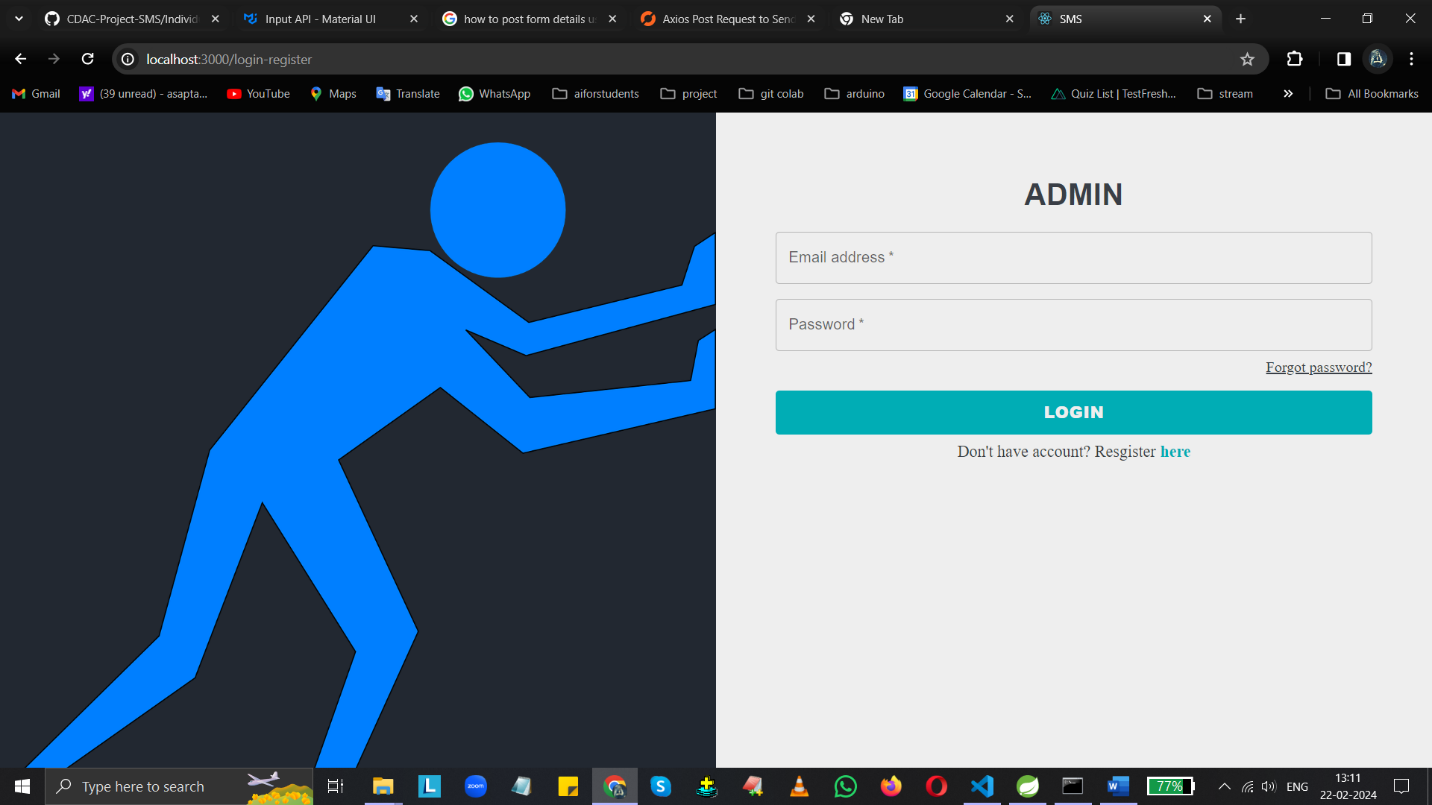


Fig. 7

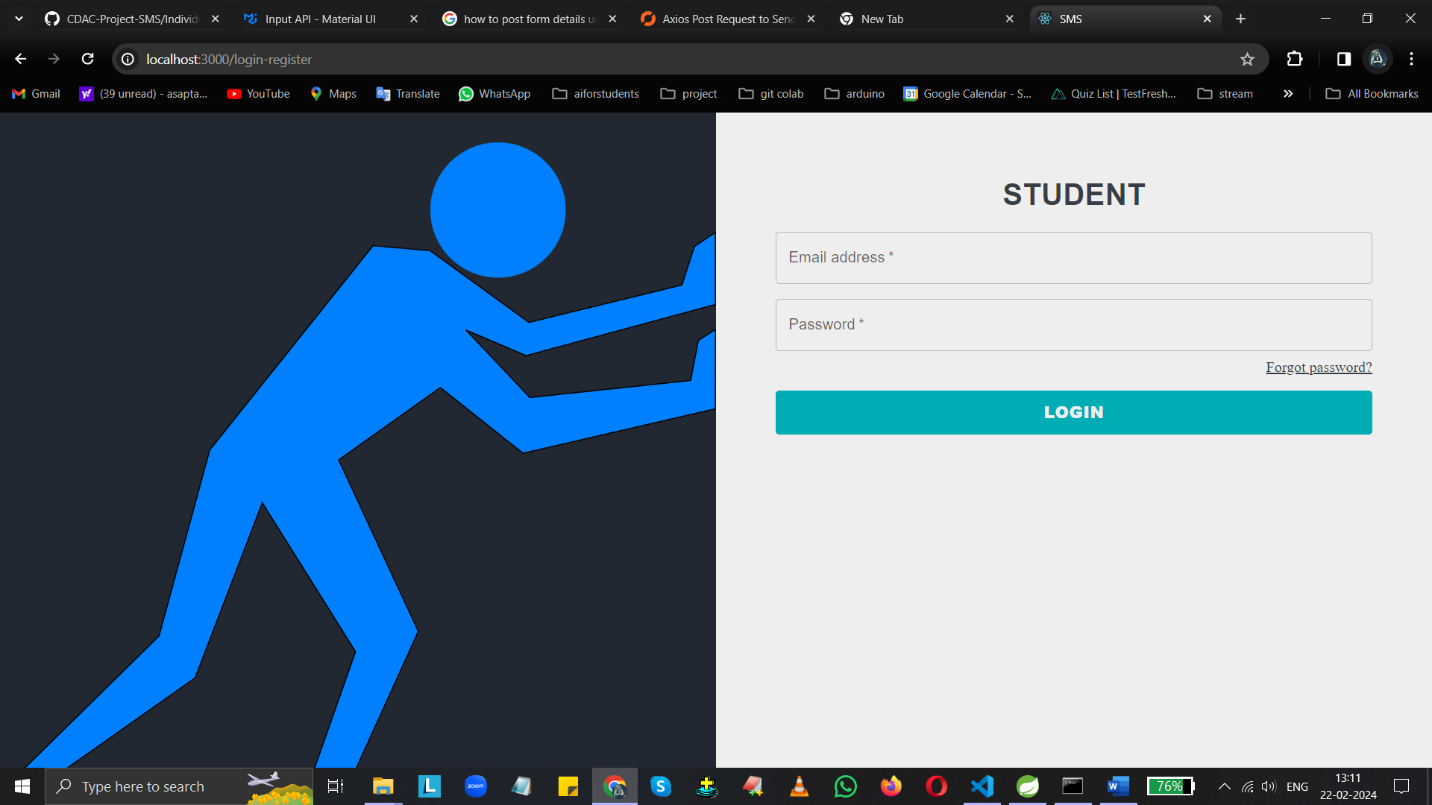


Fig. 8

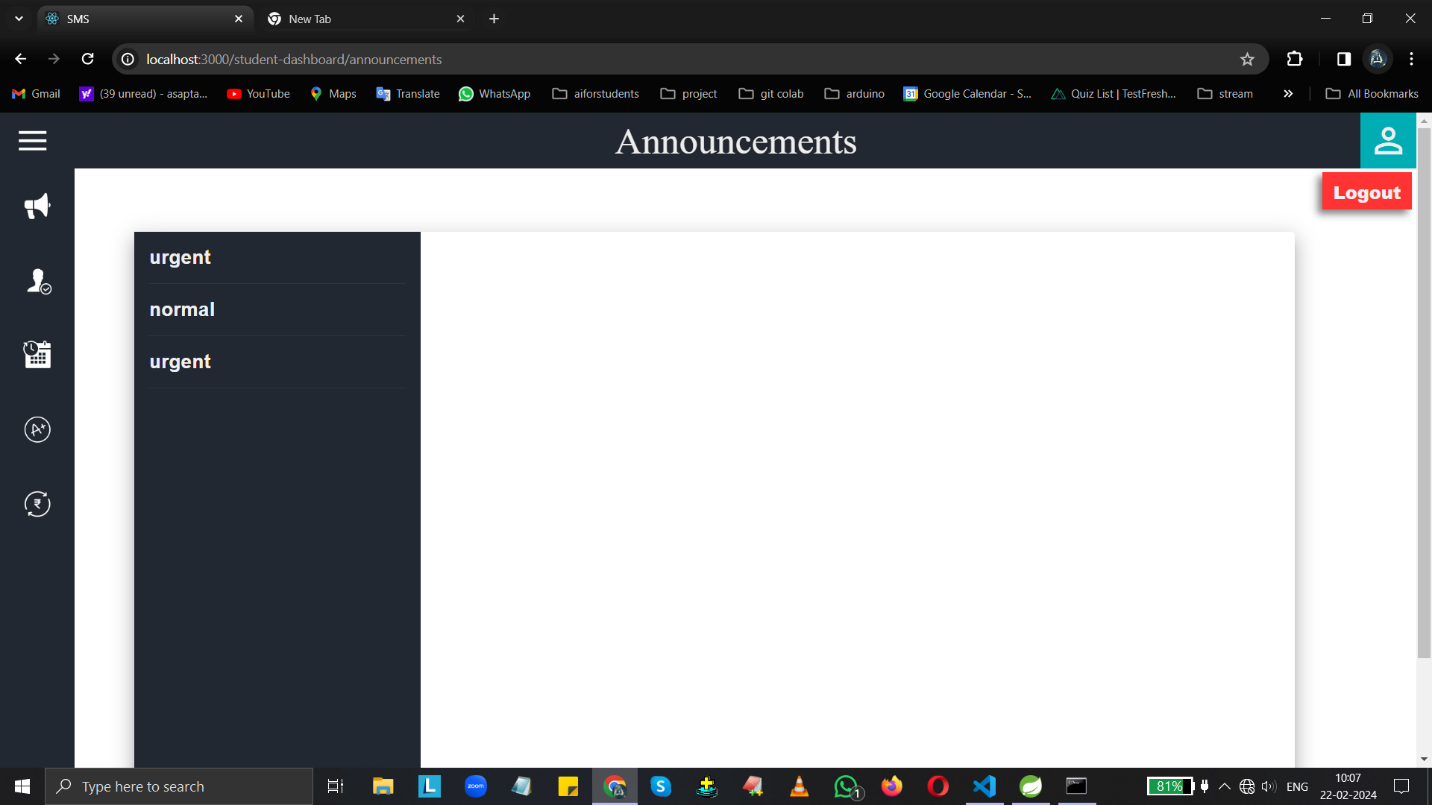


Fig. 9

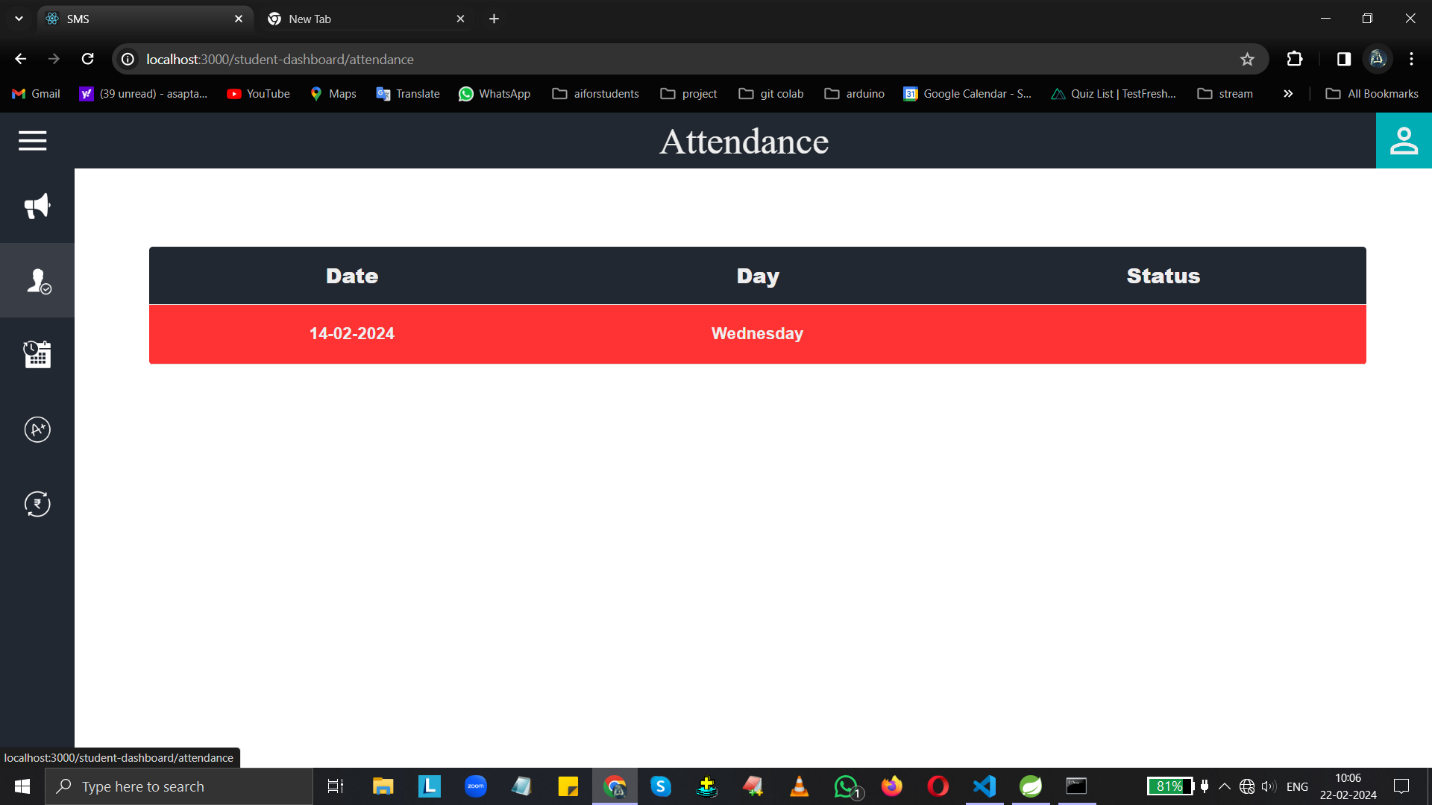


Fig. 10

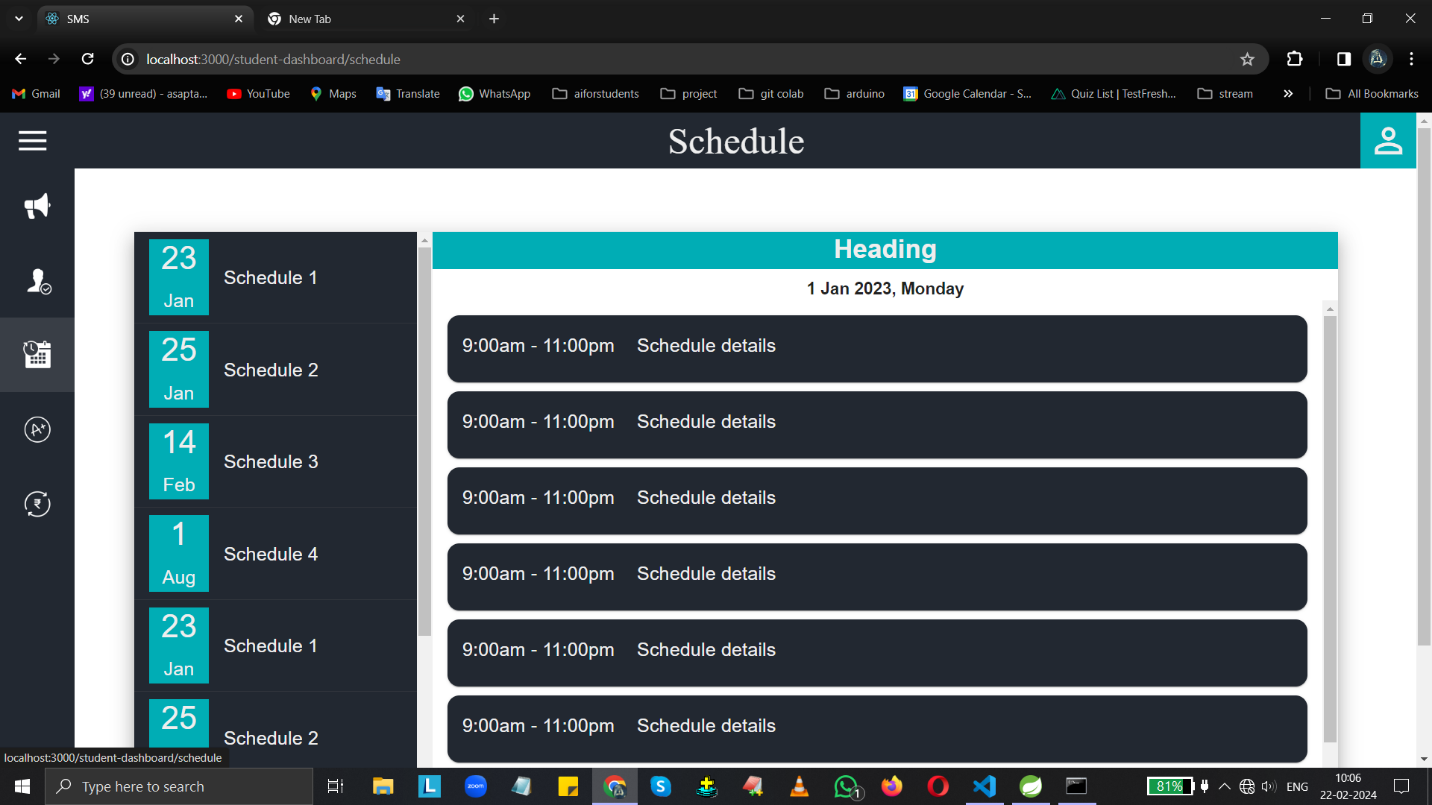


Fig. 11

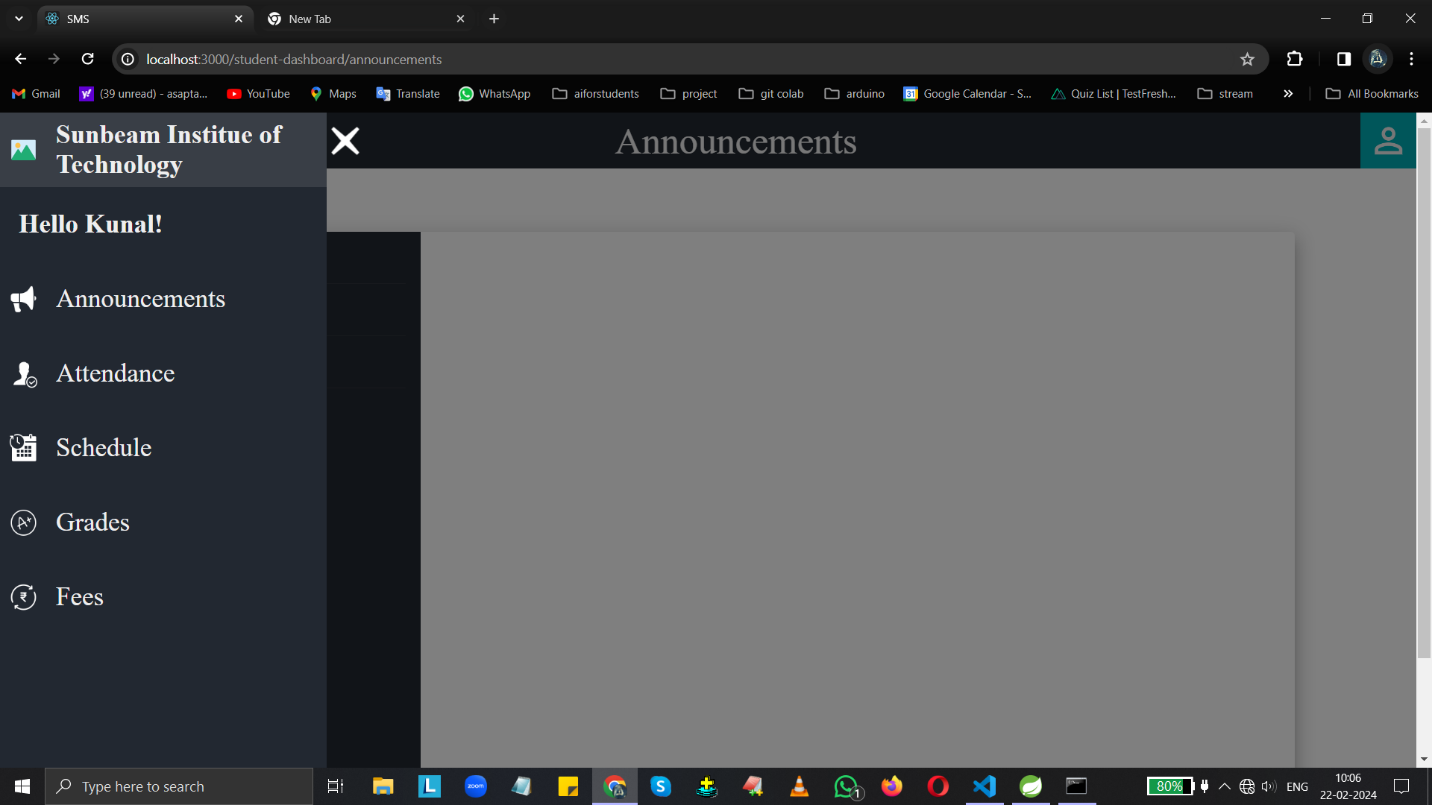


Fig. 12

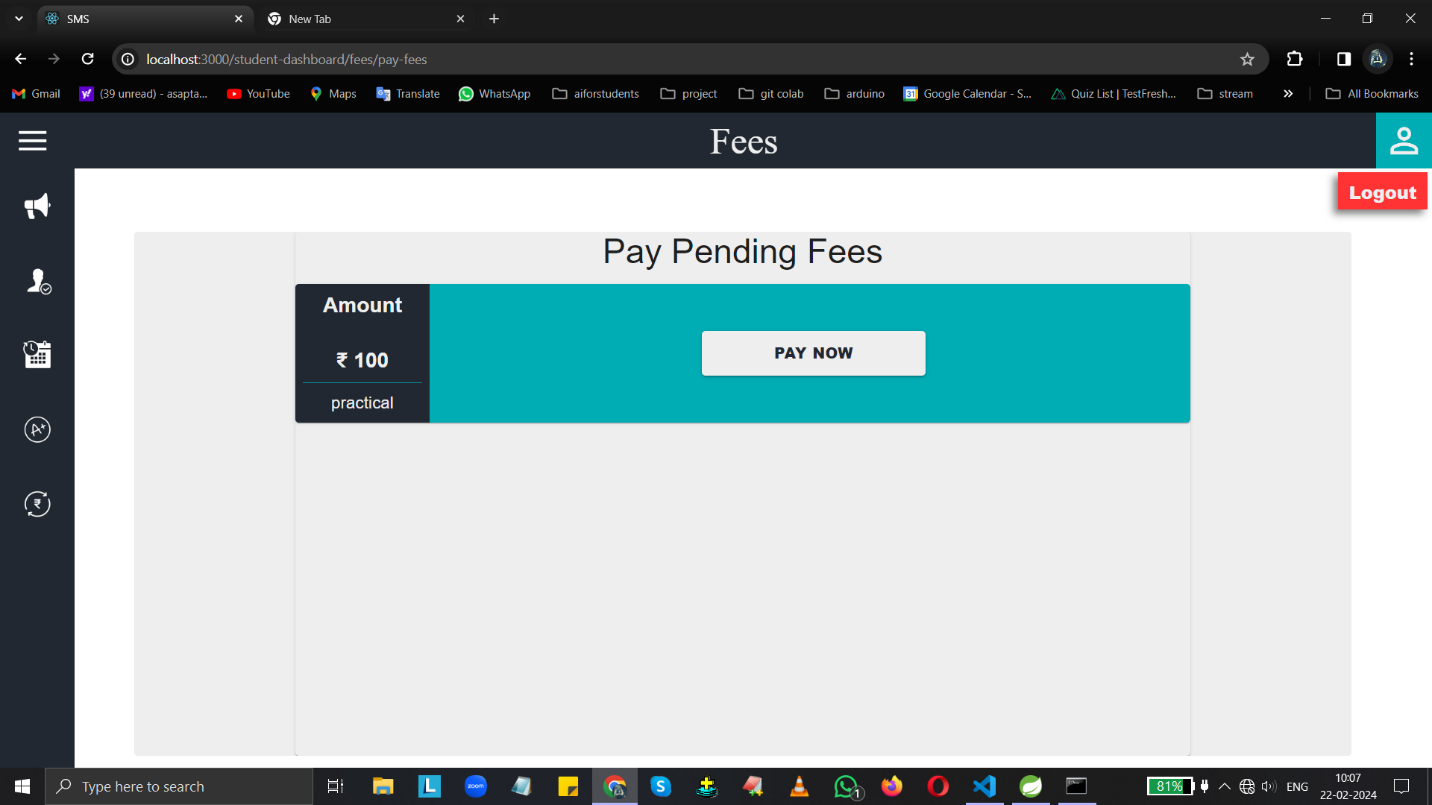


Fig. 13

**| Conclusion:**  Accomplishments While it was a challenge to develop, our project, team successfully created a prototype Student management web application capable of tracking, recording, and displaying data relevant to a user’s attendace, marks, and fees. While the full scope of the initial app design was not realized, all of the core data tracking functionality has been successfully implemented

**| Future Work:** While I feel that I’ve successfully implemented the desired Student management functionalities, there are still several elements that could be added to improve the effectiveness of the app. Another possible improvement is to integrate the feature of automated email or text based messaging for notification alerts. While we have already devoted a significant portion of our development time to writing and testing these features, theoretically, they can always be improved. The attendance tracking would likely be the most in need of improvement, due to the difficulty of integrating biometric device that could track and record the attendance data, would contribute tremendously to the accuracy of the attendance data.

**| References:**

<https://swagger.io/docs/open-source-tools/swagger-ui/usage/installation/> swagger Ui documentation

<https://docs.github.com/en> github documentation

<https://start.spring.io/> spring initializer webpage for generating template

<https://mvnrepository.com/> maven repository

<https://mui.com/material-ui/getting-started/> material UI documentation

<https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/> spring boot official documentation

<https://axios-http.com/docs/api_intro> Axios documentation

<https://spring.io/guides/gs/rest-service-cors> spring boot cross origin resource sharing documentation

<https://legacy.reactjs.org/docs/create-a-new-react-app.html> react documentation

<https://react-redux.js.org/introduction/getting-started> react redux documentation