College website(UMS)

Software Requirements Specification

23-04-2024

Pushkar Varshney

Prepared for

Continuous Assessment 3

2024

Table of Contents

1. Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 Definitions, Acronyms, and Abbreviations 5

1.4 Overview 5

2. General Description 6

2.1 Product Functions 7

2.2 User Characteristics 8

2.3 General Constraints 8

2.4 Assumptions and Dependencies 8

3. Specific Requirements 9

3.1 Functional Requirements 9

3.2 Non-Functional Requirements 10

3.3 Design Constraints 10

4. Analysis Models 11

4.1 Screenshots…………………………………………………………………………………………………………………………………………..11

4.2 Data Flow Diagrams (DFD) 13

5. Github link…………………………………………………………………………………………………….15

A. Appendices 15

A.1 Appendix 1 15

Table of Figures

|  |  |
| --- | --- |
| Fig No.: Detail | Page No. |
| Fig.1: Home page | 11 |
| Fig.2: Time table | 12 |
| Fig.3: Exam Schedules | 12 |
| Fig.4: Community section | 13 |
| Fig.5: Notes Downlaod | 13 |

Introduction

The project is a web-based application developed to enhance the management of student-related activities within our college. This system provides a centralized platform for students to efficiently handle various aspects of their life. It offers features such as attendance tracking, exam schedules, announcements, and collaborative group discussions, all aimed at improving the overall student experience.  
  
With a simple and intuitive interface built on Angular, students can easily access their attendance records, exam results, and important announcements. The system also includes a personalized timetable feature for organizing daily schedules. Additionally, real-time updates on teacher availability due to leave and a notes downloader for course materials enhance the learning process.

* 1. Purpose

The Student College Account Management System aims to streamline student-related activities by providing a centralized platform for attendance tracking, exam schedules, announcements, and group discussions. This project intends to enhance the student experience within our college, fostering better communication and organization. The system is designed to be user-friendly for students, teachers, and administrators, offering easy access to essential information and resources to support academic success and efficiency in daily tasks.

* 1. Scope

**The software product we're creating is the Student College Account Management System.**

**What it Will Do**

* **Attendance Tracking: Keep a record of our class attendance.**
* **Exam Schedule Management: Display upcoming exam schedules and results.**
* **Announcements Display: Show important college news and events.**
* **Timetable Creation: Allow us to create our personalized timetables.**
* **Teacher Leave Updates: Notify us about teacher leaves to manage our expectations.**
* **Notes Downloading: Access course materials for studying.**
* **Community Group Feature: Join group discussions for learning and support.**

**What it Will Not Do**

* **The system will not manage financial transactions or grades.**

**Application**

**This system is designed to make our college life easier and more organized. It will help us keep track of attendance, exams, announcements, and manage our study schedules efficiently. Our main goal is to improve our academic performance and make collaboration with classmates smoother. This system aims to provide us with easy access to necessary information and resources for a better college experience.**

**Present I make this project for the Frontend. So, it doesn’t have any backend part like server. But in future I will add it in my project to make it better.**

## Definitions, Acronyms, and Abbreviation

* Definitions, Acronyms, and Abbreviations  
  SRS: Software Requirements Specification - This document outlining the requirements and specifications of the Student College Account Management System.  
  HTML: Hypertext Markup Language - The standard language for creating web pages and applications.  
  CSS: Cascading Style Sheets - Used to style the visual presentation of HTML elements.  
  JS: JavaScript - A programming language used to add interactivity and dynamic behavior to web pages.  
  Angular: A JavaScript framework for building web applications, including our Student College Account Management System.  
  UI: User Interface - The visual elements and layout of our system that users interact with.  
  Bootstrap: A front-end framework for developing responsive and mobile-first websites and web applications.  
  PDF: Portable Document Format - A file format used for documents, such as notes and other downloadable resources within our system.  
  URL: Uniform Resource Locator - Web address used to access specific resources within our system.  
  HTTP: Hypertext Transfer Protocol - The protocol used for transferring data over the web, such as fetching notes or accessing announcements.  
  API: Application Programming Interface - Defines interactions between our Angular application and other software components, such as retrieving data from the database.  
  UX: User Experience - The overall experience and satisfaction a user has when interacting with our system, influenced by its design and functionality.
  1. Overview

This project is a web-based application developed to simplify and enhance the management of student-related activities within our college. Built using HTML, CSS, JavaScript (JS), Angular, and Bootstrap, this system offers a user-friendly interface for students, teachers, and administrators. It includes features such as attendance tracking, exam schedules, announcements, timetable management, teacher leave updates, notes downloading, and community group discussions. The system aims to improve the overall student experience by providing easy access to essential information and resources. This Software Requirements Specification (SRS) outlines the detailed requirements and specifications of the system, ensuring a clear roadmap for its development and implementation.

1. General Description

 The Student College Account Management System is influenced by various factors that shape its requirements and design. These factors contribute to the system's functionality and usability, ensuring an effective solution for managing student-related activities within our college.

Technological Environment

* Web-Based Application: As a web-based application, the system relies on HTML, CSS, JavaScript (JS), Angular, and Bootstrap to create an interactive and responsive user interface accessible via web browsers.
* Cross-Browser Compatibility: The system is designed to work seamlessly across different web browsers such as Chrome, Firefox, Safari, and Edge, ensuring accessibility for all users.

User Interaction

* User-Friendly Interface: The system's interface is designed with simplicity and intuitiveness in mind, allowing students, teachers, and administrators to navigate and use its features with ease.
* Responsive Design: Utilizing Bootstrap, the system adapts to various screen sizes, including desktops, tablets, and mobile devices, providing a consistent user experience.
* Real-Time Updates: Features such as teacher leave updates and announcements display ensure users have timely and relevant information.

Functional Requirements

* Attendance Tracking: The system allows students to view and monitor their class attendance records, promoting accountability and engagement.
* Exam Schedule Management: Providing access to exam schedules and results aids in effective exam preparation and planning.
* Announcements Display: Centralized display of important college news and events keeps users informed and up-to-date.
* Timetable Creation: Personalized timetables help students organize their schedules efficiently, optimizing their study and leisure time.
* Notes Downloading: Accessing downloadable course materials enhances learning opportunities outside the classroom.
* Community Group Feature: Group discussions foster collaboration and support among students, creating a sense of community within the college.

Scalability and Maintenance

* Scalable Architecture: The system is designed with scalability in mind, allowing for future expansion and addition of new features.
* Ease of Maintenance: Well-structured code and documentation facilitate system maintenance and updates, ensuring its longevity and reliability.

These general factors provide a foundation for understanding the requirements and design of the Student College Account Management System. They shape the system's development to meet the needs of its users while ensuring a robust, user-friendly, and reliable solution for managing student activities within the college environment.

* 1. Product Functions

This project encompasses a range of functions designed to enhance the college experience for students. The software will:

Attendance Tracking:

* Allow users to view and track their attendance records for various courses.

Exam Schedule Management:

* Provide a platform to access and manage upcoming exam schedules and results.

Announcements Display:

* Display important college announcements and events prominently for all users.

Timetable Creation:

* Enable users to create personalized timetables based on their courses and schedules.

Teacher Leave Updates:

* Notify students about teacher availability, including leave statuses and substitutes.

Notes Downloading:

* Offer downloadable course materials for students' reference and study.

Community Group Feature:

* Facilitate group discussions and interactions among students for collaborative learning.

User Profile Management:

* Allow users to manage their profiles, including personal information and settings.

Search Functionality:

* Provide a search feature to quickly find specific information within the system.

Responsive Design:

* Ensure the application is responsive and accessible across various devices.

Accessibility Options:

* Include accessibility features to support users with disabilities.

These functions collectively aim to streamline and improve the management of student-related activities within the college environment, enhancing communication, organization, and collaboration among users.

* 1. User Characteristics

The users of this project will include:

Students: As the primary users, students will interact with the system to manage their attendance, access exam schedules, download course materials, and participate in group discussions.

Teachers: Teachers will use the system to update leave statuses, view their schedules, and access student attendance records.

Administrators: Administrators will oversee the system, manage user accounts, and ensure smooth operation.

Technical Support: There will be a need for technical support staff to address any issues that users may encounter while using the system.

* 1. General Constraints

Technology Stack: The project is constrained by the technologies used: HTML, CSS, JavaScript, Angular, and Bootstrap.

Timeline: The project must be completed within the semester timeframe, limiting the scope of features and design changes.

Resource Limitations: Limited resources may affect the depth and complexity of certain functionalities.

* 1. Assumptions and Dependencies

Assumption: It is assumed that users have basic computer literacy and familiarity with web applications.

Dependency: The availability of a stable internet connection is crucial for accessing and using the system effectively.

Assumption: The project assumes that the college infrastructure supports the deployment of a web-based system.

Dependency: The project's success depends on the availability of necessary data, such as exam schedules and teacher leave information.

Assumption: It is assumed that user feedback will be collected for iterative improvements, enhancing the system's usability.

Identified Drawbacks

The community page is not currently dynamic, limiting real-time interaction among users.

Angular has not been fully integrated with the existing HTML pages, which may affect the seamless transition and user experience between different sections of the system.

Mitigating Drawbacks

Future iterations will focus on making the community page dynamic, enabling users to engage in real-time discussions and collaboration.

Integration of Angular with existing HTML pages will be prioritized to ensure a cohesive and seamless user journey throughout the system.

These identified drawbacks are opportunities for improvement and development in future iterations of the project, aimed at enhancing user engagement and system functionality.

1. Specific Requirements

1. User Authentication

* The system shall require users to log in with a username and password.
* Users shall have the option to reset their password through a secure email verification process.
* Failed login attempts shall trigger account lockout for a specified duration.

2. Attendance Tracking

* Students shall be able to view their course-wise attendance records.
* Teachers shall have access to attendance records for the courses they teach.
* The system shall display attendance percentage for each course.

3. Exam Schedule Management

* The system shall display upcoming exam schedules for each course.
* Users shall be able to download exam schedules as PDF files.
* Students shall receive notifications for upcoming exams.

4. Announcements Display

* Important college announcements shall be displayed on the home page.
* Announcements shall include details such as event dates, times, and locations.
* Users shall be able to dismiss announcements once they have been read.

5. Timetable Creation

* Students shall be able to create personalized timetables by selecting courses.
* The system shall display the timetable in a weekly format with course names and times.
* Users shall have the option to print or download their timetables.

6. Teacher Leave Updates

* Teachers shall be able to update their leave status through their accounts.
* Students shall receive notifications about teacher leave and substitute arrangements.
* The system shall display teacher leave information on the timetable.

7. Notes Downloading

* Course materials such as lecture notes, presentations, and study guides shall be available for download.
* Users shall be able to download notes in PDF format.
* Notes shall be categorized by course and topic for easy access.

8. Community Group Feature

* Users shall be able to create and join community groups for discussions and collaboration.
* Group creators shall have the ability to invite others to join.
* Groups shall have chat functionality for real-time communication.

9. Responsive Design

* The system shall be responsive and accessible on various devices, including desktops, tablets, and mobile phones.
* The user interface shall adapt to different screen sizes and resolutions.

10. Search Functionality

* The system shall include a search bar for users to search for specific information.
* Search results shall be displayed in a relevant and organized manner.
* The search feature shall support filtering and sorting options.

Non-Functional Requirements

Performance

* The system shall load pages and retrieve data within 3 seconds on average.
* Concurrent user handling: The system shall support at least 1000 concurrent users without performance degradation.

Usability

* The user interface shall follow best practices for readability and accessibility.
* Error messages shall be clear and concise to assist users in troubleshooting.
* Reliability
* The system shall have a minimum uptime of 99.9%.

Scalability

* The system architecture shall be designed to accommodate future expansion and additional features.
* Database scalability: The system shall handle a growing volume of data without significant performance issues.

Compatibility

* The system shall be compatible with the latest versions of popular web browsers (Chrome, Firefox, Safari, Edge).
* Mobile compatibility: The system shall provide a mobile-friendly experience on smartphones and tablets.

Security

* Role-based access control: Users shall have roles (student, teacher, administrator) with specific access permissions.
* Cross-Site Scripting (XSS) prevention: The system shall implement measures to prevent XSS attacks.
* Data encryption: All sensitive data, such as passwords and personal information, shall be encrypted at rest and in transit.

Maintenance

* The system shall be easy to maintain and update with clear documentation.
* Code quality: The codebase shall follow best practices and coding standards for maintainability.

Legal and Regulatory

* Compliance with data protection laws (e.g., GDPR) shall be ensured.
* Copyright compliance: All downloadable materials shall be properly attributed to their respective owners.

3.2 Design Constraints

Technology Stack

* The project must adhere to the chosen technology stack: HTML, CSS, JavaScript (JS), Angular, and Bootstrap.

Browser Compatibility

* The system must be compatible with the latest versions of major web browsers: Chrome, Firefox, Safari, and Edge.

External Libraries

* Angular libraries are required for it and bootstrap.

Responsive Design

* The user interface must be responsive and adaptable to various screen sizes, including desktops, tablets, and mobile phones.

4. Analysis Models

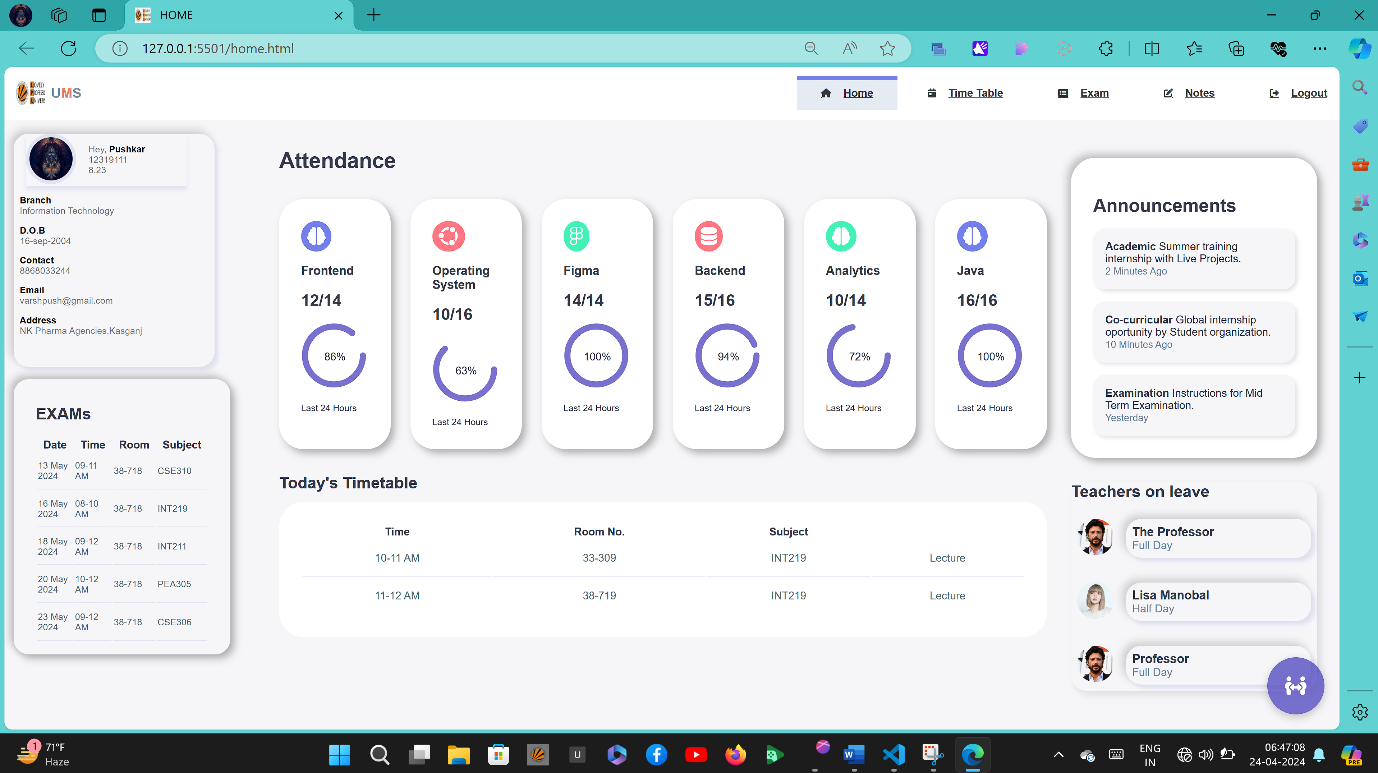
4.1 Screenshots

Fig1: Home page

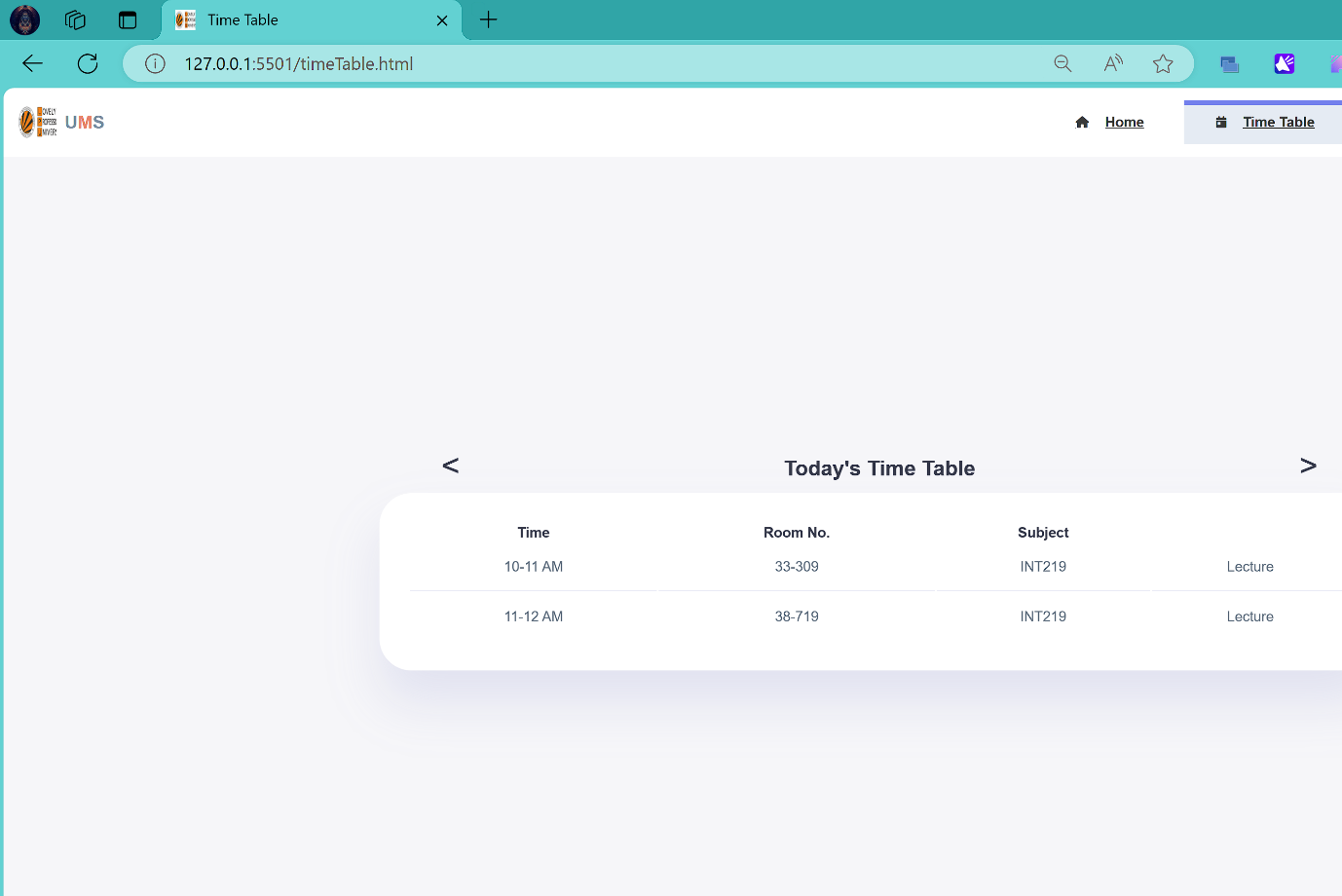
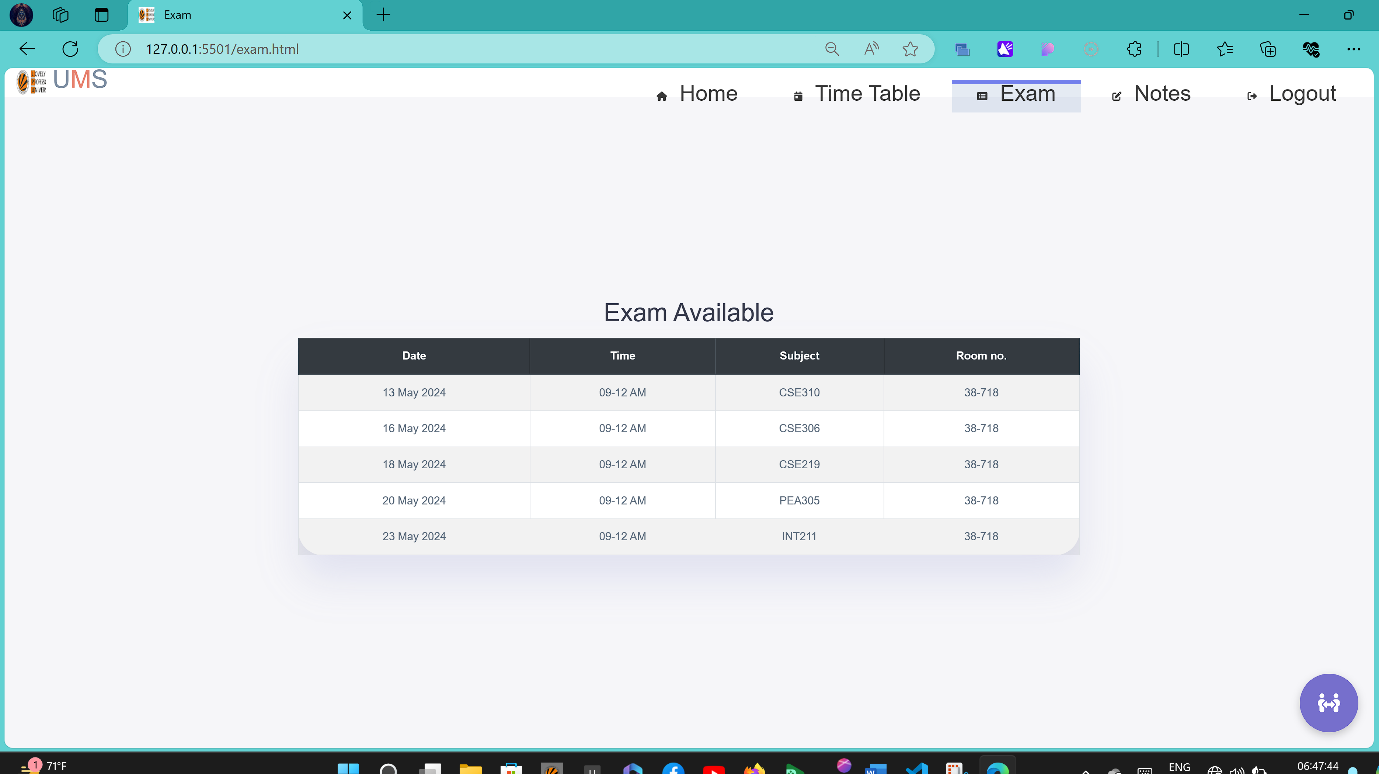
 

Fig.2: Time table page

Fig.3:Exam page(bootstrap only)

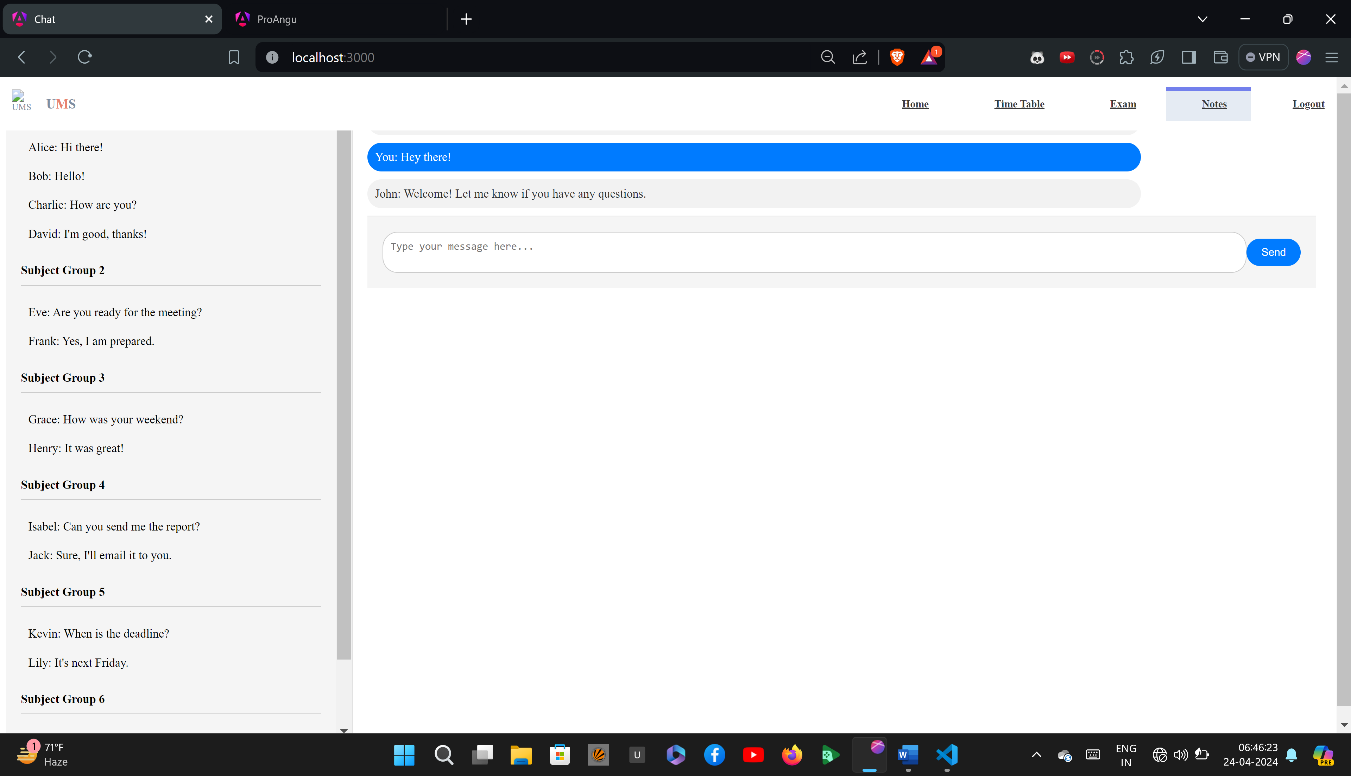
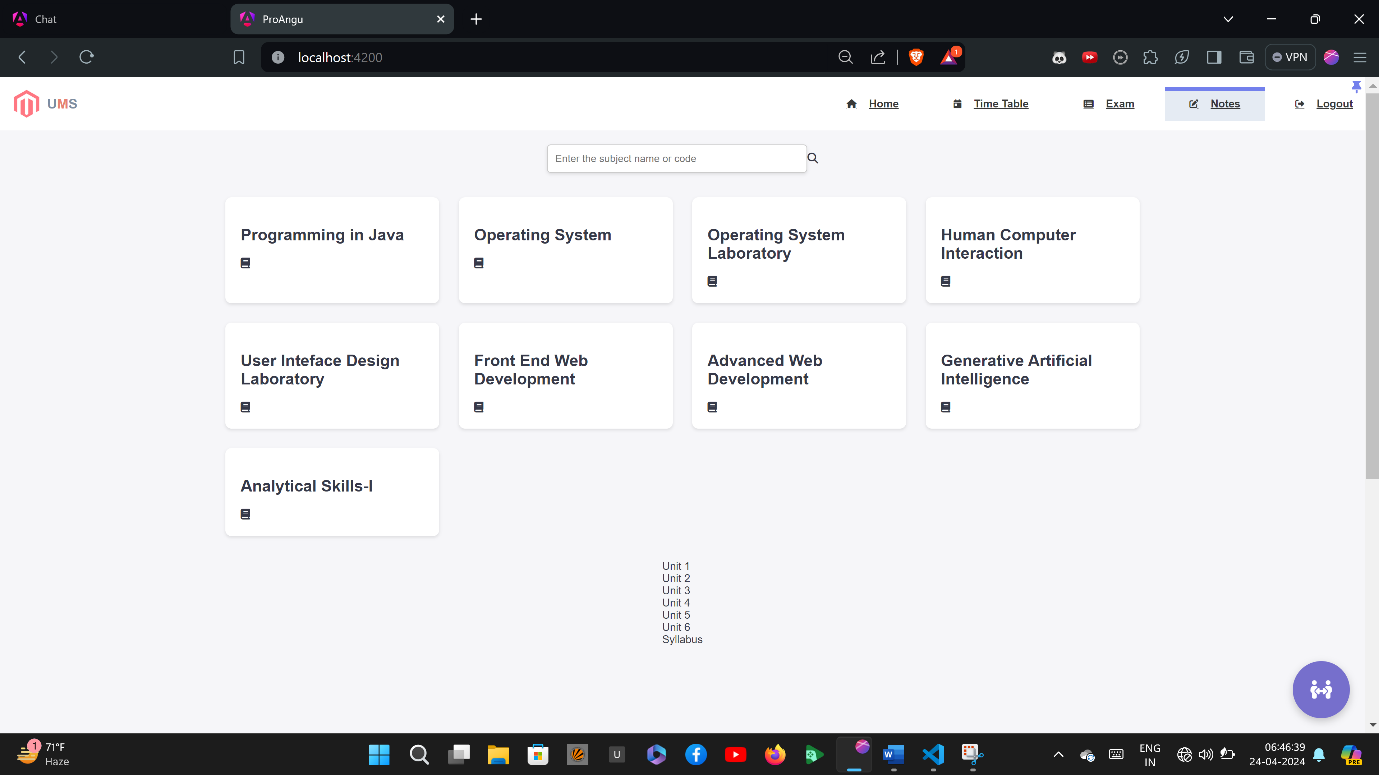


Fig4:Community page

Fig.5: Notes downloading page

4.2 Data Flow Diagrams (DFD)

Introduction:

Data Flow Diagrams (DFDs) are graphical representations of the flow of data within a system. They illustrate how data moves through various processes and interactions within the system. In the context of the Student College Account Management System, DFDs are used to visualize the flow of information related to attendance tracking, exam schedules, announcements, timetables, and other functionalities.

In my project there is no backend till now in it but I can tell how DFD looks like if I added in it.

Narrative Description:

Level 0 DFD (Context Diagram)

This high-level diagram shows the system as a single process interacting with external entities such as users (students, teachers, administrators), external databases, and the college infrastructure.

It provides an overview of the entire system and its interactions with external entities, without going into detail about internal processes.

Level 1 DFDs

Attendance Tracking (DFD)

This DFD details the flow of data related to attendance tracking.

Processes include viewing attendance records, updating attendance, and generating attendance reports.

External entities are students and teachers interacting with the system.

Exam Schedule Management (DFD)

This DFD illustrates the flow of data for managing exam schedules.

Processes include uploading exam schedules, viewing schedules, and sending notifications.

External entities are students and administrators.

Announcements Display (DFD)

This DFD outlines the flow of data for displaying college announcements.

Processes include posting announcements, viewing announcements, and dismissing read announcements.

External entities are all users of the system.

Timetable Creation (DFD)

This DFD shows the flow of data for creating and managing personal timetables.

Processes include selecting courses, generating timetables, and printing or downloading timetables.

External entities are students.

Teacher Leave Updates (DFD)

This DFD illustrates the flow of data for updating teacher leave status.

Processes include updating leave, notifying students, and displaying leave information.

External entities are teachers and administrators.

Notes Downloading (DFD)

This DFD details the flow of data for accessing downloadable course materials.

Processes include uploading notes, downloading notes, and organizing materials.

External entities are students and teachers.

Community Group Feature (DFD)

This DFD outlines the flow of data for managing community groups and discussions.

Processes include creating groups, joining groups, posting messages, and receiving notifications.

External entities are users participating in group activities.

User Authentication (DFD)

This DFD shows the flow of data for user authentication and account management.

Processes include logging in, resetting passwords, and managing user profiles.

External entities are all users interacting with the system.

Traceability to SRS Requirements:

Each DFD model is directly linked to specific requirements outlined in the SRS.

For example, the DFD for "Attendance Tracking" is directly linked to the specific requirements related to attendance tracking functionality.

These DFDs provide a visual representation of how data flows through the system to fulfill the defined requirements.

1. Github Link:

A. Appendices

Appendices may provide additional information related to the project. If present, the SRS will explicitly state whether the information in the appendices is considered part of the overall set of requirements.

A.1 Appendix 1

This appendix may include initial conceptual documents for the software project, such as early design sketches, wireframes, or prototypes.

These documents can provide context and background information for the development process.