Software Requirements Specification

for

TRACKit

Version 1.0

Prepared by

Group 5: **Group Name: Chill guys**

Aditya Gautam	**220064**	**agautam22@iitk.ac.in**
Aayush Singh	**220024**	**aayushs22@iitk.ac.in**
Dhruv Rai	**220365**	**dhruvrai22@iitk.ac.in**
Ved Prakash Vishwakarma	**221180**	**vedprakash22@iitk.ac.in**
Sharique Ahmad	**221002**	**asharique22@iitk.ac.in**
Dhruv Varshney	**220366**	**vdhruv22@iitk.ac.in**
Mayur Agrawal	**220641**	**mayurag22@iitk.ac.in**
Akash Verma	**220097**	**akashv22@iitk.ac.in**
Rahul Ahirwar	**220856**	**rahula22@iitk.ac.in**
Abhijeet Agarwal	**210025**	**abhijeeta21@iitk.ac.in**
Aryan Bansal	**200198**	**aryanb20@iitk.ac.in**

Course: CS253

Mentor TA: Jeswaanth Gogula

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
v1.00	Dhruv Rai Aditya Gautam Aayush Singh Ved Prakash Sharique Ahmad Dhruv Varshney Mayur Agarwal Akash Verma Rahul Ahirwar Abhijeet Agarwal Aryan Bansal	The First and Final version of the requirement document before submission.	24/01/25

1 Introduction

1.1 Product Scope

Academics lie at the heart of our college IITK, serving as the foundation for learning, growth, and innovation. However, managing a large number of courses, each with its own set of lectures, assignments, quizzes, and exams, can become overwhelming for both instructors and students. Often different instructors use different platforms for their courses, leading to duplication of the platforms that students have to keep up with. The challenge of staying organized and ensuring nothing falls through the cracks often leads to inefficiencies, missed deadlines, and undue stress. This is where TRACKit steps in, offering a modern, streamlined solution to tackle these complexities effectively. Our product will offer a centralized platform for all faculty and students.

TRACKit is a web application designed to provide a comprehensive platform that caters to the diverse needs of academic management. By integrating features such as scheduling, task tracking, and progress monitoring, TRACKit enables students to stay on top of their coursework while allowing instructors to manage their courses with ease. The platform centralizes information about lectures, assignments, quizzes, and exams, making it accessible in one convenient location. With intuitive navigation and real-time updates, users can track deadlines, access assignments, and view schedules seamlessly, fostering a more organized and productive academic experience.

Beyond its functional benefits, TRACKit promotes collaboration and transparency, enhancing the overall educational ecosystem. It allows instructors to share resources, communicate updates, and provide timely feedback, ensuring students remain informed and engaged. For students, the application reduces the stress of juggling multiple responsibilities by offering reminders, priority settings, and personalized dashboards. By addressing the pain points of academic management, TRACKit empowers both students and instructors to focus on what truly matters – learning and teaching – while minimizing administrative burdens.

1.2 Intended Audience and Document Overview

This software requirement documentation is written to be used and understood by:

- Software developers, as members of the team, utilize this document as a guide throughout the development process to ensure that the software being created meets all necessary requirements. It further aids in identifying an appropriate architectural pattern and planning the development workflow accordingly.
- Product Managers, who are responsible for overseeing the planning, development, and execution of the software project.
- Testers conduct product and quality checks, undertaking rigorous testing to assess the software's performance. They provide valuable feedback on the user interface, potential areas for improvement, and overall quality.
- Users can refer to this document to verify and explore the requirements that our product is designed to fulfil.

Section 1

This section delineates the product scope and provides essential background information, including document conventions and abbreviations, to facilitate a better understanding of the content. While seasoned readers may choose to skip this section, it serves as a helpful reference to clarify any potential ambiguities within the document.

Section 2

This section offers a concise yet comprehensive overview of the software, outlining its core functionalities, key assumptions, and dependencies. Readers are encouraged to engage with this part, as it lays a solid foundation for a deeper understanding in the subsequent sections.

Section 3

This section delivers detailed information about the software, explaining its functions in depth and utilizing various diagrams. This information is critical for both end-users and developers, serving as a guide for the development process for developers and as an instruction manual for end-users.

Section 4

This section provides insight into the non-functional requirements that the software must fulfil. It is of particular significance to developers, as it guides their implementation decisions to ensure the system meets all essential performance, usability, and other critical non-functional criteria.

1.3 Definitions, Acronyms and Abbreviations

- FCH - First Course Handout
- ID - Identity
- OTP - One-Time Password
- SRS - Software Requirement Specification
- TRACKit - Testing Reporting Academic Comprehensive Kit (website name)
- UI - User Interface

1.4 Document Conventions

While preparing this document, to make readability user-friendly and important parts clear, we have used the following conventions:

• The headings of all the sections are written in bold and underlined using Arial font size 14.
• The headings of subsections are bold and written in the same font as that of content using font size 13.
• The content of the section is written in Arial font size 11.
• Any important term is written in bold.
• The alignment of the whole content is justified.
• Text has been indented wherever required to highlight the hierarchy of the content.
• The document follows the IEEE formatting, indenting, and numbering conventions. Any deviations from the same will be explicitly specified.

1.5 References and Acknowledgments

References:

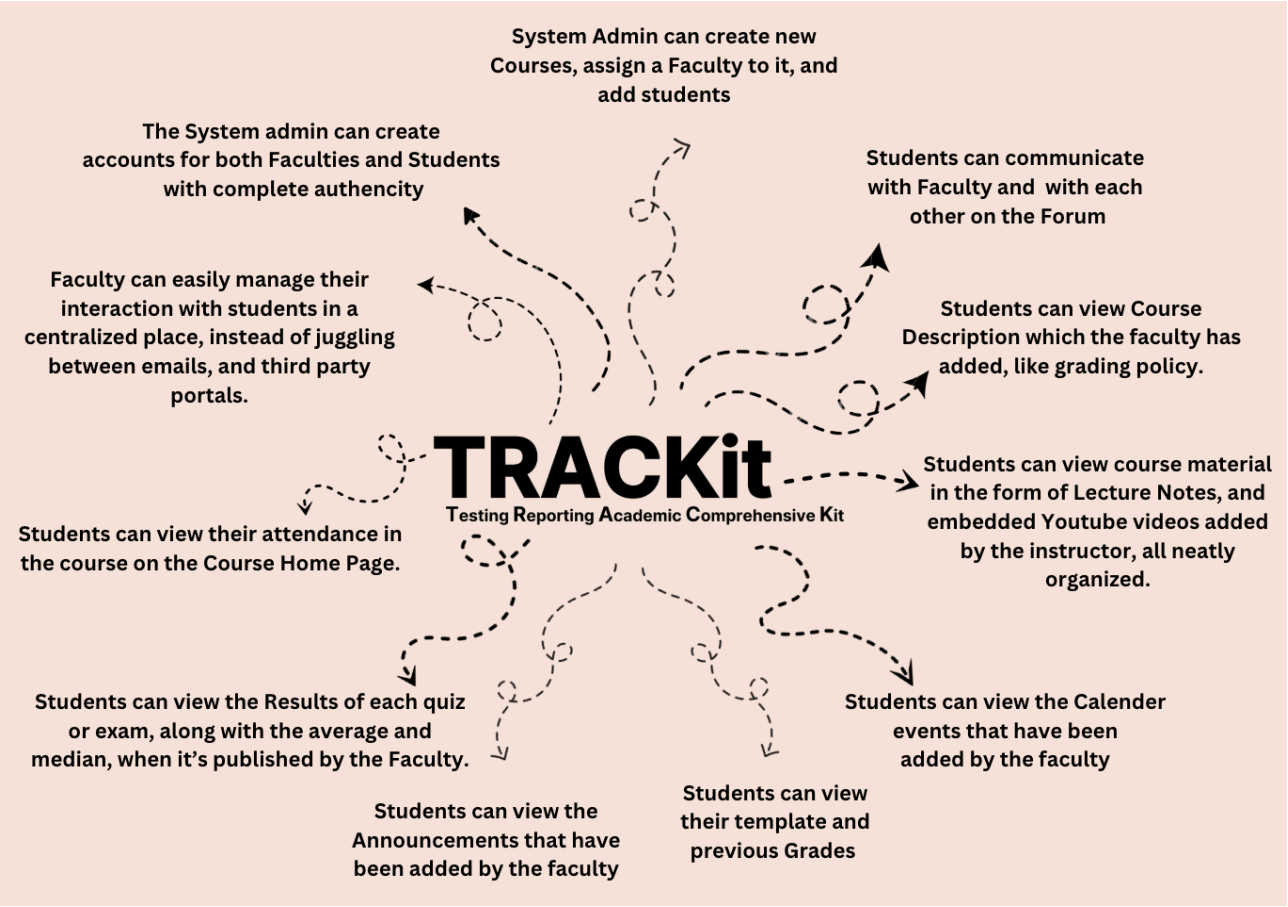
- Style Guide Template as given by Professor Indranil Saha.
- Professor Indranil Saha's CS253 Lectures for concepts and definitions.

Acknowledgements:

- The fundamental principles of our system design were established through the lecture notes and guidance provided by Professor **Dr Indranil Saha**, along with the assistance of our Teaching Assistant, **Jeswaanth Gogula.**

2 Overall Description

2.1 Product Overview



Our product, TRACKit, or the *Testing Reporting Academic Comprehensive Kit* – is a one-stop shop replacement for the myriad of academic management platforms and techniques already used by professors and Students. Some professors use competitors like HelloIITK, Piazza, Moodle, course-specific static websites, or simply just email the lectures and announcements to the students. Our product, TRACKit will not only replace these competitors, but it will also provide an excellent tailored solution for IIT Kanpur and even other institutions which currently don't use any digital academic management system.

Our Product is primarily being designed for IIT Kanpur, but it can be easily extendible to any academic institution around the world.

TRACKit will have three user types: the Faculty/Instructor, the student, and the System Admin. The roles of the Faculty and the student are as the name suggests, and the System admin is a special user who we have created for the sole purpose of ensuring authentic behaviour on our platform. Only System Admin will be able to create new accounts for Students and Faculty when they join the institute. After creating the accounts with the correct user information, the System Admin will mail the login info to the new user. This is to enforce authenticity and ensure that no fake accounts

are made, and common user errors in account creation (like accidentally filling in wrong data) are prevented.

After logging in the first page that the faculty and students will see will be the *Main Dashboard*. The main dashboard will have the student's current courses, and the merged schedule thereof, on the first section of the main dashboard, ie, *My Courses*. The Main Dashboard will also have a unique *Performance* section on its sidebar which will allow students to view their template and grades in completed courses at a glance. The *Profile* section would allow for actions like Password Change.

To view/create content, a student/faculty will select a Course from *My Courses,* respectively. TRACKit will have multiple features which will ease student's access to academic materials, announcements, resources, discussions and information about future events. On the faculty side, it will allow them to easily create these elements and push them to the students with the simple press of a button. It will have a Forum where students will be able to discuss aspects of the course with the Faculty and with each other. Thus both the Students and Faculty will have the following sections available to them on the sidebar after selecting a Course: *Course Info*, *Lectures*, *Announcements*, *Calendar*, *Results* and *Forum*.

Thus, our product will be a web-based application with an efficient Front-End and a robust Back-End.

2.2 Product Functionality

- **Login Page:** Users can log in using their username and password provided by the system administrator. Additionally, a "Forgot Password" option enables users to reset their password through an authentication process if they forget it.
- **Main Dashboard:** After logging in, users are directed to the dashboard, which displays their enrolled courses and a combined calendar featuring events and schedules for all courses.
- **Performance Tab:** Accessible from the sidebar of the main dashboard, this section provides users with their course progress template, highlighting completed courses, remaining courses, courses lagging behind the ideal template, and their full grade sheet.
- **Profile and Contact Us Tabs:** Also accessible from the sidebar of the main dashboard, the **Profile** tab allows users to view their personal details, while the **Contact Us** tab enables users to reach out to the support team for assistance in case of any issues.
- **Course Home:** Clicking on a course icon takes the user to the course home page, where they can view the event list, attendance, and materials (such as FCH and references) provided by the instructor. Only the instructor has the ability to add or edit materials and can also access overall class attendance statistics.
- **Course Sidebar:** Inside a course, the sidebar features six tabs—Course Home, Lectures, Announcements, Calendar, Results, and Forum—allowing users to navigate swiftly through different sections of the website.
- **Lectures:** This page displays all lectures posted by the instructor in chronological order, organized into sections and subsections. Each lecture can include videos, PDFs, and text, and only the instructor can upload or edit them.
- **Announcements:** This page provides users access to all announcements in chronological order. Instructors have exclusive access to add or edit announcements.
- **Calendar:** This page displays a detailed calendar that highlights various events specific to the selected course. Users can customize the calendar's appearance by choosing different

date ranges, such as day, week, or month views. Only instructors have access to add or edit events on the calendar.
- **Results:** This page allows users to view the results of various exams taken so far. Instructors have the ability to add, edit, and publish the results, and can also access overall class statistics.
- **Forum:** This page enables users to ask queries and respond to previously posted questions, primarily supporting text-based discussions. Instructors will have special abilities to moderate the forum, ensuring proper engagement and management.
- **System Admin:** This page allows the system administrator to create new courses, faculty, and user accounts, as well as manage them effectively.

2.3 Design and Implementation Constraints

- A user without administrator access should not be able to access the data available to the admins.
- Calendar updates and new information synchronization must occur in real time or within 1-2 minutes of an update.
- Resource-intensive features, such as file uploads or real-time notifications, must not exceed server capacity.
- Synchronization with Google Calendar for managing deadlines and events.
- The system must support parallel operations such as a student viewing performance data while receiving announcements or calendar updates.
- Instructors should be able to update results while students access the course portal.

2.4 Assumptions and Dependencies

- Third-party tools and APIs (e.g.calendar integration) will remain available and functional.
- End users (students/instructors) will access the platform on devices that support modern browsers and sufficient performance capabilities.
- The institution will provide ongoing support for maintaining and updating the system after implementation.
- Dependencies on open-source libraries and frameworks (e.g., React.js, Django) for efficient development.
- Dependence on institutional network stability to ensure consistent availability and performance.

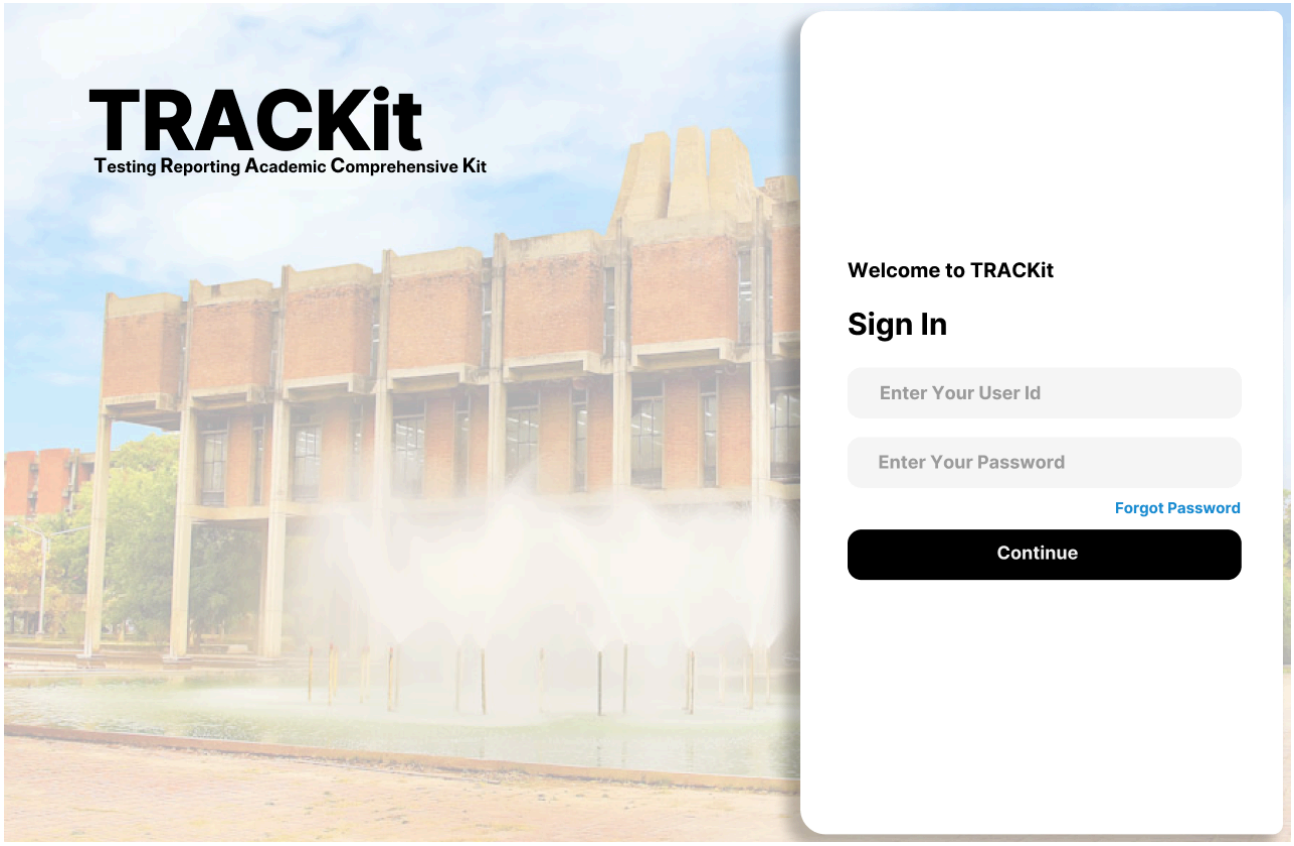
3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The Login Page

This is the landing page of our website. This will be the same for all users, who will use their credentials to log in to their accounts. It will have a separate Forgot Password button, which will allow users who have forgotten their Password to regain access to their accounts. An OTP will be created and sent to their emails, and they will be able to use this OTP to set a new password.

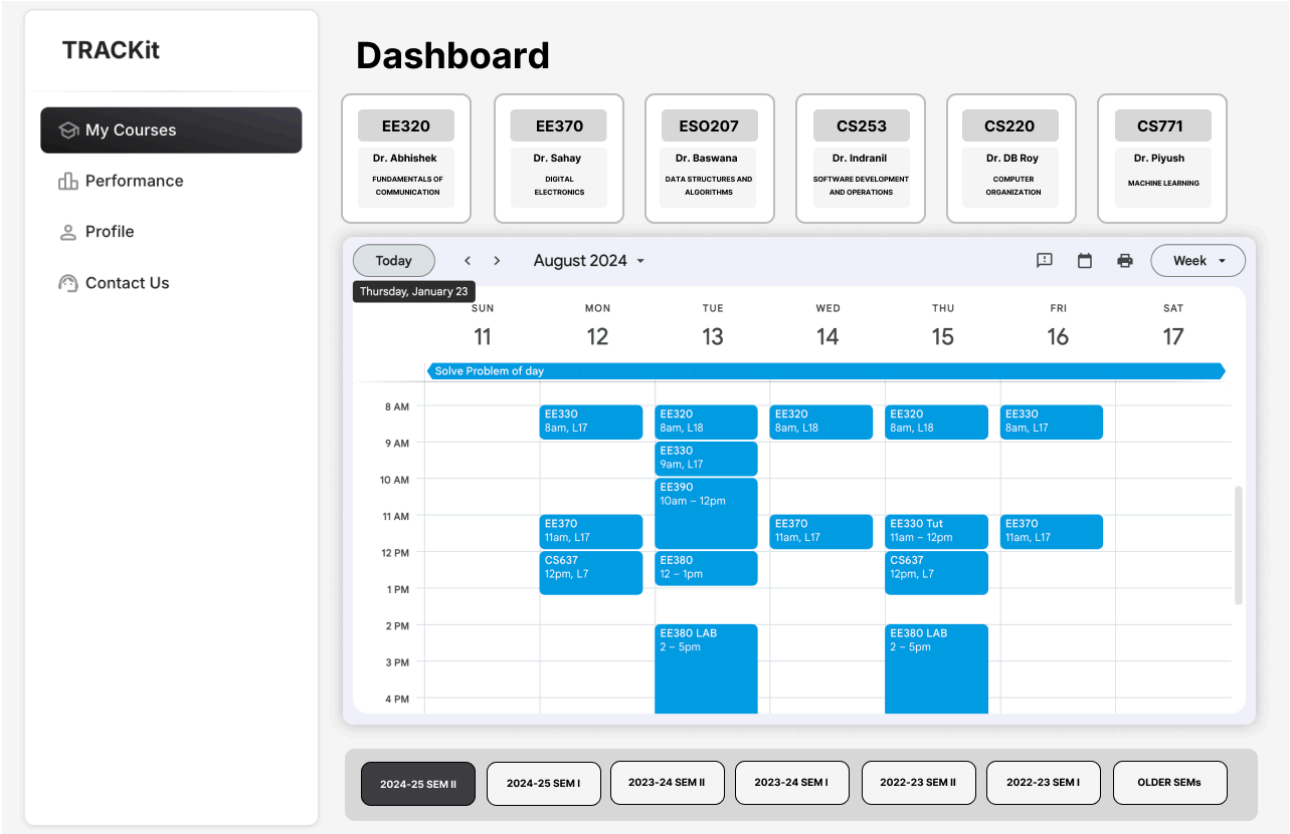


The Main Dashboard

The Main Dashboard will be the first screen both students and faculty will see upon signing into TRACKit. The Dashboard for the system admin will be characterised in an upcoming section.

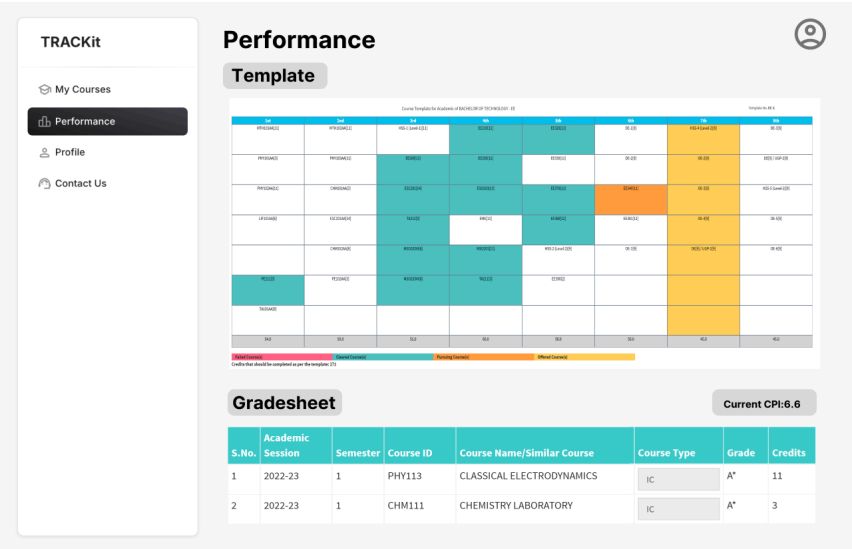
For Students, it will be a place to access the courses which they are doing in the current semester on the *My Courses* section. The *My Courses* section will also have a calendar displaying the merged data of events from all courses in which the student is enrolled. They can also access course pages of courses done by them in their previous semesters. There will also be a *Performance* section in the sidebar which will allow students to view their template and grades in completed courses at a glance. The *Profile* section on the sidebar would allow for actions like Change Password.

The UI for the Instructors would be exactly similar, except for the fact that they will not see the *Performance* section on their sidebar. Just like students can view the courses they are enrolled in for the current semester, the Faculty can view the courses which they are teaching in that semester. The calendar which they see will simply be a merged calendar of all the events they have created in the courses they are taking.



The Performance Page

As mentioned above, this page is viewed by clicking the namesake icon on the *Main Dashboard's* sidebar. This Page is available for students only. Students can see the template of their department as a chart. On this chart information like pending, failed, current & completed courses will also be displayed. The grade sheet of students will also be displayed here.

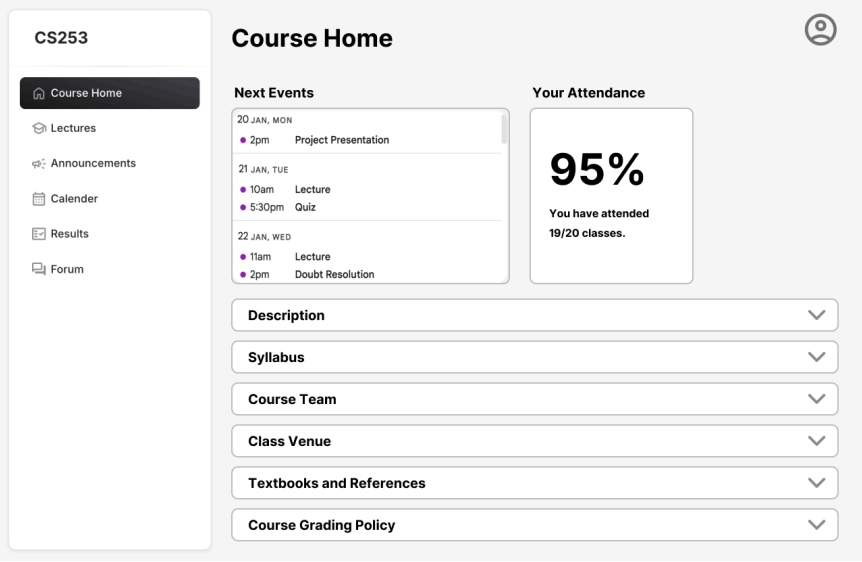


The Course Home Page

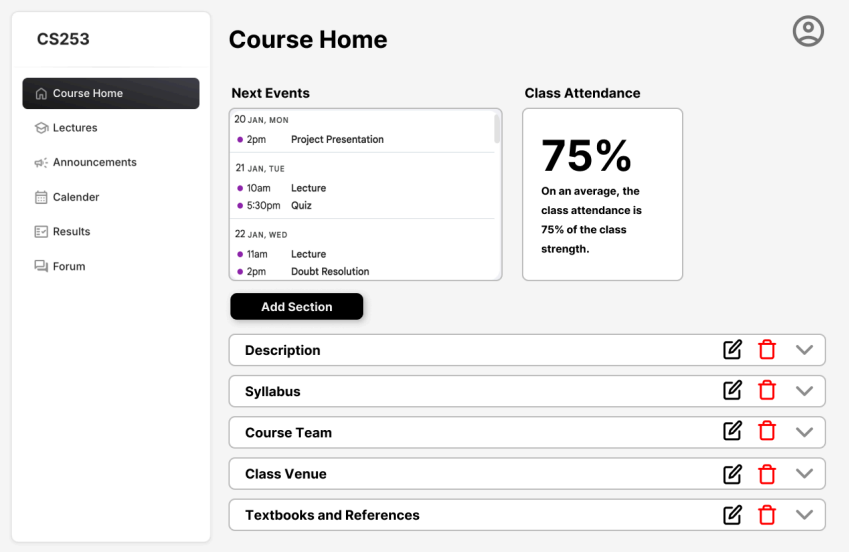
The *Course Home Page* is the first page which both students and faculty see after clicking the icon of a course on the *Main Dashboard*. The *Course Home Page* contains a small Events list in the top left corner which contains the upcoming events under that course scheduled by the course Instructor. Towards the immediate right of the Events list is the current attendance status box. To the students, it shows their current attendance in the course. To the Instructor, it shows the percentage of students attending lectures, on average.

Right below these are the Course Info sections. These collapsible boxes contain the most important facts and attributes about the course, similar to an FCH. The faculty may choose to add sections for Course Description, Grading Policy, Books and References, and anything else he/she thinks fit. He/She will also have the capability to edit existing sections.

Student's Course Home Page



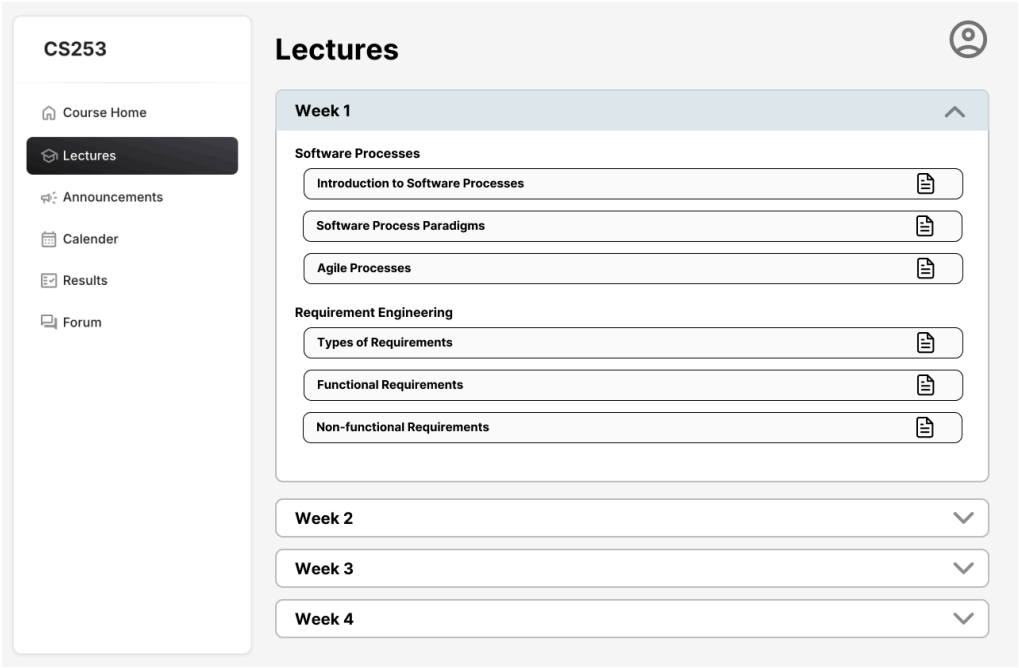
Instructor's Course Home Page



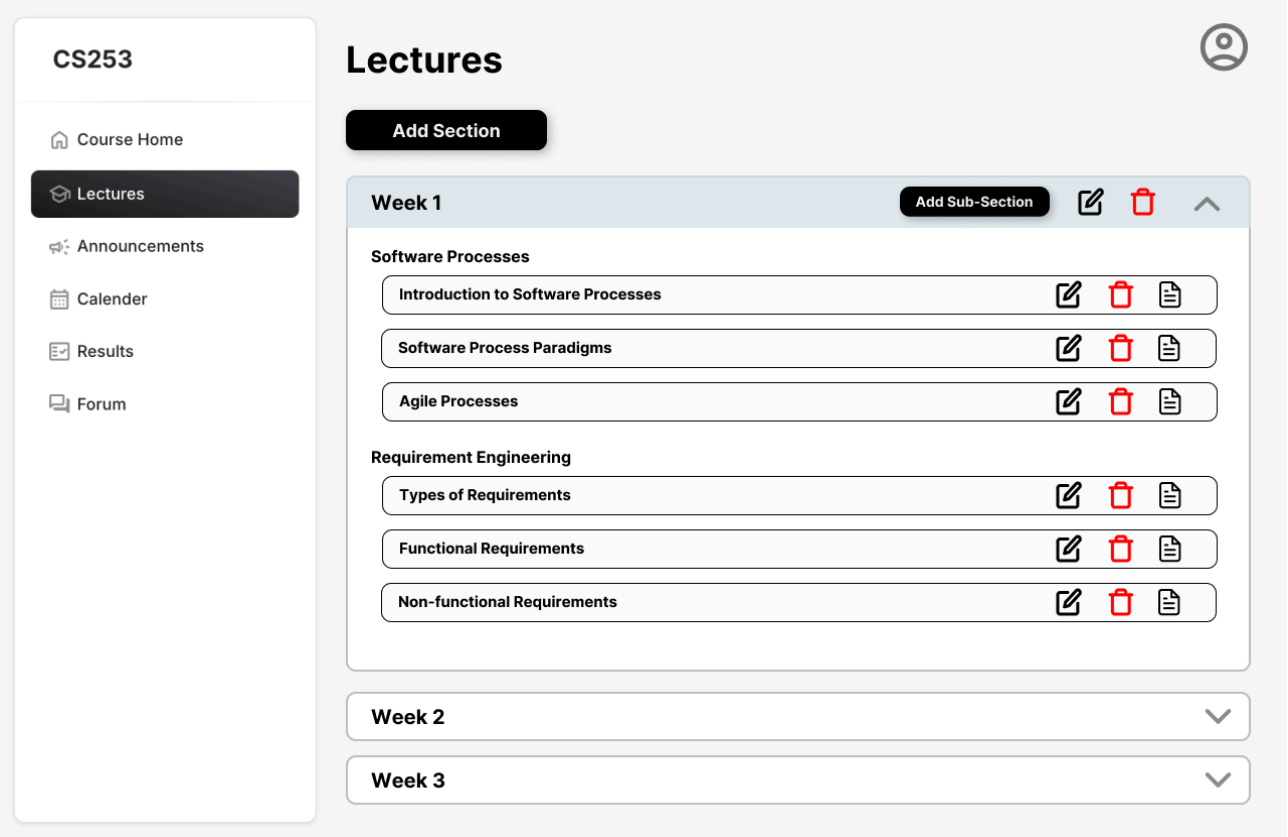
The Lectures Page

The Lectures Page will be a centralised location for course material. The Faculty will be able to create sections, ideally into weeks, and then further organise the material into sub-sections. Each subsection will be capable of holding a PDF, text, and an embedded YouTube video.

Student's Lecture Page



Faculty's Lecture Page



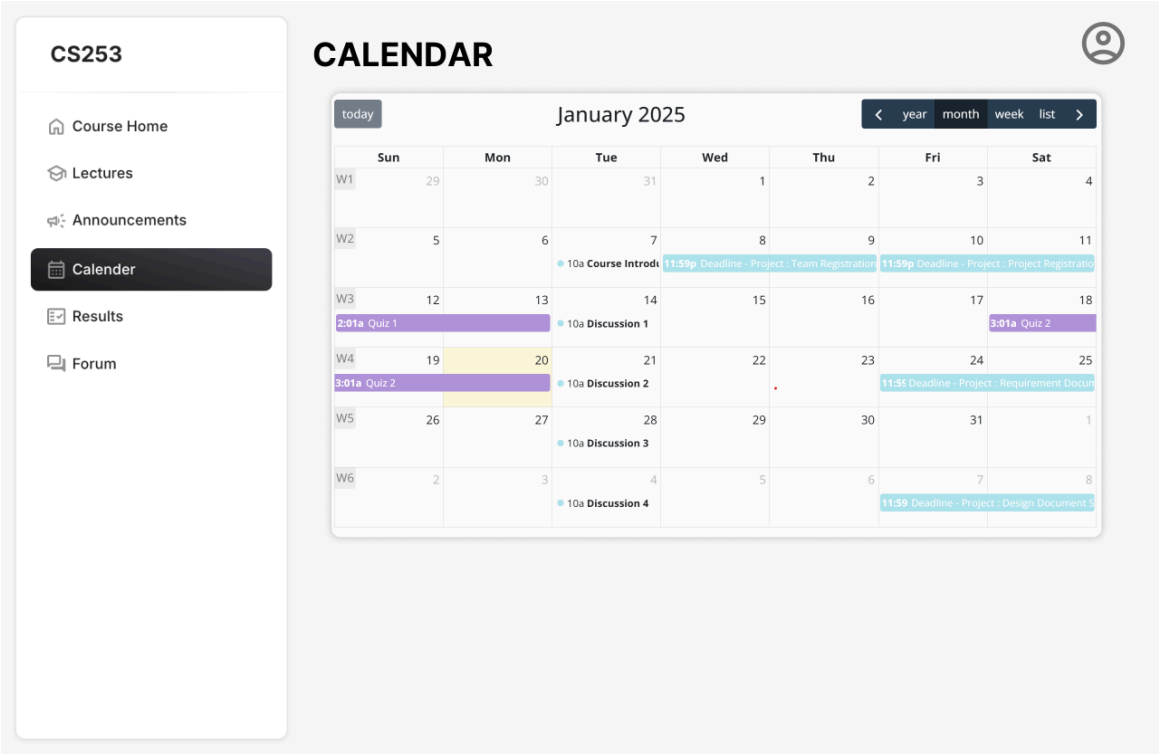
The Calendar Page

Each faculty member or student will have access to a *Calendar* page for every course they are teaching or enrolled in. This page can be accessed by selecting the desired course from the *Main Dashboard* and clicking on the *Calendar* option in the sidebar.

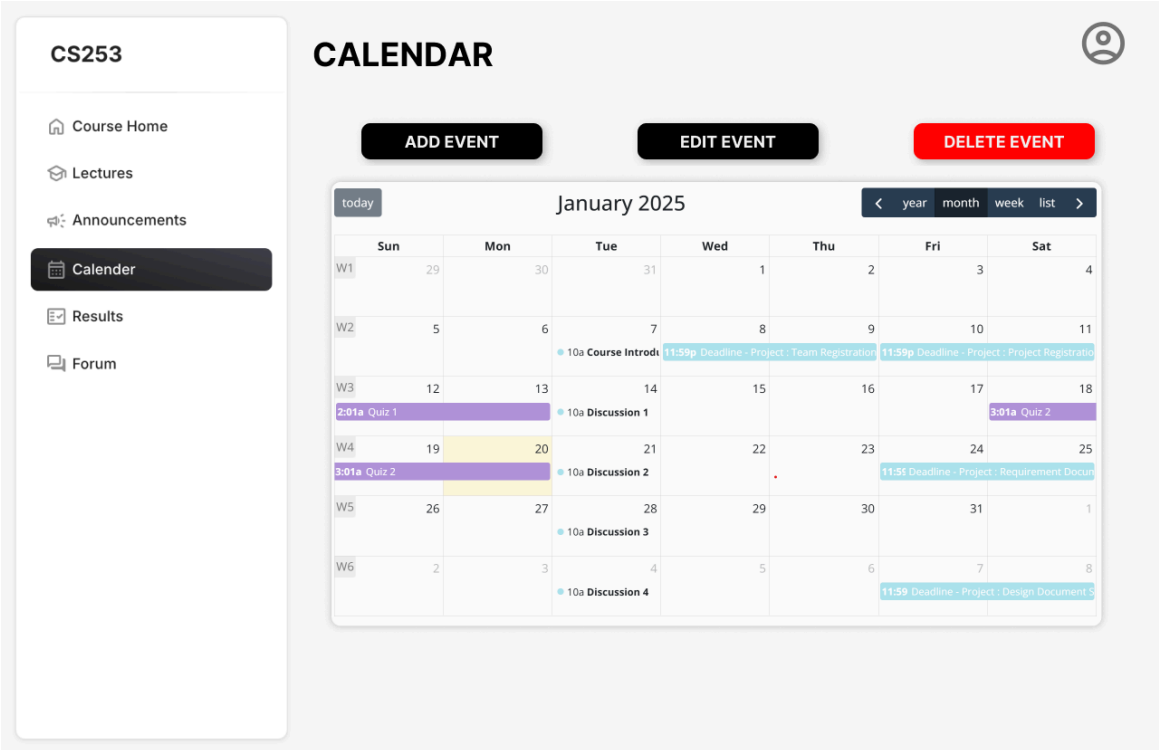
The *Calendar* provides a comprehensive overview of the scheduled activities for the course, including lectures, labs, tutorials, quizzes, discussion hours, deadlines for assignments, submissions, and other activities.

In addition to viewing events, the course instructor will also have features to: add, edit or delete events. Changes made by instructors are instantly reflected on the calendar, ensuring real-time updates. Additionally, instructors can choose to generate an announcement to notify students about the changes.

Students Calendar



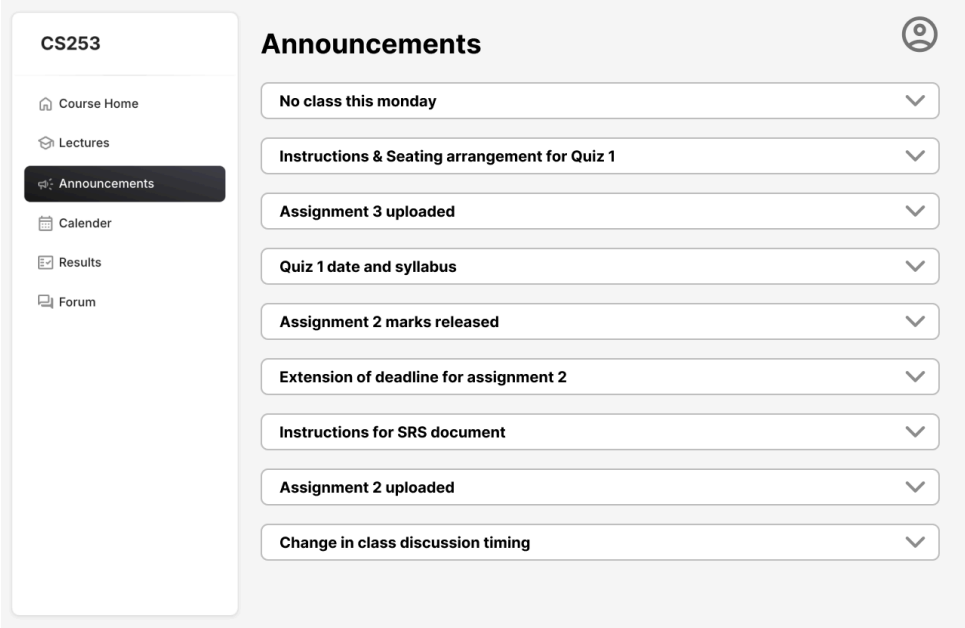
Instructor's Calendar



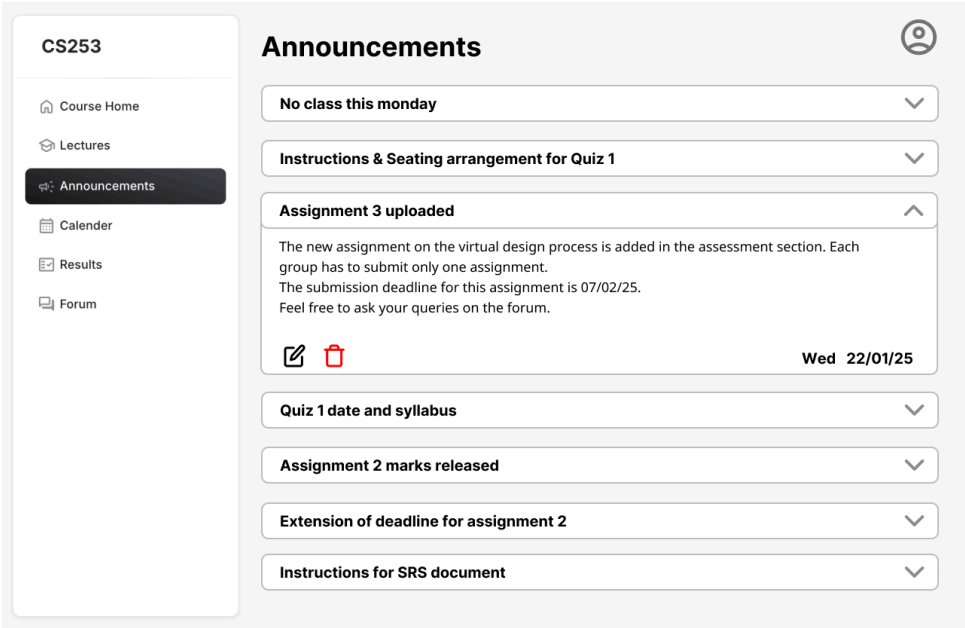
The Announcement Page

Instructors can use this page to post important announcements related to their courses. They can also edit or delete existing announcements. All students enrolled in said course will have access to the announcements made by the instructor. Additionally, the page will display the count of unseen announcements, corresponding to each student.

Student's Announcement Page



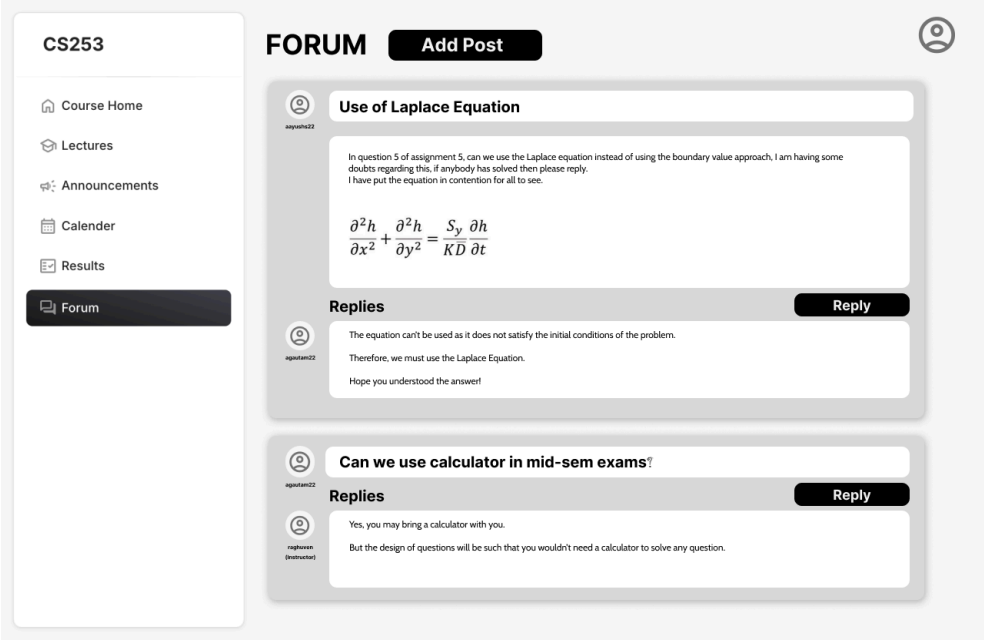
Instructor's Announcement Page



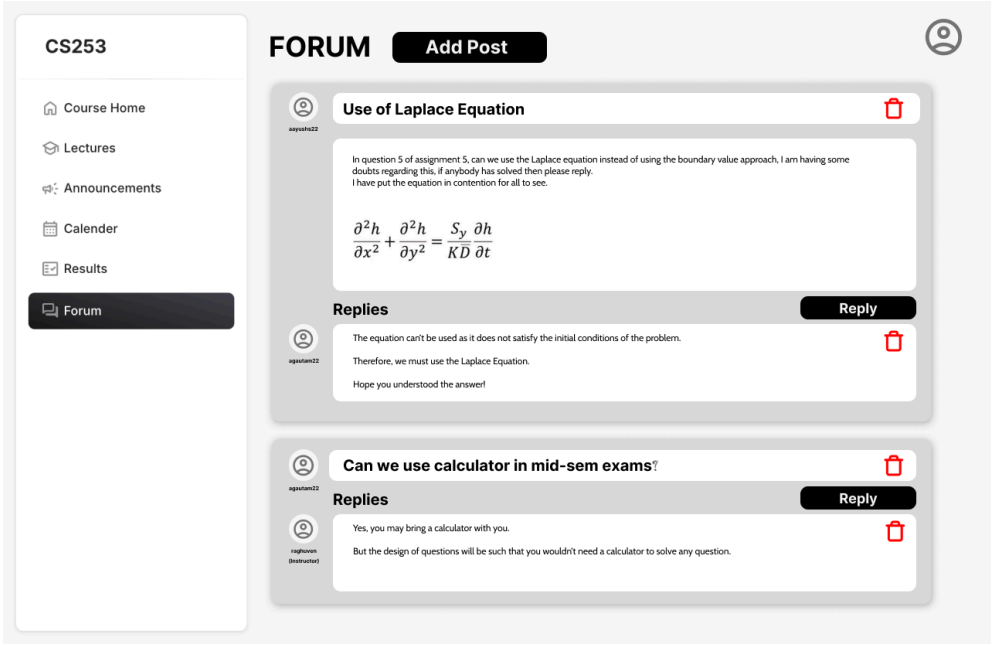
The Forum Page

The Forum Page will be accessible to both students and instructors enrolled in a particular course. Students can use this page to post their doubts, to which instructors can respond. Any user can reply to existing posts.

Student's Forum



Instructor's Forum



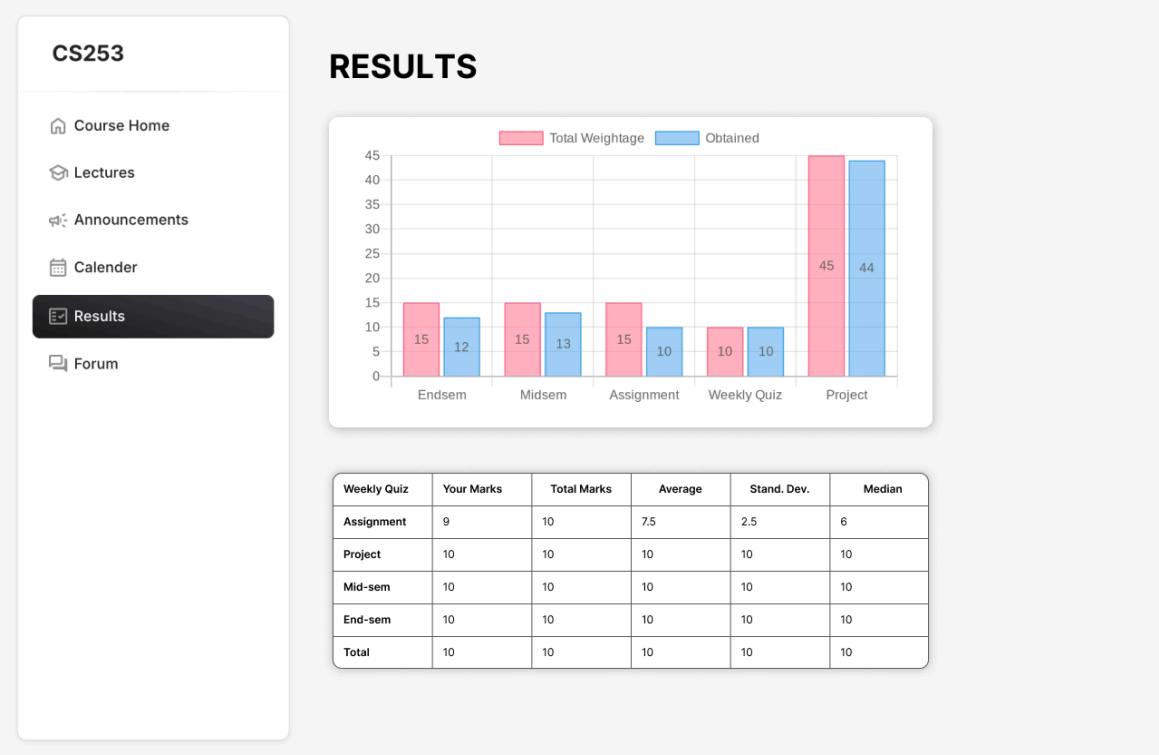
The Results Page

The Result Page will provide tailored functionality for both students and instructors:

- **For Students**: The Result Page will allow students to view their performance in a specific course. It will display their scores, total marks, average marks of all students, standard deviation, and median for each exam, quiz, and project in the course. Additionally, the page will show the weightage of each component and the corresponding weighted marks, offering a detailed breakdown of their performance.

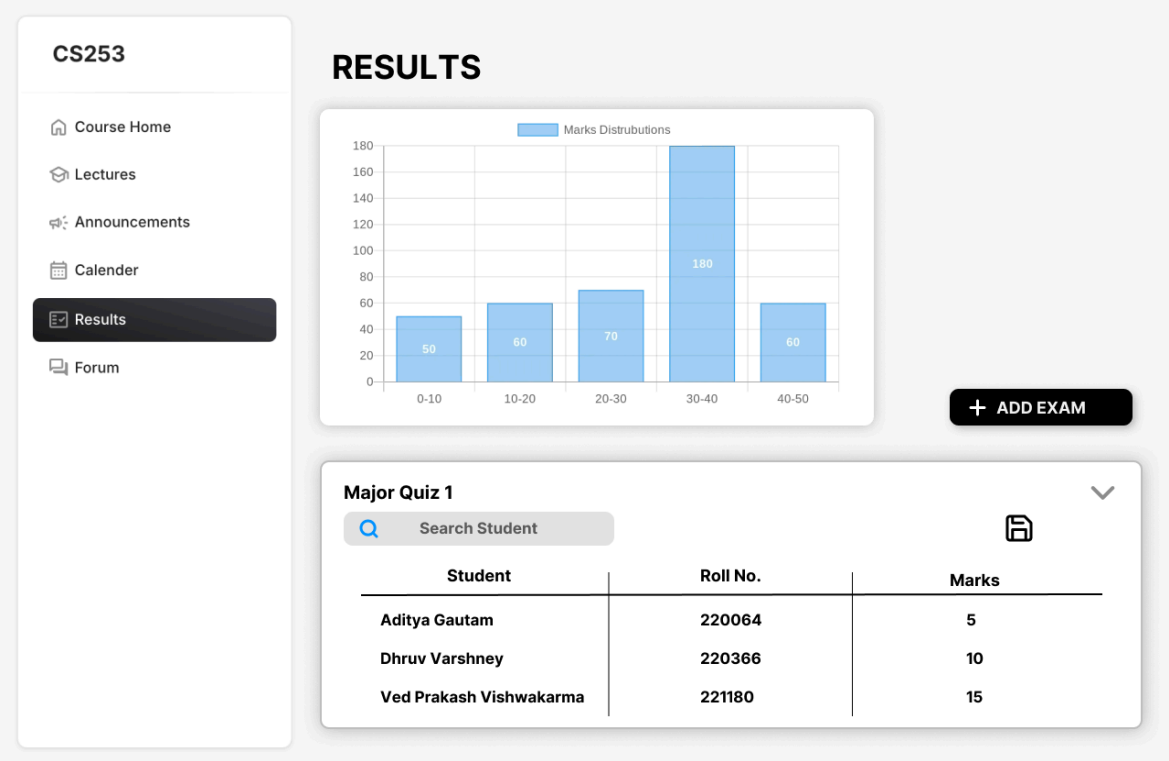
- **For Instructors**: The Result Page will enable instructors to analyze the performance of each student in the course. It will display the marks scored by individual students for quizzes, assignments, projects, and other components. Furthermore, the page will include a histogram representing the distribution of overall marks, assisting instructors in grading and evaluating students at the end of the course.

Student's Results Page



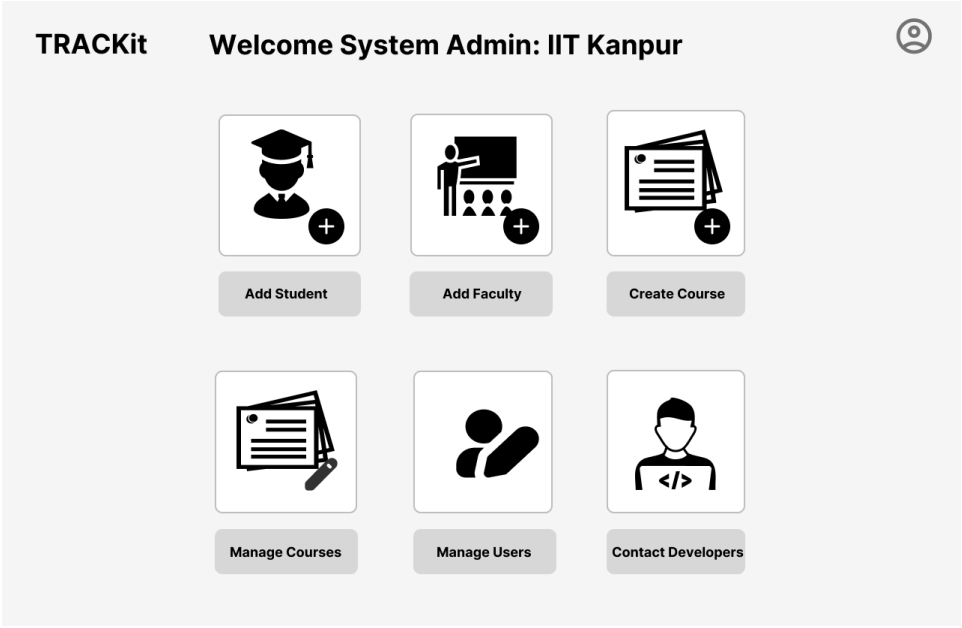
Weekly Quiz	Your Marks	Total Marks	Average	Stand. Dev.	Median
Assignment	9	10	7.5	2.5	6
Project	10	10	10	10	10
Mid-sem	10	10	10	10	10
End-sem	10	10	10	10	10
Total	10	10	10	10	10

Instructor's Results Page



The System Admin's Dashboard

The System Admin will be a unique user. This Dashboard will allow the system administrator to create new courses, faculty, and user accounts, as well as manage them effectively.



3.1.2 Hardware Interfaces

Clients must have internet-enabled devices such as desktops, laptops, or smartphones to access the platform seamlessly. On the server side, a robust web server will be required to host and manage the website efficiently.

3.1.3 Software Interfaces

- The client-side components of the software must be compatible with various modern web browsers, including Apple Safari 7+, Google Chrome 44+, Microsoft Edge 90+, and Mozilla Firefox 40+.
- To ensure user authenticity, the software shall verify credentials by cross-checking with the institution's database.
- The software will use multiple databases to store the following information:
    - User login credentials (for students, faculty, and admin)
    - Course enrollment data
    - Attendance data
    - Faculty data
    - Student data
    - Course material and resources
    - Assessment and grade data
    - Forum data
    - Announcement data etc.

3.2 Functional Requirements

3.2.1 User Management

1. The system shall allow the admin to create, edit, or delete courses by providing details such as the course name, description, and schedule.
2. The system shall allow the admin to add a new student or faculty member by providing their name, email, and student ID through a .csv file or individually.
3. Admins shall have the ability to add faculty or students to an existing course by uploading their details through a .csv file or entering them individually.
4. Users will be able to reset their passwords using 'Forget Password' or 'Profile tab'.

3.2.2 Course Management

1. Students will be able to view the current courses on the main dashboard.
2. Students can check their attendance for each course via the Course Home page.
3. Students will be able to view course policies like fch, grading, etc. on the Course Home Page.
4. Students will be able to access the study materials uploaded by the respective instructors on the Lecture page.
5. Students will be able to view their exam marks in both tabular and graphical formats on the Results page and instructor can add/edit the marks.

3.2.3 A Service for Interactive Forum Discussions

1. Students will be able to ask questions in the forum section. They can also reply to existing questions or share their insights in ongoing discussions through comments.
2. The forum section will allow professors to actively engage in discussions by responding to questions.

3.2.4 A Service for Course Announcements and Notifications

1. Respective Instructors can post updates or announcements related to the course schedule, assessments, or events.
2. Students will be able to access notifications in real time through the announcements tab.

3.2.5 A Service for Managing Activity through Calendar

1. Instructors can view, add, and manage course-related events, deadlines, and activities.
2. Students can view their upcoming academic events.
3. Users can switch between daily, weekly, or monthly views to plan their schedules effectively.

3.2.6 Performance tab for viewing the progress

1. The Performance page allows students to view their progress through a template that highlights completed, remaining, and delayed courses using distinct colors for easy tracking.
2. The page also displays the student's full grade sheet, which is timely updated by the system admin to reflect the most current information.

3.3 Use Case Models

3.3.1 Use Case #1 (Secure Account Creation and Login Password Management)

Author: Aditya Gautam and Rahul Ahirwar

Purpose: To authenticate and create accounts for valid users via system admin, and to illustrate options for users to change their password.

Requirements Traceability: Authenticated creation of user accounts on the software. Secure ways for users to change their password.

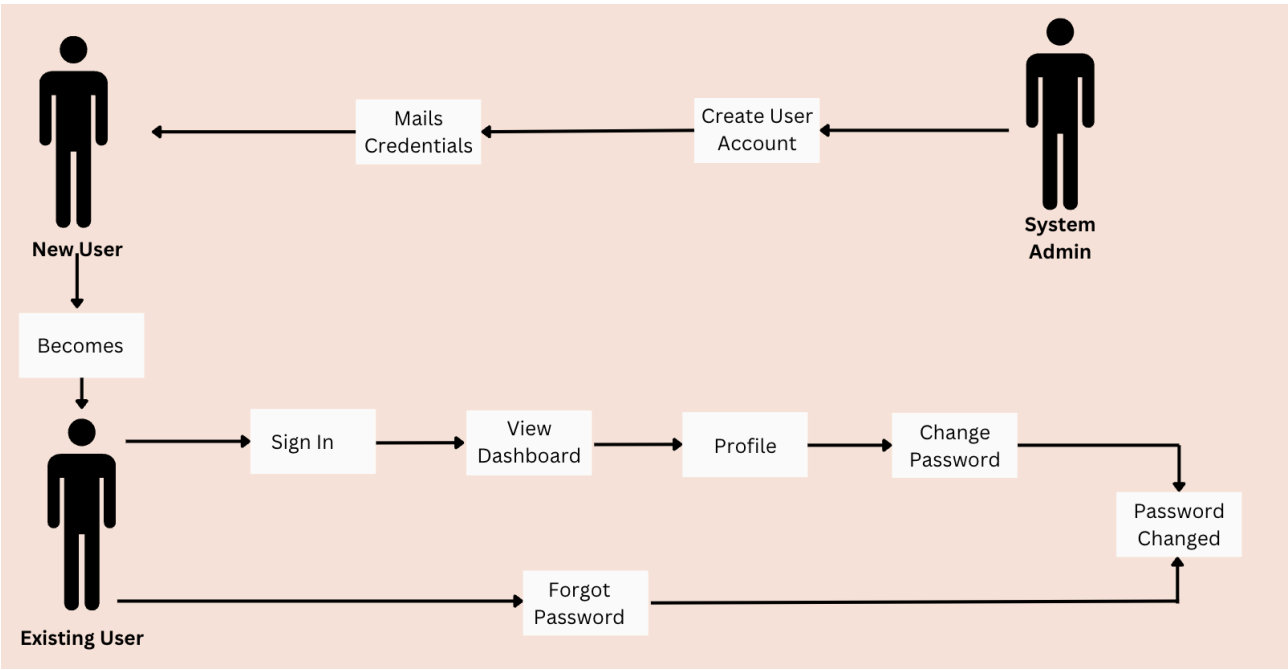
Priority: Very High

Preconditions: The user is a real student/ faculty, and has a valid Email ID.

Postconditions: The user is logged in and is able to view the dashboard and use the software.

Actors: System Admin, Students, Faculty.

Exceptions: The entered User ID should be valid, and be recalled by User.



3.3.2 Use Case #2 (Forum)

Author: Dhruv Rai and Dhruv Varshney

Purpose: To enable users to ask questions and comment on various topics covered in the class.

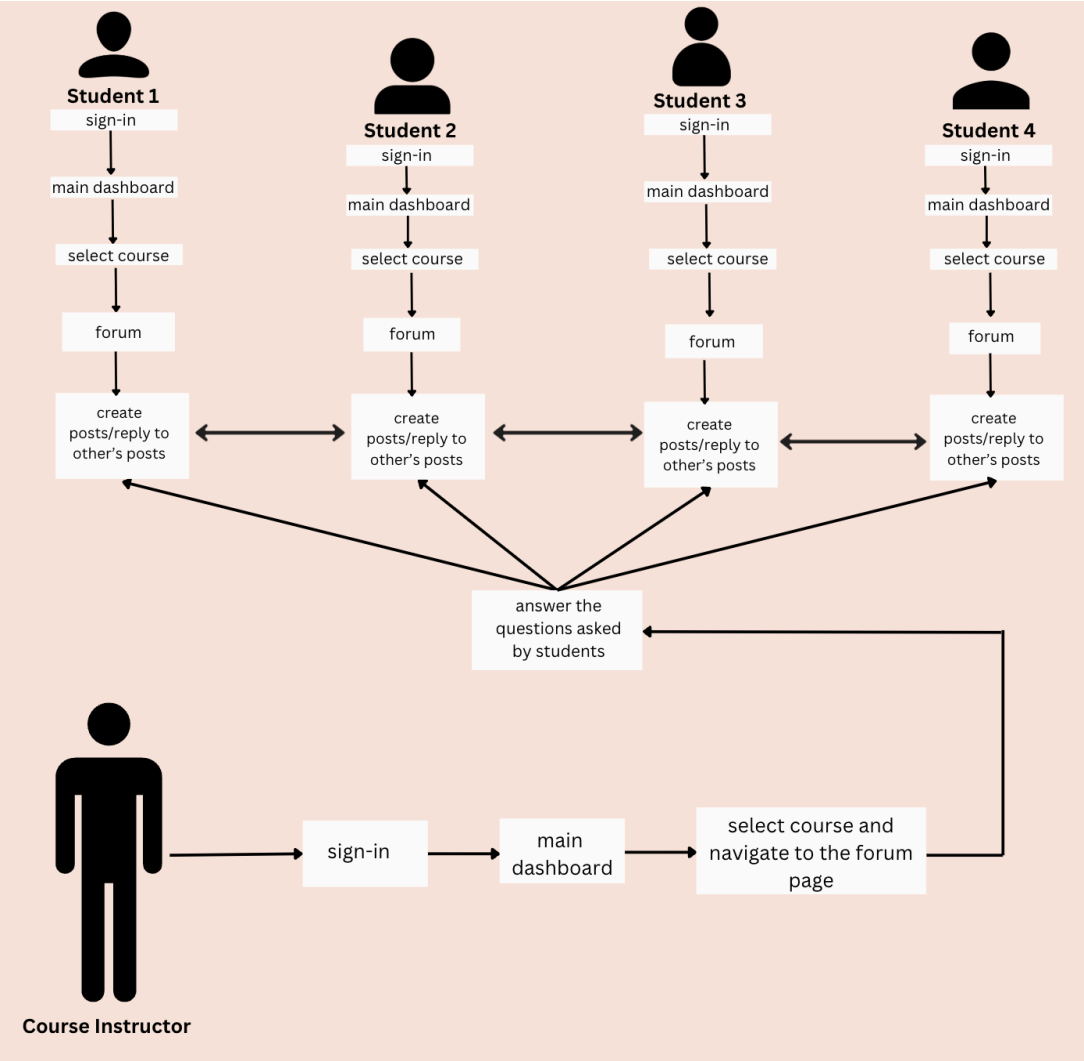
Requirements Traceability: This feature provides a collaborative platform for students to ask questions and discuss class topics, fostering interactive learning, peer engagement, and clarity. Faculty and registered students can create and comment on forum posts.

Priority: Medium

Pre-conditions: The user must have access to the specific course, either as an instructor of the course or a student enrolled in the course.

Post-conditions: After posting a new question, follow up or answer the user and all the other stakeholders will be able to see it. The number of unseen posts also increases for all the other users.

Actors: The primary actors in this use case are students and course instructors.



3.3.3 Use Case #3 (Calendar and Event List)

Author: Dhruv Rai and Sharique Ahmad

Purpose: To allow users to manage their day more effectively.

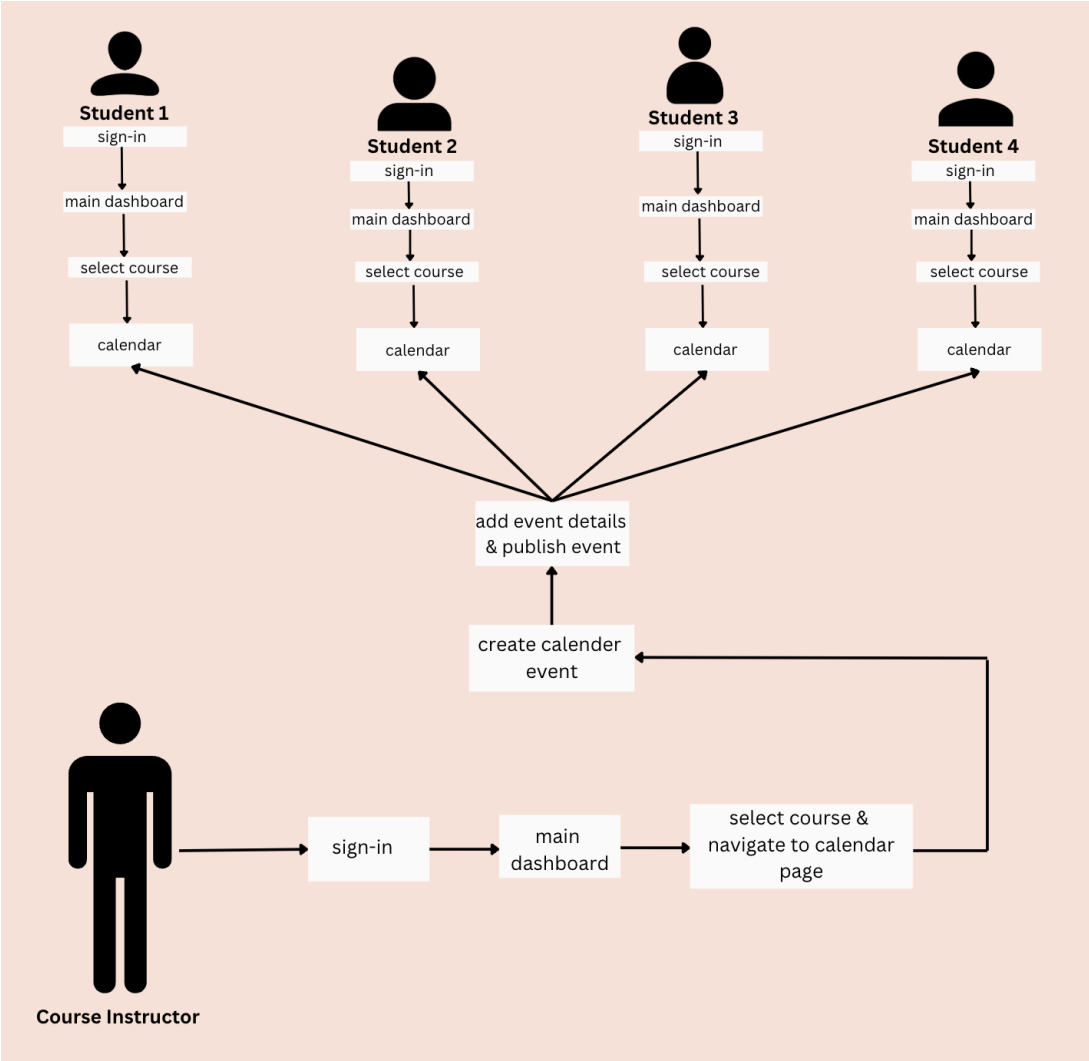
Requirements Traceability: This feature addresses the problem of poor management of daily tasks.

Priority: High

Pre-conditions: The user must have access to the system and be logged in.

Post-conditions: The user will be able to manage their daily tasks successfully.

Actors: The primary actors in this use case are students, who act as the users of the system. The events are created by the faculty.



3.3.4 Use Case #4 (Course Home)

Author: Rahul Ahirwar and Aayush Singh

Purpose: To enable instructors to upload, update, and manage course-related information, such as grading schemes, syllabus, and other critical components, ensuring effective communication and accessibility for students.

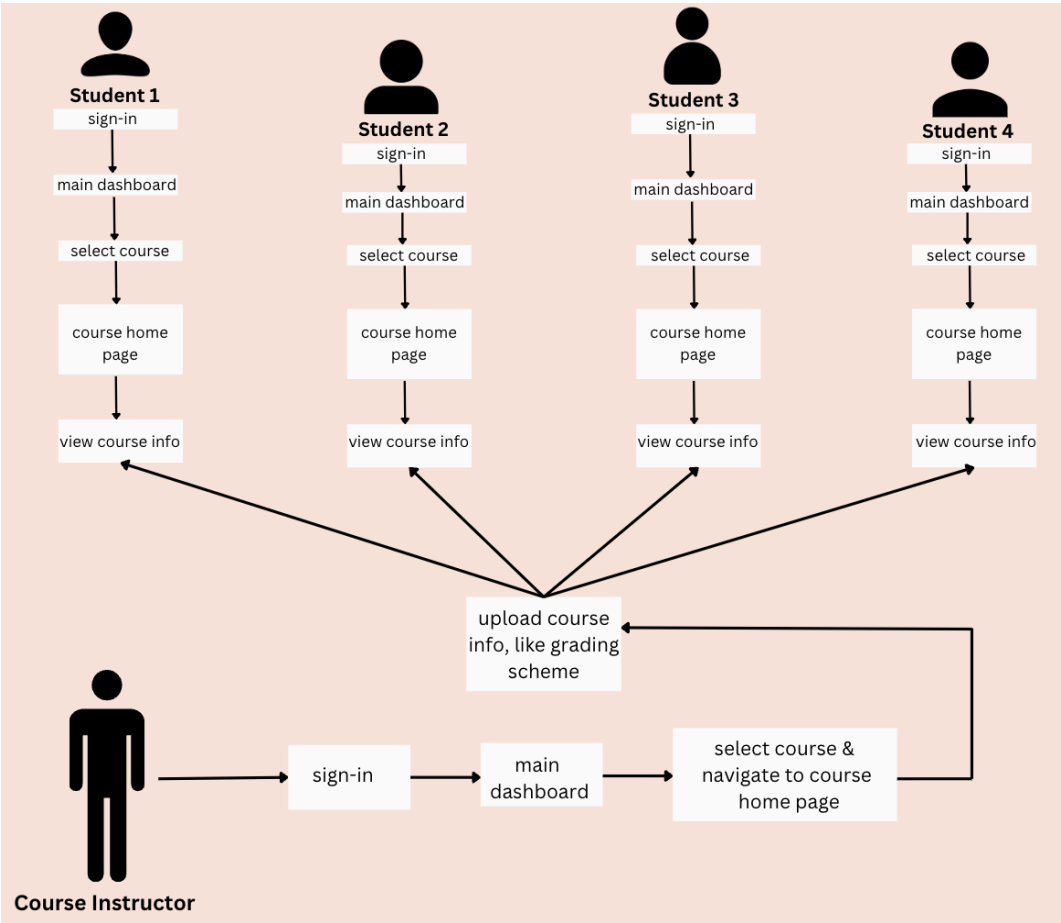
Requirements Traceability: This use case ensures instructors can share essential course details with students in a structured and accessible format, reducing manual effort and ensuring clarity for students regarding their coursework and performance metrics.

Priority: High

Pre-conditions: The instructor must have administrative access to the specific course in the system. The course must be created and active on the platform. The user must be enrolled in that course.

Post-conditions: Course information, including grading schemes, is successfully uploaded and visible to enrolled students.

Actors: Instructor and Students



3.3.5 Use Case #5 (Attendance)

Author: Akash Verma and Dhruv Rai

Purpose: Enable users to constantly check and monitor their attendance in different courses they are enrolled in.

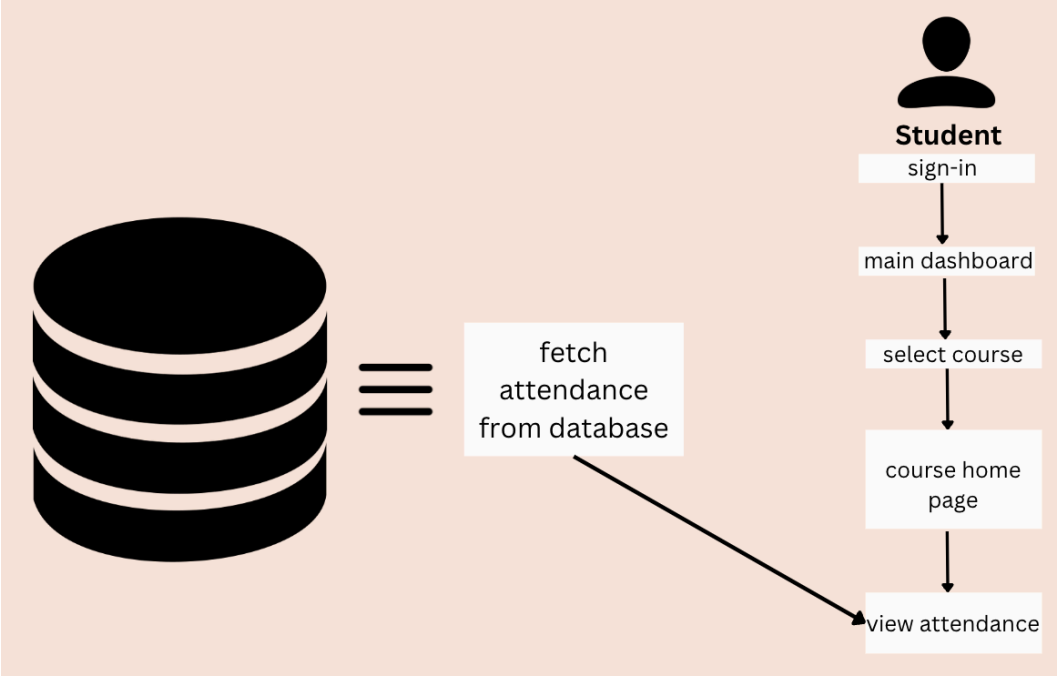
Requirements Traceability: This use case is required so that students can check their attendance without visiting any office. It also saves time for biometric staff and at the same time, instructors can keep track of student's attendance in their courses.

Priority: Medium

Pre-conditions: The user should be enrolled in that particular coursework. The biometric office must diligently maintain the attendance database.

Post-conditions: Users can view their current attendance in every course seamlessly.

Actors: Student and Instructor



3.3.6 Use Case #6 (Announcements)

Author: Abhijeet Agarwal and Mayur Agrawal

Purpose: To allow users to access announcements for a course.

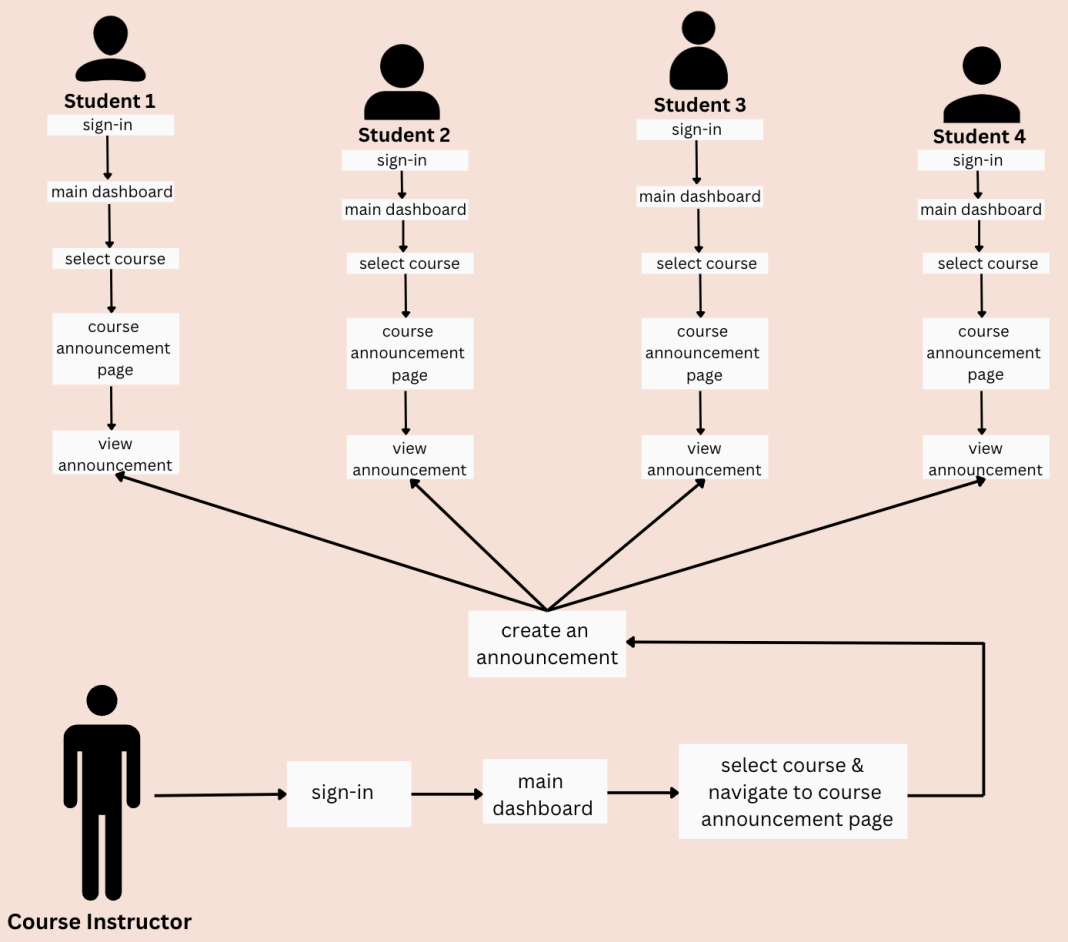
Requirements Traceability: This feature addresses the problem of missing major announcements for a course and helps users manage all important notifications easily.

Priority: High

Pre-conditions: Users are currently enrolled in the course and interested in knowing any general announcement or update about a particular course.

Post-conditions: Users will be informed of any new announcements.

Actors: The primary actors in this use case are the students acting as users, and the faculty publishing an announcement.



3.3.7 Use Case #7 (Results)

Author: Dhruv Varshney and Ved Prakash Vishwakarma

Purpose: To enable instructors to upload and manage student results and view overall course statistics, such as averages and performance trends.

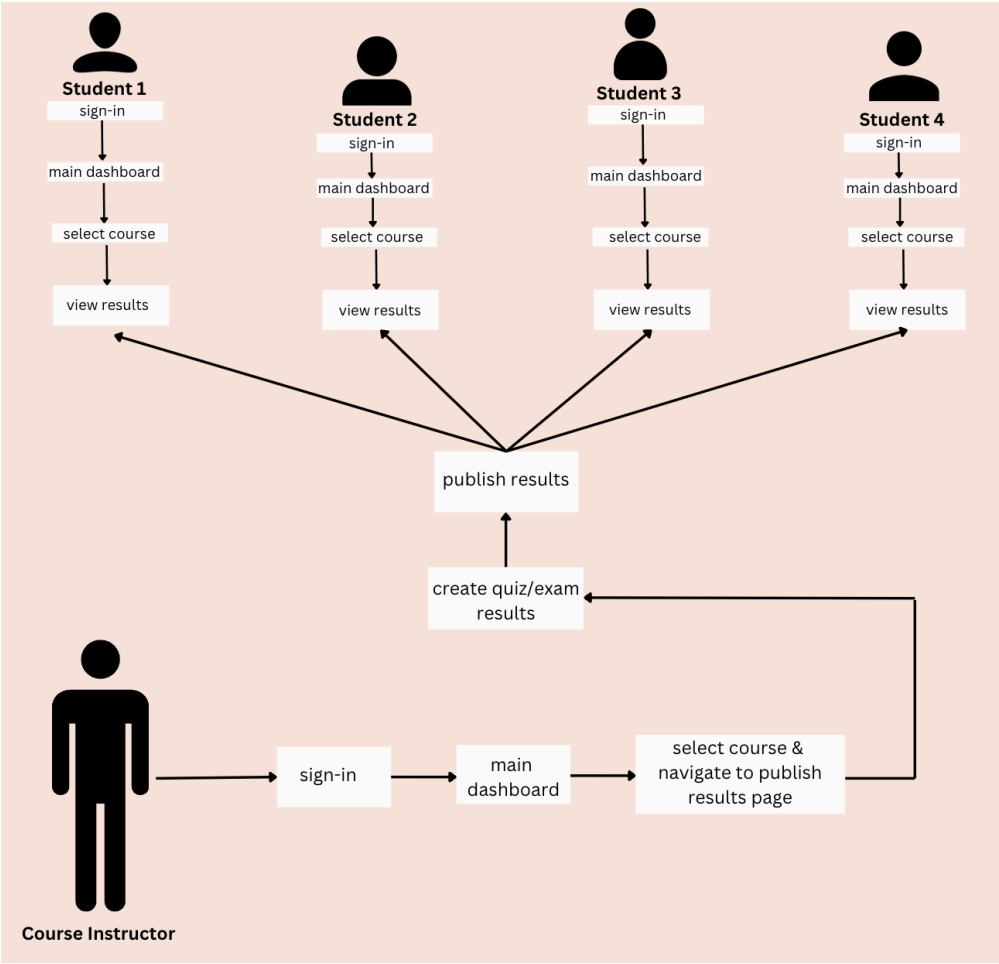
Requirements Traceability: This feature ensures instructors can analyze students' results and course performance to enhance assessments and delivery strategies.

Priority: High

Pre-conditions: The course must be active with enrolled students, grading criteria predefined, and the instructor authorized to manage results.

Post-conditions: Results are uploaded and stored, overall course statistics are displayed, and insights into student performance are provided for analysis. The students can also view graphs and other visual analyses on the basis of their results data.

Actors: The primary actors in this use case are the instructors acting as users.



3.3.8 Use Case #8 (Lectures)

Author: Akash Verma and Aditya Gautam

Purpose: To allow students to access lectures and resources of the course at a single place, at the same time allowing easy and structured distribution of course material for course instructors. Instructors and students often encounter significant challenges in managing course materials in a structured and organized manner, which our product will solve.

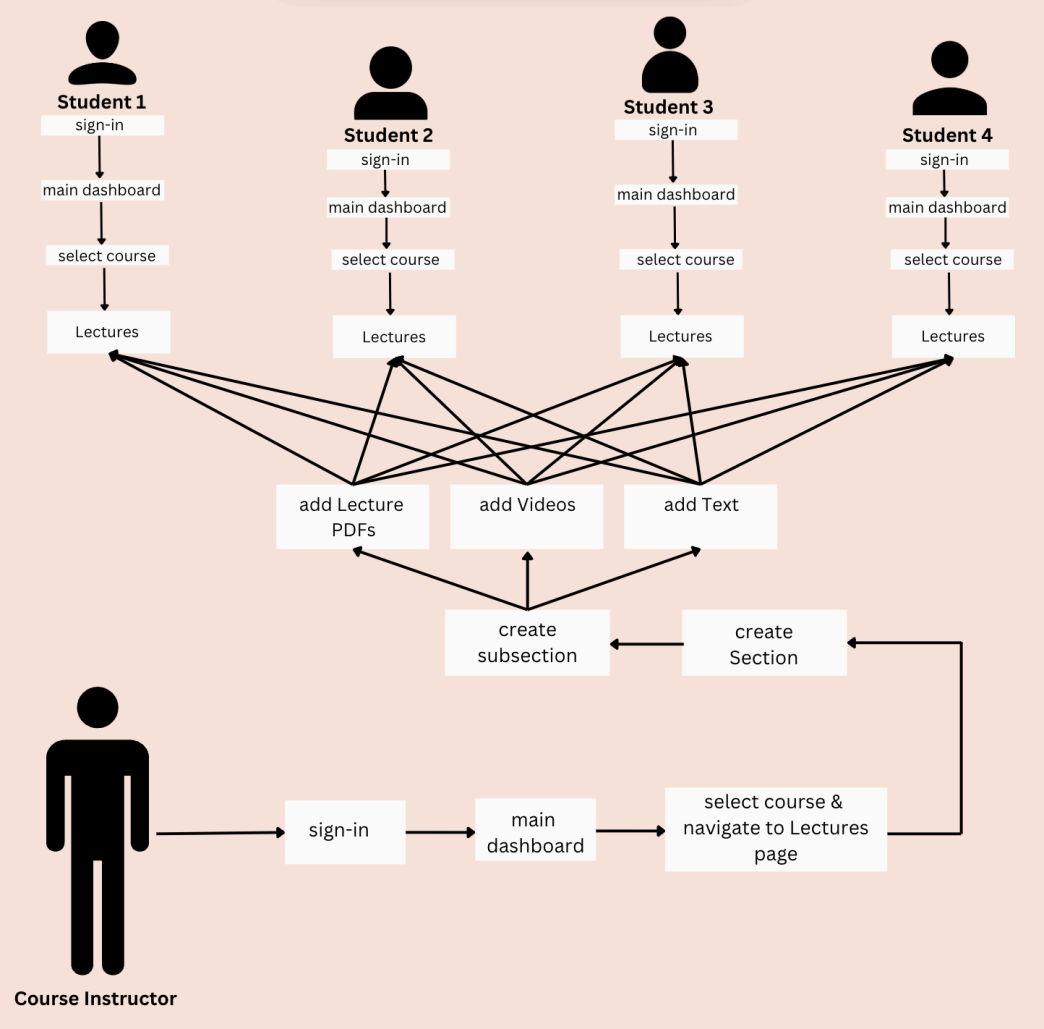
Requirements Traceability: Students should be able to view PDFs, text, and videos(tentatively embedded YouTube videos) uploaded by the Faculty, all of which should be neatly organized in sections and subsections.

Priority: High

Pre-conditions: The student must be enrolled on the course. The Lectures and course works of all courses must be recorded and updated in the system database.

Post-conditions: Users are able to access various Lecture materials.

Actors: The primary actors in this use case are the students and course instructors



3.3.9 Use Case #9 (Overall Performance History)

Author: Aryan Bansal and Ved Prakash Vishwakarma

Purpose: To allow students to see their templates and grades.

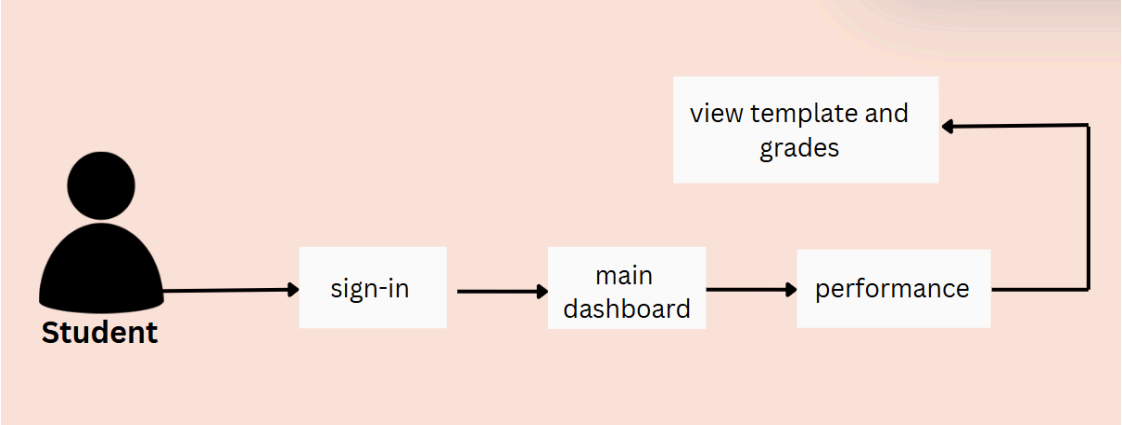
Requirements Traceability: Templates of the department of students are displayed with crucial information like their pending, failed, current, and completed courses, along with grades from completed courses, helping them manage courses and evaluate their performance effectively.

Priority: Medium

Pre-conditions: The student must be registered.

Post-conditions: The user can see his template and grades.

Actors: The primary actors in this use case are the students acting as users.



.

4 Other Non-functional Requirements

4.1 Performance Requirements

- **Concurrency -** The platform can handle 4000 concurrent users.
- **Responsiveness -** The average response time should be below 1 second during normal usage for a seamless user experience. During peak usage, it should be less than 3 seconds.
- **Resource Efficiency -** The product must adhere to industry best practices for storage, memory, and processing to optimize performance and minimize resource consumption.

4.2 Safety and Security Requirements

- Proper login mechanisms should be used to avoid hacking.
- Periodic and safe backups of the entire database will be taken to ensure no loss of data.
- Secure and reliable transmission of information: Secure storage of passwords will be done after encrypting the passwords.
- The minimum set of browsers that have the required safety standards are:
    - Apple Safari 18+
    - Google Chrome 134+
    - Microsoft Edge 131+
    - Mozilla Firefox 130+.
- Users will be prohibited from setting weak passwords, which are defined as those that are less than eight characters in length, consist solely of either lowercase or uppercase letters, or include commonly used phrases stored in a predefined database.
- There will be a CAPTCHA system as needed to avoid DDOS attacks.
- ‘Forgot password’ features authentication by security questions to avoid misuse.

4.3 Software Quality Attributes

4.3.1 Maintainability

- The architecture, design, implementation and documentation of the software must be such that maintenance overhead is minimized.
- Resolving a security defect, including updating relevant documentation and performing necessary testing, must be completed within a maximum of two working days.
- The average effort required to implement a minor feature, including documentation and unit testing, should not exceed one person-week.

4.3.2 Availability

- In case of a server crash, the system state must be restored within one hour.
- A large number of students concurrently access TRACKit just before exams. The system should be robust enough to provide all the functions of the calendar and booking in peak time.

4.3.3 Reliability

- The Mean Time to Failure (MTTF) must exceed three months.

- The system must undergo comprehensive feature testing, load testing, and regression testing before its release or deployment.
- The system should be reliable in giving correct data consistently. Data should not accidentally get irreversibly corrupted in any case.

4.3.4 Portability

We are using portable Web Frameworks to design the front-end and back-end of our application, thus our application will be portable, responsive and can run on any modern web browser.

Appendix A – Data Dictionary

We will make our data directory on the basis of the object-oriented programming paradigm. Authored by Aditya Gautam.

Classes	Description	Methods and Attributes	Datatype
`user`	The overarching class which defines users of the product. Other user classes inherit its attributes like `user_id`, and `password`.	`user_id`	string
		`password`	string
`system_admin`	The controlling user class capable of creating new users, modifying existing users, creating courses and assigning faculty and students to them.	`create_user`	function
		`modify_user`	function
		`create_course`	function
		`modify_course`	function
`student`	The end-user class which consumes other data published by the faculty user class, and can communicate with other students via the forum. The student can exist in the states of being enrolled in particular courses.	`name`	string
		`branch`	string
		`batch`	string
		`roll_no`	int
		`create_forum_post`	function
		`create_forum_post_comment`	function
		`enrolled_courses`	list of course
`faculty`	The end-user class which produces other data consumed by the student user class. The faculty can exist in the states of being the instructor of particular courses.	`name`	string
		`branch`	string
		`create_lecture_module`	function

		`create_announcement`	function
		`create_forum_post`	function
		`create_forum_post_comment`	function
		`create_calender_event`	function
		`create_result`	function
		`create_course_description_entry`	function
`course`	A paramount Class Variable which contains within itself a multitude of other Class Variables like `lecture_module`, `calender_event` etc. It has two primary fields: one is a variable of the Class faculty, while the other is a list of variables of the Class student.	`course_name`	string
		`course_id`	string
		`course_semester`	string
		`course_faculty`	`faculty`
		`course_students`	list of `students`
`lecture_module`	A Class Variable created by a faculty which appears in the *Lectures* section and can contain PDFs, text, and embedded videos.	`lecture_text`	string
		`lecture_pdf`	pdf document
		`lecture_video`	embedded youtube video
`calender_event`	A Class Variable created by a faculty which appears in the *Calendar* section and describes an event's title, time and date.	`event_title`	string
		`event_time`	time
		`event_date`	date

announcement_entry	A Class Variable created by a faculty which appears in the *Announcements* section and describes an Announcements title and text.	announcement_heading	string
		announcement_text	string
exam_result_db	A Class Variable created by a faculty which defines the results of all students for a particular exam in a particular course.	exam_name	string
		exam_max_marks	integer
		student_result_table	table of results of students
course_description_entry	A Class Variable created by a faculty which appears in the *Course Home* section, and describes aspects of a course like grading scheme.	course_description_entry_heading	string
		course_description_entry_text	string
forum_post	A Class Variable created by a student or a faculty which defines the title and text of a forum post.	forum_post_heading	string
		forum_post_text	string
forum_post_comment	A Class Variable created by a student or a faculty which defines the title and text of a forum post's comment.	forum_post_comment_heading	string
		forum_post_comment_text	string
student_template	A static predefined table-like Class which details a student's template as decided by his/her branch.	template_data	dataframe

Appendix B - Group Log

Since the beginning of the project, our entire team has been very enthusiastic. Apart from following offline meetings, we have formed a Whatsapp group and Discord server for effective communication. Log Maintained by Akash Verma.

DATE	TIMINGS	DURATION	AGENDA AND OUTCOMES
Jan 05, 2024	12:00 - 14:00	2 hrs.	● Idea suggestions by everyone ● The main ideas selected were : 1. Campus online shopping app 2. Personal finance tracker 3. Online ride booking app 4. Academic management portal
Jan 07, 2024	10:00 - 11:00	1 hrs.	● Discussed pros and cons of each of the idea, and finalised 2 ideas for further discussion
Jan 10, 2024	14:00 - 16:00	2 hrs.	● Discussed our ideas with the prof and finalised the idea of making an academic management portal . ● Explored and discussed various features for the academic management portal .
Jan 14, 2024	15:00 - 16:30	1hr 30 min	● Discussed the ideas presented by each team and refined them ● looked upon overall structure of website and worked on site navigation
Jan 15, 2024	17:00 - 17:45	45 mins	● Started work on SRS document and discussed its various subheadings.
Jan 17, 2024	18:00 - 19:30	1hr 30mins	● Discussed UI of website ● Divided the work of making SRS document among the group members
Jan 18, 2024	14:00 - 19:00	5 hrs	● Designed the UI for each page in Figma and standardized the design elements to ensure consistency across all pages. Discussed and standardized the design of charts used in the SRS.
Jan 19, 2024	14:00 - 17:00	3 hrs	● Reviewed the work of SRS document done by each team.
Jan 21, 2024	16:30 - 17:30	1 hrs	● Engaged in an introductory meeting with the TA.

			● Discussed various technical doubts related to our software. ● Clarified doubts regarding SRS
Jan 22, 2024	19:00 - 1:00	6 hrs	● Continued working on SRS document.
Jan 23, 2024	14:00 - 16:30	2 hrs 30 mins	● Proofread the document and corrected any irregularities
Jan 24, 2024	16:00 - 22:00	6 hr	● Finalised SRS for submission