Software Requirements Specification

for

ProfInfo Central

Version 1.0

Prepared by

Group 2:

Group Name: Error 404:Team not found

Pavani priya	220415	gppriya22@iitk.ac.in
Sontam Deekshitha	221075	sontamd22@iitk.ac.in
Lakshyta Mahajan	220581	lakshyta22@iitk.ac.in
Kartik	220503	kartik22@iitk.ac.in
Mohd Nasar Siddiqui	220661	snasar22@iitk.ac.in
Atharv Moghe	220250	atharvm22@iitk.ac.in
Prabhat Kumar Yadav	220774	prabhatky22@iitk.ac.in
Nilesh Maneshwar	220715	mnilesh22@iitk.ac.in
Kuldeep Sandip Thakare	220557	kuldeeps22@iitk.ac.in
Sanapala Jaswanth	220955	sjaswanth22@iitk.ac.in

Course: Cs253

Mentor: Abhilash Chandra

Date of Submission: 26-01-2024

CONTENTS

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Error 404: Team not found	Built a website where students can access all the professional and the research-oriented information regarding many different professors of the campus from different departments and can also request for projects under the professors according to their specific interests.	26/01/24

1 Introduction

1.1 Product Scope

The specified software is a comprehensive project management website designed to streamline the project allocation process for students under professors at IIT Kanpur. The primary objective is to simplify and enhance the experience for students by providing a centralized platform where they can effortlessly explore a diverse range of projects offered by different professors. This platform enables students to master relevant skillsets within specific domains, gaining valuable exposure in the process. The website facilitates a seamless interaction between students and professors, allowing students to send project requests with all the necessary credentials specified by the professors. Students can search for projects based on domains, specific professors, or branch-wise preferences

On the professor's end, the software provides a unified platform for managing student requests, eliminating the need to sift through numerous emails. Professors can easily view a list of student requests, including relevant credentials such as CPI, resumes, and experience details. Additionally, professors can communicate with all students simultaneously, keeping them informed about specific activities and posting materials to help them prepare for tests or interviews. Overall, the software aims to optimize project management workflows for both students and professors, saving time and enhancing collaboration within the academic community at IIT Kanpur.

1.2 Intended Audience and Document Overview

The intended audience for this innovative online platform includes undergraduate and graduate students across various disciplines who are seeking meaningful collaborations with professors for research projects. By providing a centralized repository of crucial information about professors, such as their research focus, contact details, and project availability, this platform streamlines the process for students looking to connect with mentors and is especially beneficial for students navigating the challenging task of identifying suitable professors for their research endeavors, offering them a user-friendly interface to submit project requests efficiently. We want to make things easier by revolutionizing the traditional approach to engaging with professors, fostering a dynamic environment that encourages collaboration and accelerates the initiation of research projects.

This document further contains

- Section 2 of the overall product documentation typically covers the following aspects:

  - Product Overview:

    - Provide a concise yet comprehensive overview of the project. This also include the purpose of the product, its key features, and the problem it aims to solve. Consider addressing the target audience and the unique points that make our platform stand out.

  - Functionality:

- Outline the main functionalities of the product. Describe the key features and capabilities it offers. This section provide a clear understanding of what users can expect from the product and how it addresses their needs.

  - Design:
    - Discuss the design principles and considerations that guided the development of the product.
    - This may include user interface design, user experience considerations, and any specific design patterns or methodologies followed during development.
  - Implementation constraints:
    - Identify any limitations or restrictions that influenced the implementation of the product.
    - This may include technical constraints, resource constraints, or external factors that impacted the development process. Be transparent about any challenges faced during implementation.

- Section 3 of your product documentation focuses on specific requirements. Here's brief description of each subsection:

  - External Interface Requirements:
    - Describe how users will interact with the system. This includes details about the graphical user interface (GUI), navigation, and any user input methods. Specify compatibility and integration requirements.
  - Functional Requirements:
    - List and describe the specific features and functions that the product must perform. This could include actions the user can take, system responses, and any other functional aspects. Specify the performance criteria that your product must meet, such as response times, throughput, and resource usage.
  - Use Case Model:
    - Provide detailed descriptions of specific use cases, which represent interactions between users and the system. Include actors (users or external systems) and the steps involved in each use case.
    - Visualize the relationships between actors and use cases using diagrams. These diagrams help stakeholders understand the high-level functionality of the system.
    - For each of these categories, we include both functional and non-functional requirements. Functional requirements describe what the system should do,

while non-functional requirements describe how well the system should do it. Clear and detailed requirements are crucial for guiding the development process and managing stakeholder expectations.

- Section 4 of your product documentation focuses on other non-functional requirements. Here's a brief description:

  - Performance Requirements:

    - Specify the maximum acceptable time for the system to respond to user inputs or requests.
    - Throughput: Define the number of transactions or operations the system must handle within a given time period.
    - Outline the system's capacity to handle a specific number of concurrent users or data volume.

  - Safety and Security Requirements:

    - Describe how sensitive data will be protected, including encryption methods and access controls.
    - Authentication and Authorization: Specify the mechanisms for user authentication and the levels of authorization within the system.
    - Ensure that the system maintains the accuracy and consistency of data throughout its lifecycle.
    - Address any privacy concerns and compliance with data protection regulations.

  - Software Quality Attributes:

    - The system's capability to adjust to changing requirements or environments.
    - The system's ability to maintain performance over an extended period.
    - Emphasizes the system's ease of use and positive user experience.
    - The implementation of measures to protect the system from unauthorized access, data breaches, and other security threats.
    - Includes encryption, authentication, and authorization protocols to ensure data confidentiality and system integrity.
    - Ensures the system's consistent performance and availability to users.

1.3 Definitions, Acronyms and Abbreviations

API	Application Programming Interface
CSS	Cascading Style Sheets
GUI	Graphical User Interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IITK	Indian Institute of Technology, Kanpur
SRS	Software Requirement Specification

1.4 Document Conventions

- Arial font size 11 has been maintained throughout the text.
- The document text is single spaced and 1” margin has been maintained throughout the document.
- Italics have been used for writing comments.

1.5 References and Acknowledgments

- In developing this platform, we took reference from professors' institute websites to gather valuable information including their research focus, contact information, and project availability.

- We extend our gratitude to all professors who embraced our platform, contributing valuable insights and support and expressed their interest, contributing to the success of this initiative.

- Our heartfelt thanks go to our instructor Dr. Indranil Saha and tutor Abhilash for their guidance in crafting this platform.

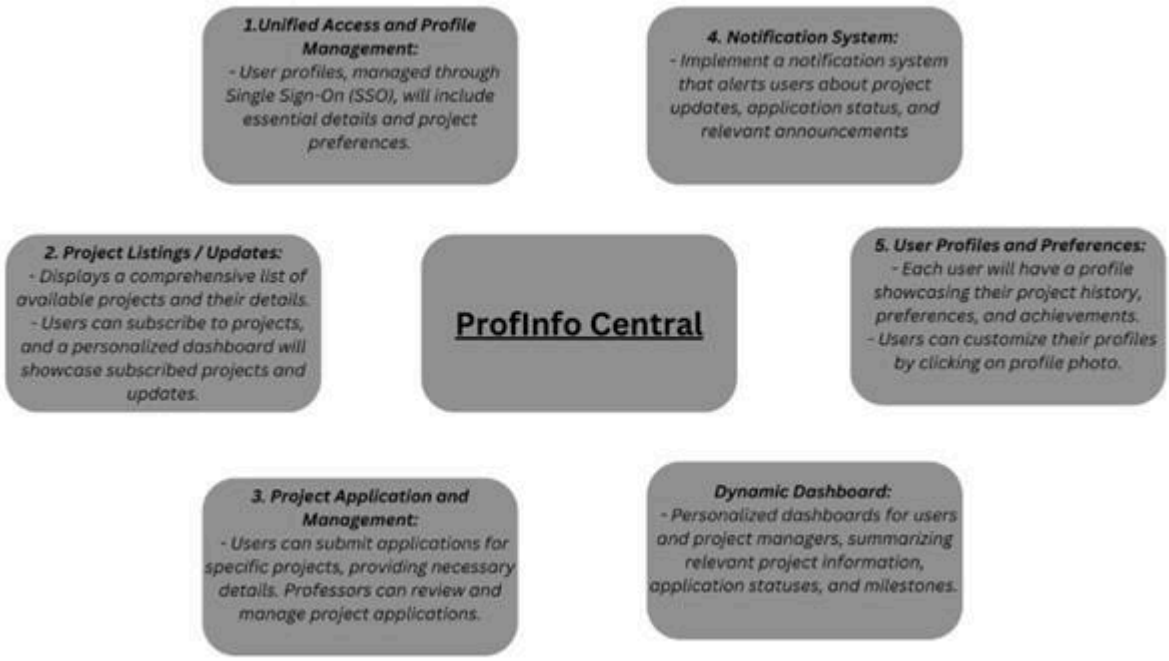
2 Overall Description

2.1 Product Overview

ProfInfo Central is a web-based platform designed to serve the academic community, focusing on providing a centralized hub for professors and students to facilitate efficient project coordination. Recognizing the current challenges in project communication within academic institutions, ProfInfo Central aims to be the go-to solution for accessing project details, coordinating events, and fostering collaboration among students and professors.

The platform addresses the existing chaos in project communication at our college, where students often struggle to navigate through different websites. ProfInfo Central serves as a dedicated space for accessing project-related information, announcements, and communication, eliminating the need for students to sift through disparate channels.

Key Features include a Unified Project Database, User Registration and Profiles, Streamlined Project Exploration, Efficient Project Requests, Professor-Focused Interface, a robust Notification System, and Customized Dashboards for both students and professors. ProfInfo Central aims to simplify the project management process, fostering better collaboration between students and professors while providing a user-friendly experience. The platform's goal is to consolidate project-related information, reduce communication clutter, and facilitate efficient project coordination within the college community.



2.2 Product Functionality

1.Centralized Project Database:
- The system will maintain a centralized database of all projects offered by professors.
- It will categorize projects based on fields of interest and departments.

2.Student Registration:
- Users can register themselves on the platform to access project information.
- Users can browse and search for projects based on their field of interest or department.
- Detailed project information, requirements, and objectives will be accessible.
- Registered users can submit requests to participate in specific projects. Users will provide necessary information to express their interest.

3. Professor Interface:
- Professors will have a dedicated interface to view incoming requests for their projects.
- They can accept or decline project participation requests.

4.User Profiles:
- Each user, including both students and professors, will have a profile. Profiles will store relevant information, project history, and preferences.
- Users will have a personalized dashboard summarizing relevant project information.
- Students can track their project applications and status while Professors can monitor project requests and project team composition.

5.Notification System:
- The system will notify professors of new project requests and users of request status.
- Email notification system will be implemented.

Design and Implementation Constraints

1. **Database Integration:**
- All users and project information must be stored in an accessible database linked to the website.

2.Authentication Requirements:
- To perform any operations, users need to provide accurate usernames and passwords for account login.

3. Response Time Limitation:
- The product must ensure that the loading time does not exceed 2 minutes, ensuring a prompt user experience.

4. Device and Connectivity Compatibility:
- Users should be able to access the system from any computer with internet browsing capabilities and an active internet connection.

5. Basic Computer Skills Prerequisite:
- Utilizing the product requires a general understanding of basic computer skills, ensuring users can navigate and operate the system effectively.

2.3 Assumptions and Dependencies

No specific assumptions or dependencies are considered at this time

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The user interface is designed to operate through web-based platforms, allowing users to remotely connect to the system using various applications such as Microsoft Internet Explorer, Mozilla, Safari, Chrome, among others. This ensures accessibility from a range of devices and browsers.

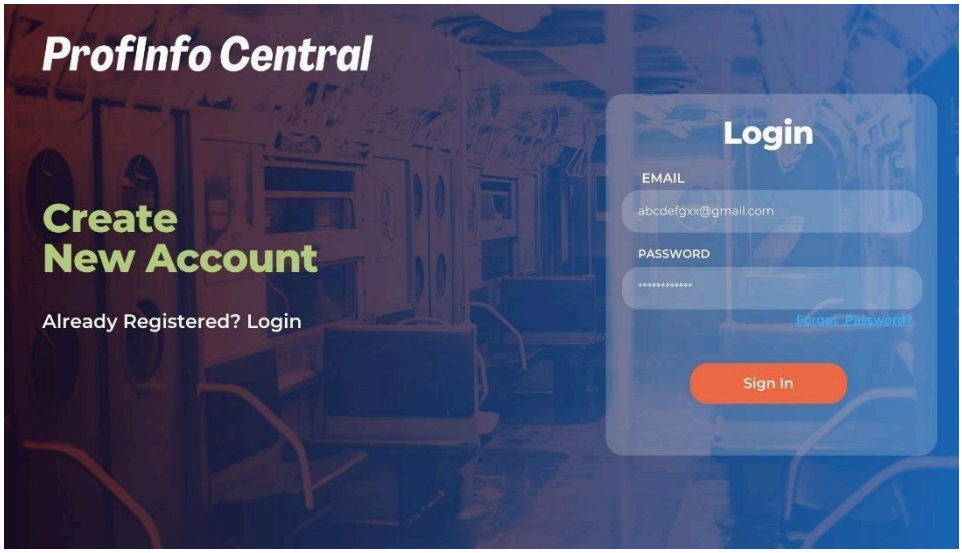
Emphasizing user-friendliness, each component of the interface aims for optimal ease of use. The choice of fonts and buttons is geared towards swift loading on web pages. The web pages are intentionally kept lightweight to facilitate quick loading times, a goal reinforced by the incorporation

of React JS to enhance page loading efficiency and prevent unnecessary refreshes during navigation.

Front-end software - HTML, CSS, JAVASCRIPT, REACT.JS
Back-end software – NODE.JS, EXPRESS.JS, MongoDB

- **Login Page**

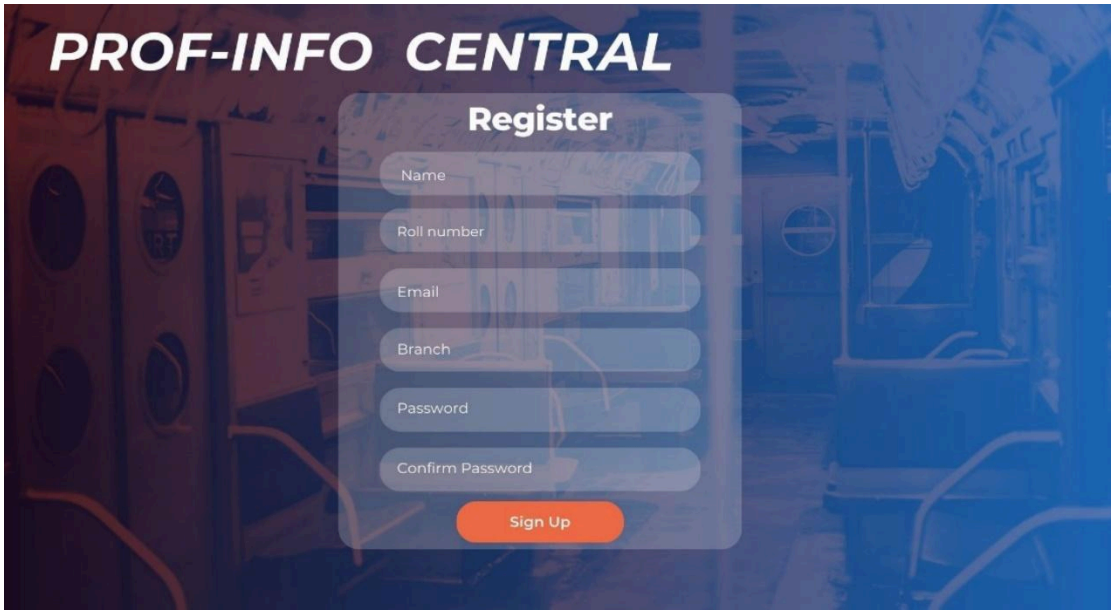
  - Students who have already got registered in the platform will login here with their IITK email ID and password they have created during registration.



- **Registration Page**

- Students who are new to the platform will register here.

- They will enter their Name, Roll number, IITK Email ID, Branch, Password, and will confirm their password after.



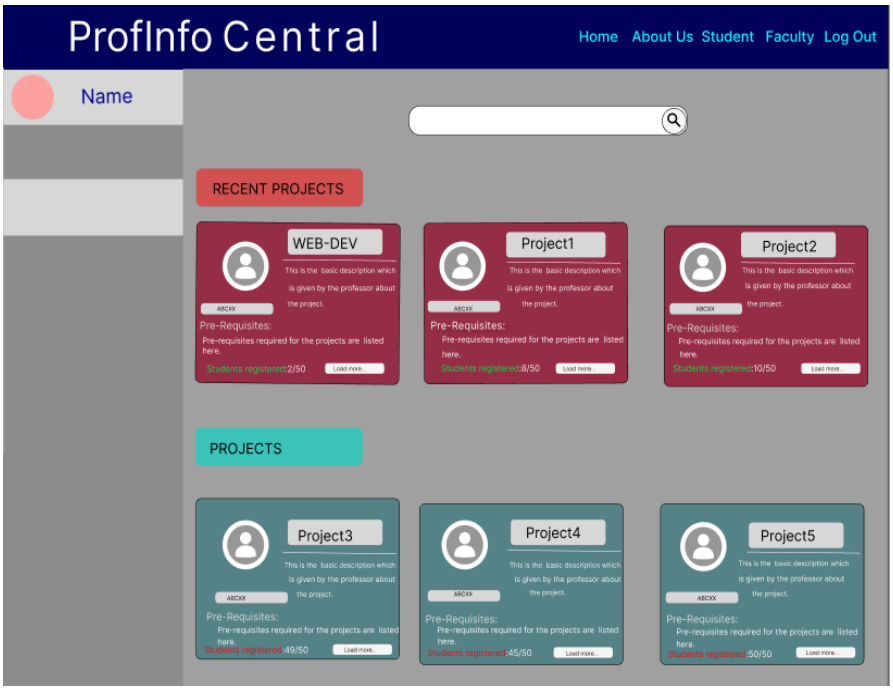
- **Student interface**

1. Home Page(for students)

- After logging into the website, students will be directed to the Home Page, which provides essential information about the platform. On this page, users will find a prominent button labeled "Explore Projects," allowing them to navigate and discover the various projects available on the site.

2. Explore Projects

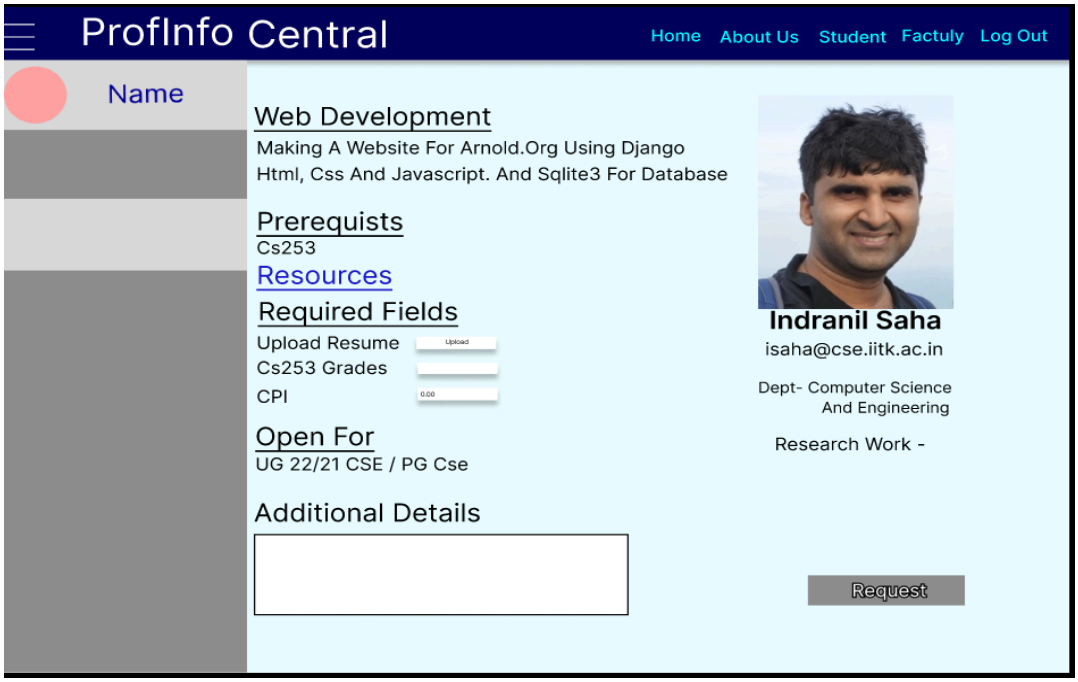
- Students can select the option ‘Explore Projects’ where they can see the projects which are available and the professors who are offering them, so that they can select the one they want to form the available ones.



3. Project Details

After selecting a particular project for which you want to apply, an option like this appears. Students will get all the information regarding the project like its

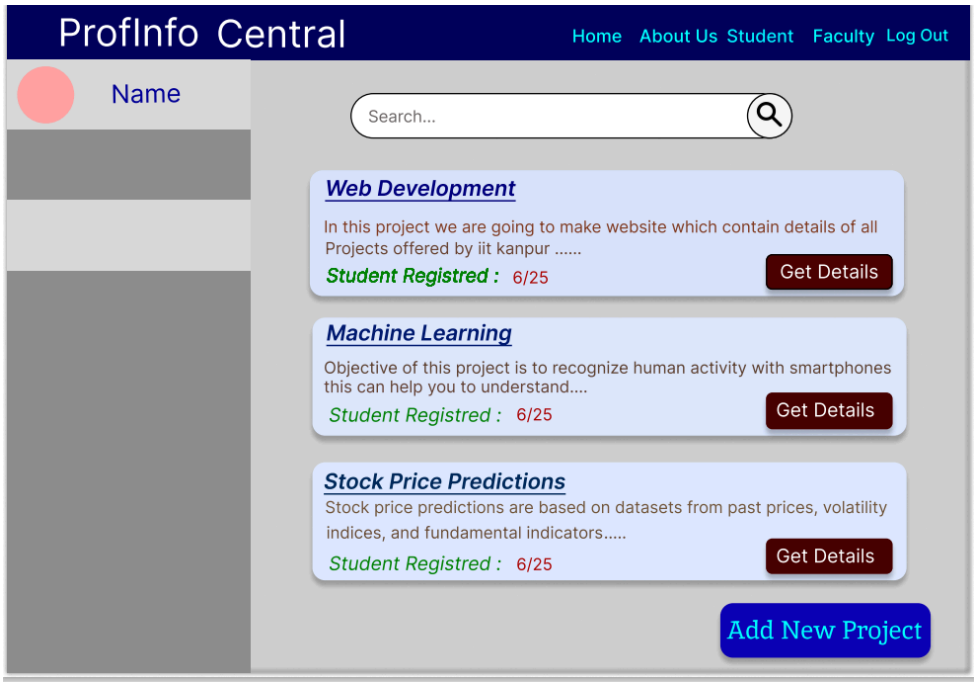
- main topic,
- pre requisites needed,
- open for,
- and also, about the professors who is offering the project



- **Professor Interface**

1. Home page

While coming to the professor's interface, this will be the home page of the professor where he can see the projects he offered and number of students registered and can also add few more by selecting the 'add new project' option.



2.Project page

- As the professor clicks on a particular project to get the details, he will get to know the number of enrolled students for the project and can also see the requests from different students.

- He can then click on the students enrolling request which finally takes him directly to the students' profile

3.1.2 Hardware Interfaces

Hardware required to connect to the internet. For example, Modem, Network Card, Network Connection, etc.

A device with an active internet connection. It can be a phone, a tablet, a desktop, anything with internet access, and a browser

3.1.3 Software Interfaces

In the process of system development, our tasks encompass the design of both static and dynamic website interfaces, the creation of website functions, implementation of a database system, and photo/picture editing. To achieve these goals, a specific set of software requirements is essential.

The requisite software includes:

1. Web Browser (with internet access):

- Utilized during the development phase for debugging and testing purposes.

2. Windows:
- Operating system serving as the development environment.

3. Node.JS, Express.JS:
- The chosen language for backend development in the creation of the progressive web application (PWA).

4. HTML, CSS, JS:
- Fundamental building blocks for constructing the responsive web component, constituting the frontend of the web application.

5. React.JS:
- Employed to enhance the visual appeal of the frontend interface for a more engaging user experience.

6. MongoDB:
- Utilized as the database management system, facilitating communication with the server database.

7. Visual Studio Code:
- Integrated Development Environment (IDE) employed for various operations, including code writing and management.

These software requirements collectively contribute to the comprehensive development toolkit necessary for designing, implementing, and maintaining the envisioned web-based system.

3.2 Functional Requirements

3.2.1 Unified Access and Profile Management:
- The login interface will offer options to log in using the provided credentials
- User profiles, managed through Single Sign-On (SSO), will include essential details and project preferences.

3.2.2 Project Listings / Updates:
- Displays a comprehensive list of available projects and their details.
- Users can subscribe to projects, and a personalized dashboard will showcase subscribed projects and updates.

3.2.3 Project Application and Management:
- Users can submit applications for specific projects, providing necessary details.
- Professors can review and manage project applications.
- Different project types, such as Research etc., with specific criteria for application.

3.2.4 Notification System:

- Implement a notification system that alerts users about project updates, application status, and relevant announcements.

3.2.5 User Profiles and Preferences:

- Each user will have a profile showcasing their project history, preferences, and achievements.
- Users can customize their profiles by clicking on profile photo.

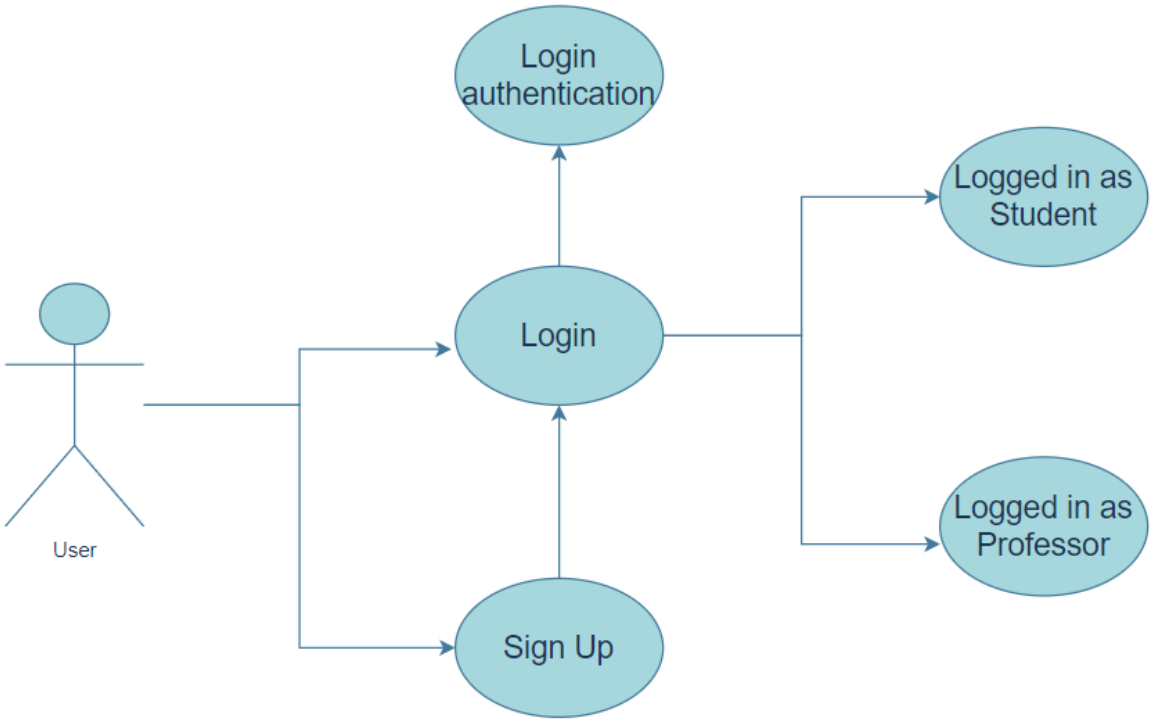
3.2.6 Dynamic Dashboard:

- Personalized dashboards for users and project managers, summarizing relevant project information, application statuses, and milestones.

3.3 Use Case Model

3.3.1 Use Case #1 (Login/Sign up)

Author – Error 404: team not found
Purpose - To sign up for new user and to login as a Student or Professor for the others.
Diagram :

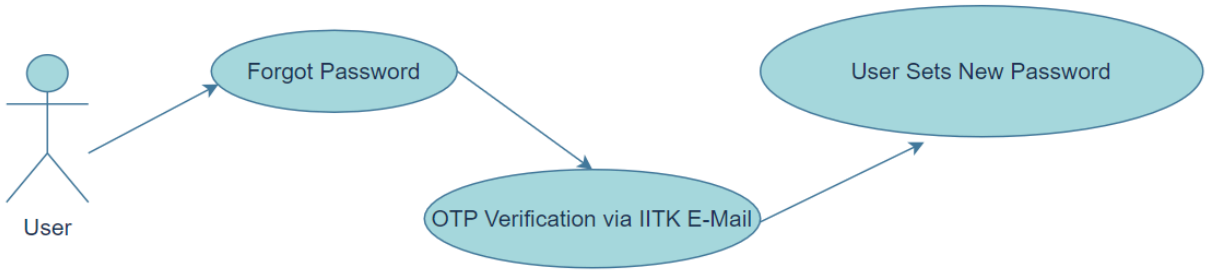


Requirements Traceability – Profile page
Priority - High
Preconditions - The user must be a student or a professor of IIT Kanpur.
Post conditions - User account creator and others logged in.

Actors – Students and Professors.
Exceptions - User can forget the password and use the forget password button.

3.3.2 Use Case #2(Forget Password)

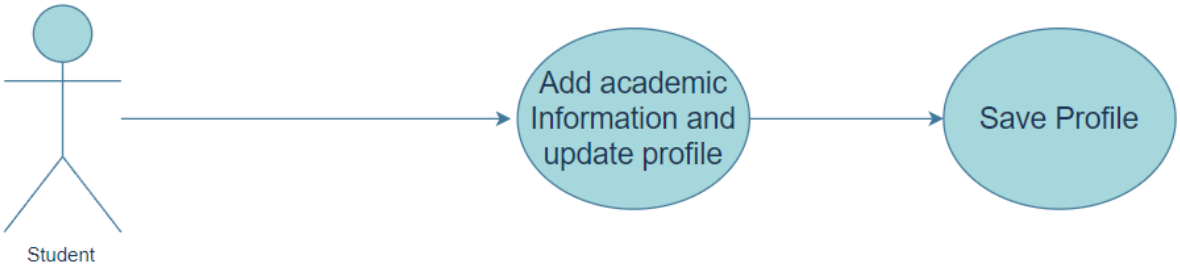
Author – Error 404: team not found
Purpose - This use case is for changing the password of a user's account if they forget it.
Diagram:



Requirements Traceability – Profile Page
Priority - Low
Preconditions - User must have signed up and set a password already.
Post conditions - The password will be changed for future login.
Actors – User

3.3.3 Use Case #3 (Profile Creation and Updation by Student)

Author –Error 404: team not found
Purpose - Creating Student Profile and updating student information.
Diagram

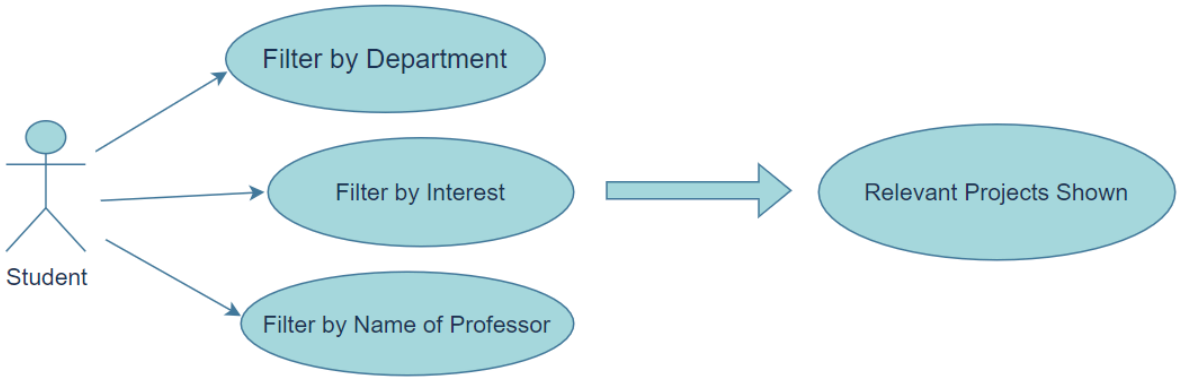


Requirements Traceability – Profile Page.
Priority - High
Preconditions - User should be student of IIT Kanpur.
Post conditions - Student Profile is created and he can apply for project.
Actors – Student.

3.3.4 Use Case #4(Projects Search by Students)

Author – Identify team member who wrote this use case

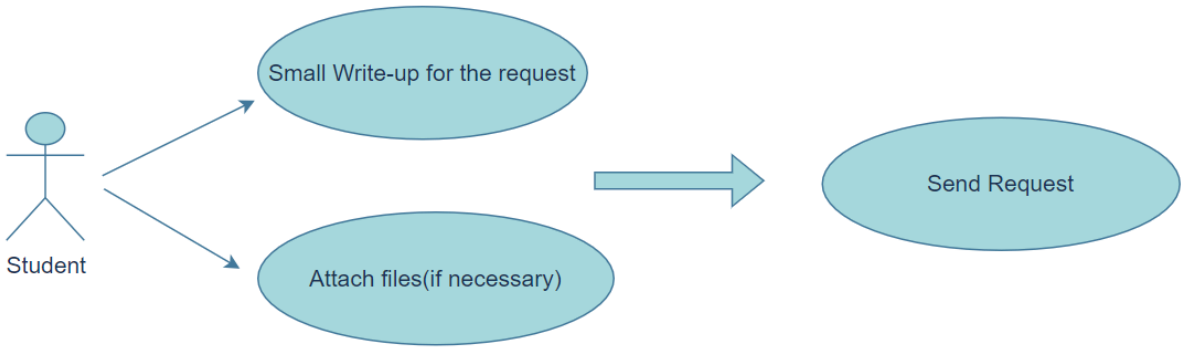
Purpose - What is the basic objective of the use-case. What is it trying to achieve?
Requirements Traceability – Identify all requirements traced to this use case
Diagram:



Priority - Medium
Preconditions - User must be logged in and be on the home page
Post conditions - Projects according to the filters provided by the student will be shown.
Actors – Student

3.3.5 Use Case #5(Requesting Project)

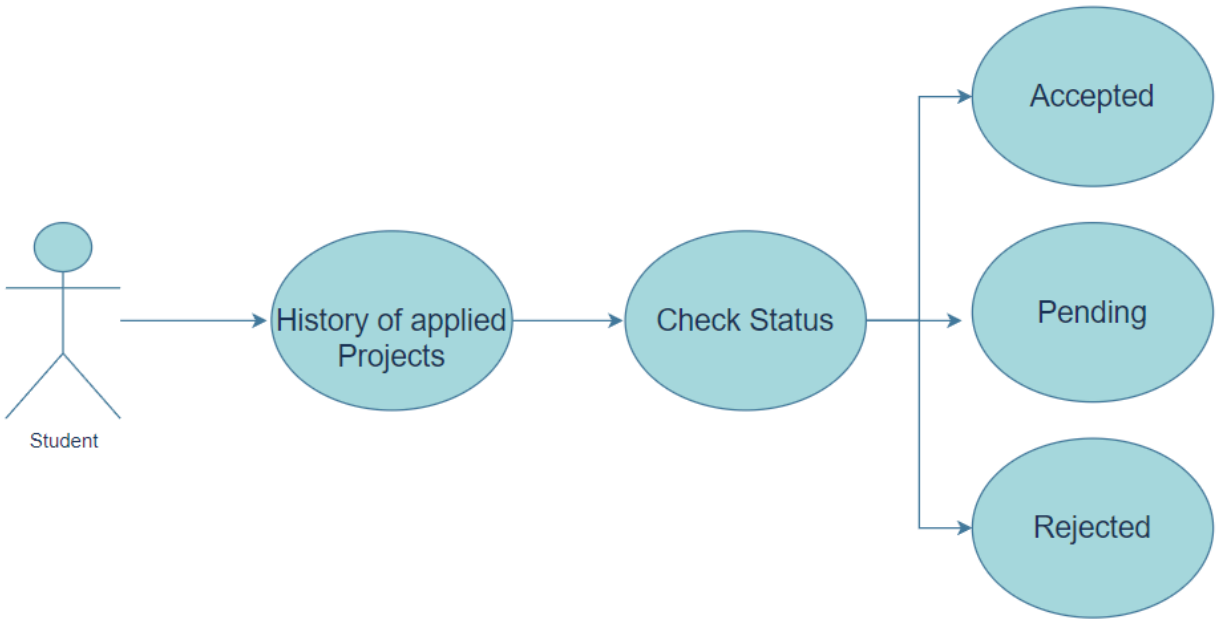
Author – Error 404: team not found
Purpose - This use case enables the student to request a project by sending a notification to the concerned professor.
Diagram:



Requirements Traceability – Project Description Page
Priority - High
Preconditions - Student must be on the required project's page
Post conditions - Student's request will be lodged and the concerned professor will be notified.
Actors – Student, Professor.

3.3.6 Use case #6 (Status Review of the Applied Projects)

Author –Error 404: team not found
Purpose - To check the status the applied projects.
Diagram

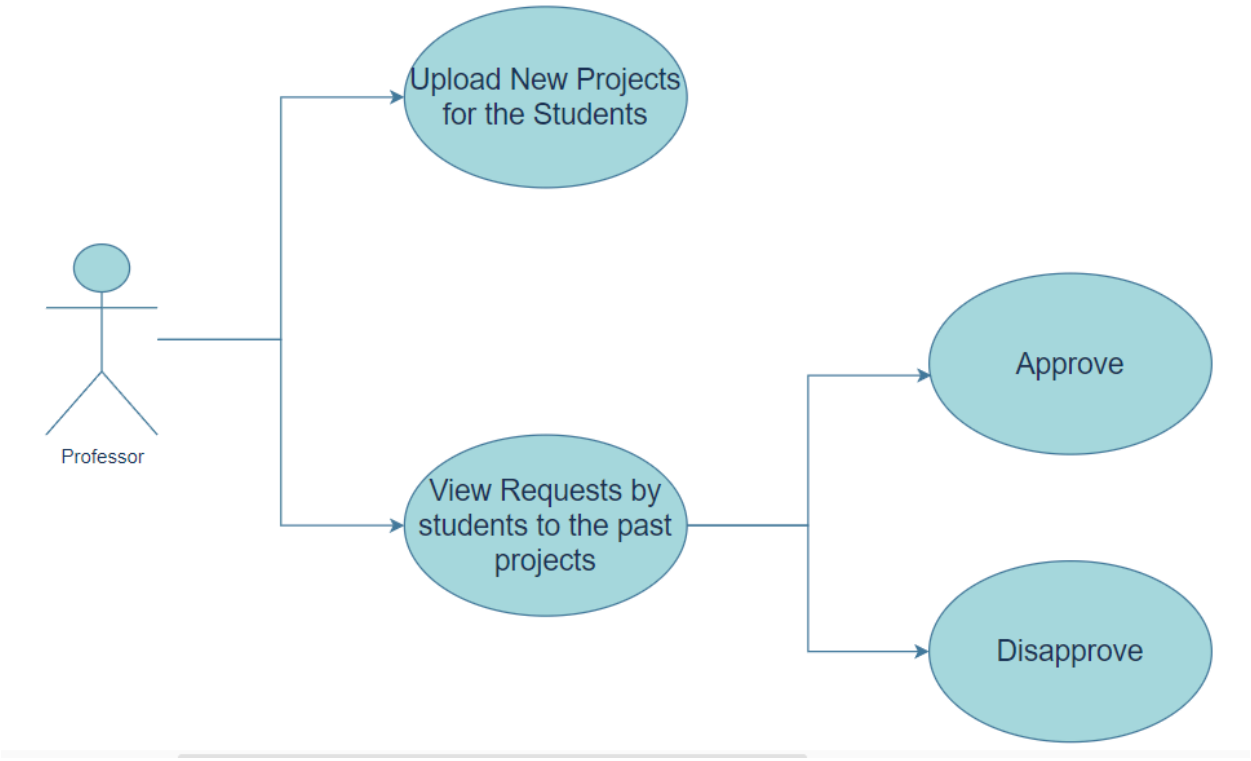


Requirements Traceability - History Page
Priority - High
Preconditions - Student should have applied for the project.
Post conditions - Student will see the status of the applied project.
Actors – Student.

3.3.7 Use Case #7(Upload Projects or View Requests)

Author – Error 404: team not found
Purpose - This use case enables the student to request a project by sending a notification to the concerned professor.

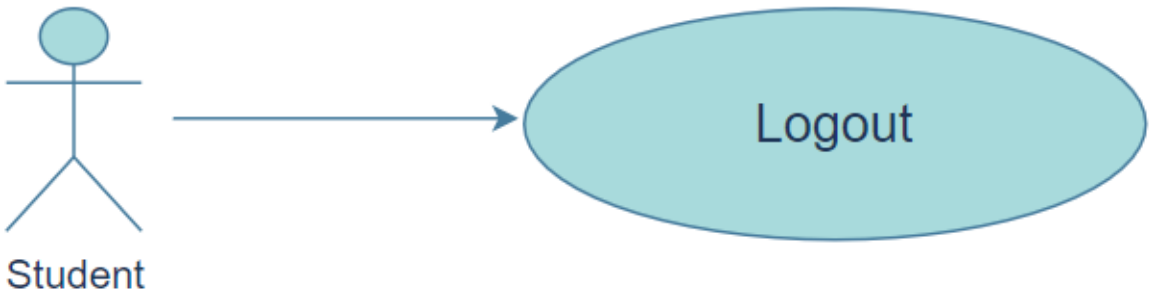
Diagram:



Requirements Traceability – User is professor of IIT Kanpur
Priority - Medium
Preconditions - For Professor Only
Post conditions - Approval or Disapproval of a project
Actors – Professor

3.3.8 Use Case #8(Logout)

Author – Error 404: team not found
Purpose - This use case is triggered when the user wants to logout.
Diagram:



Requirements Traceability – Logout Link.
Priority - High
Preconditions - User must be logged on the website
Post conditions - User is logged out
Actors – User (Student/Professor)

4 Other Non-functional Requirements

4.1 Performance Requirements

The system's responsiveness is a top priority, ensuring swift handling of user requests, even under the load of multiple concurrent requests. Real-time data updates on the client side, coupled with efficient upgrade mechanisms, are integral to maintaining a dynamic and up-to-date user experience.

Ensuring stability and maintainability, the system is engineered to operate continuously without bugs or performance impediments. The hardware and software infrastructure is designed to handle a higher volume of clients, considering the potential wide-ranging user base of the product platform.

The product's accessibility is a key feature, offering compatibility with various platforms while dynamically adapting to the specific hardware and software configurations of individual clients. Notably, the system remains responsive even on less powerful hardware, ensuring a seamless experience for users across different devices.

This dynamic nature extends to the scalability and upgradability of the product. Changes and updates can be seamlessly implemented without substantial system shutdowns. The scalability is designed to be flexible, accommodating the continuous growth of the client community the product aims to serve.

4.2 Safety and Security Requirements

Passwords will be saved encrypted in the database to ensure the user's privacy.
The user's IP address will be logged.
The system will be protected against vulnerabilities such as SQL injection attacks.
Log in for students will be based on CC user ID

4.3 Software Quality Attributes

4.3.1 Adaptability:

The application is designed with a focus on flexibility, allowing seamless integration of new requirements into any module of the system. Employing a modular format ensures that future changes, be they additions or deletions, can be easily incorporated, maintaining the adaptability of the system.

4.3.2 Portability:

Ensuring broad accessibility, the application is effortlessly portable on any Windows-based system. The frontend, constructed with React JS, transforms it into a responsive and progressive web app, enabling the application to operate seamlessly across diverse platforms.

4.3.3 Sustainability:

The product's architecture, design, implementation, and documentation are geared towards minimizing maintenance costs. Fixing security defects, inclusive of regression testing and documentation updates, aims for a maximum person-time not exceeding two persons per day. Emergencies may necessitate system downtime or disabling offending features. Minor enhancements, including testing and documentation, target an average person-time not exceeding one person per week.

4.3.4 User-Friendliness:

The frontend of the software application prioritizes user-friendliness, adhering to the principle of usability: Keep it Simple and Stupid (KISS). This design philosophy ensures effective utilization of the system by end-users.

4.3.5 Security Measures:

The application implements robust security measures by password protecting access and restricting entry upgrades and deletions to privileged users. User credentials, including custom passwords, are stored in an encrypted format in the database to enhance security. For password recovery, security questions set during account creation are employed, followed by email notifications to the user's IITK email ID or other registered email address.

4.3.6 Reliability and Availability:

The system incorporates regular database backups to safeguard user data, including passwords, posts, achievements, and activities. Emphasizing reliability, the system ensures it is consistently running and ready to execute tasks whenever users require its services.

Other Requirements

Appendix A – Data Dictionary

A.1 User Class

Element Name	Operations	Requirement	Description
Email Address	Connect user to our platform.	IIT kanpur Email address.	It takes IITK Email address as an argument and associates it with the user.
Password	Secure User login.	String, int	It takes the password as argument then checks the condition of login.
User Profile Data	Store user info	string, int	Maintain the data of each user.

A.2 User Interface Class

Element Name	Operations	Requirement	Description
Registration	First time registration and data input.	IITK Email address, string ,int	It takes the name,roll no, branch, IITK email, password, as arguments then checks the condition of registration and returns the status of registration.
Login/Logout	Enables login/logout functionality for user.	Email, password	It takes email and password as arguments then checks the condition of login and returns the status of login.

A.3 Student Interface Class

Element Name	Operations	Requirement	Description

Search Project.	Search for projects.	string	Enables students to search for project under any professor, any branch, or their area of interest.
Apply for project.	Request the prof for the project.	string	Enable student to request the concerned prof for the project.
Project status.	Show the status of applied project	string, int	Check whether the prof accepted or rejected your application for project.

A.4 Professor Interface Class

Element Name	Operations	Requirement	Description
Add project.	Adding project under prof.	string, int	Adds project under prof and students would able to apply for this.
Review requests.	Can accept or reject the requests.	string	Prof can if the student is eligible for his project and accept or reject his request accordingly.
Edit project details.	Change project resources or requirements.	string, int	Prof can alter the resources or requirements of a project as per need.

Appendix B - Group Log`

SL. NO	DATE	TIME	VENUE	DESCRIPTION
1	09/01/24	7:00pm-8 :00pm	RM	Thought for an idea focussing on major issues faced by IITK student's junta.
2	11/01/24	3:00pm-4 :00pm	RM	Had debate on the ideas we got, and all together concluded an idea.
3	12/01/24	12:00pm- 1:30pm	RM	Established our idea in a proper way and registered our project.
4	16/01/24	7:00pm-8 :30pm	RM	Discussed our idea with TA and considering his views, we planned how to do our project.
5	18/01/24	3:00pm-5 :30pm	RM	Distributed the work among ourselves and first started designing our project interfaces.
6	20/01/24	4:00pm-6 :30pm	RM	Explored some more functionalities for the product and progressed with the SRS document.
7	21/01/24	4:00pm-6 :30pm	RM	Explored various implementation tools and software for the project.
8	23/0124	2:00pm-6 :00pm	RM	Made sure to complete 80% SRS documentation work
9	24/01/24	4:00pm-6 :00pm	RM	Done with SRS documentation work and checked it thoroughly.
10	25/01/24	7:00pm-8 :00pm	RM	Had another meeting with TA and discussed our SRS documentation with him