Project Management System (ProBEE)

Software Requirements Specification

1. Introduction

This document outlines the functional and non-functional requirements for a project management system with two user types: **Instructor** and **Student**. The system will allow instructors to create and manage projects and students to upload project documents.

2. Functional Requirements

2.1 User Roles and Login

- There are two user roles: **Instructor** and **Student**.
- A **Login Page** will allow users to log in using their username and password.
- Upon successful login, users will be directed to their respective dashboards.

2.2 Instructor Functionalities

- **Create Project**: The instructor can create a new project by specifying the project name, description, and students to new projects.
- **List Projects**: The instructor can view a list of all projects they are supervising.
- **View Project Details**: For each project, the instructor can view project details, including:
  - List of students assigned to the project
  - Project documents uploaded by the students
  - General project information (Description, etc.)

2.3 Student Functionalities

- **Access Project Details**: Students can access the page showing their assigned project details.
- **Upload Project Documents**: Students should upload documents related to their project.

3. Non-Functional Requirements

- **Security**: The system must use a secure login (username and password)
- **Usability**: The user interface must be intuitive and responsive, ensuring easy navigation and accessibility.

4. User Interface Design

The system will be developed using **React** for the front end. Each role will have different user interfaces:

Login Page

Instructor Dashboard

- **Project Creation**: A form to enter project name and description.
- **Project List**: A table listing all projects where the instructor is the supervisor.
- **Project Details Page**: Displays project details, assigned students, and project documents.

Student Dashboard

- **Project Details Page**: Shows the assigned project's information and allows the student to upload documents.

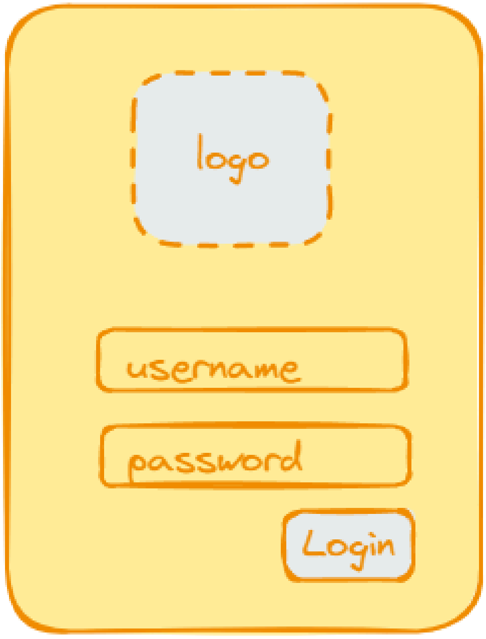
5. Integration and API

- **Frontend-Backend Integration**: React will handle the user interface, while the backend will manage the logic and database interactions.
- **Optional API Integration (+10 points)**

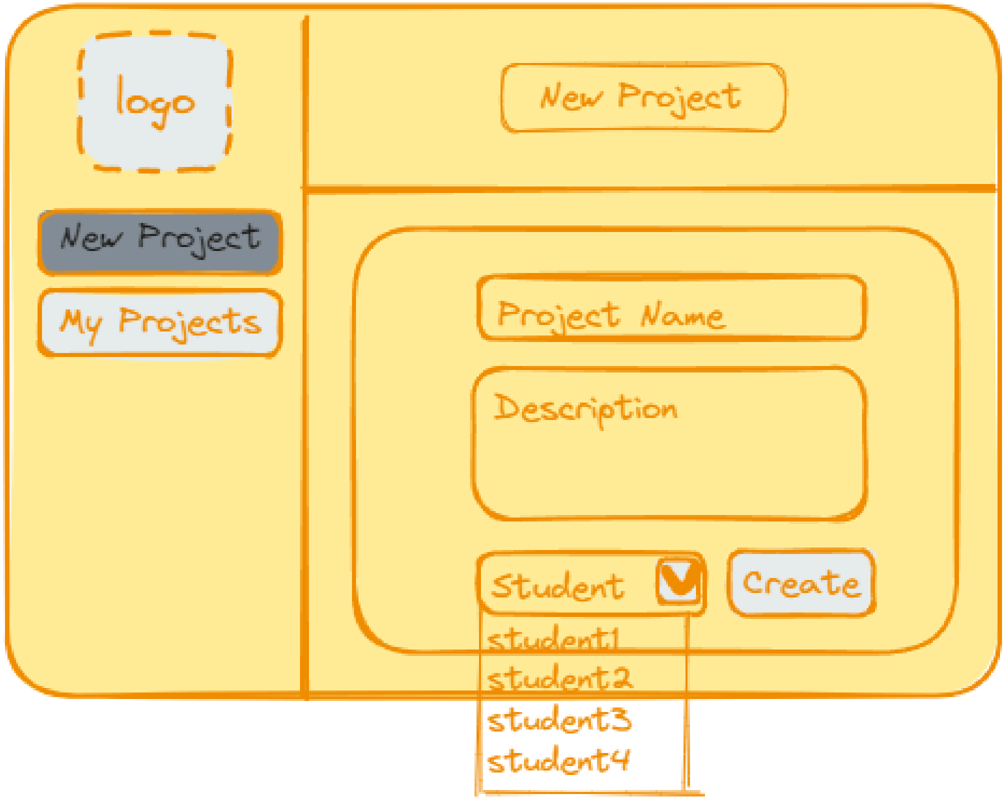
6. Development Considerations

- **Technology Stack**:
  - **Frontend**: React
  - **Backend**: Node.js, Express, or Django
  - **Database**: MySQL, PostgreSQL, or MongoDB
  - **Optional**: RESTful API for communication between frontend and backend

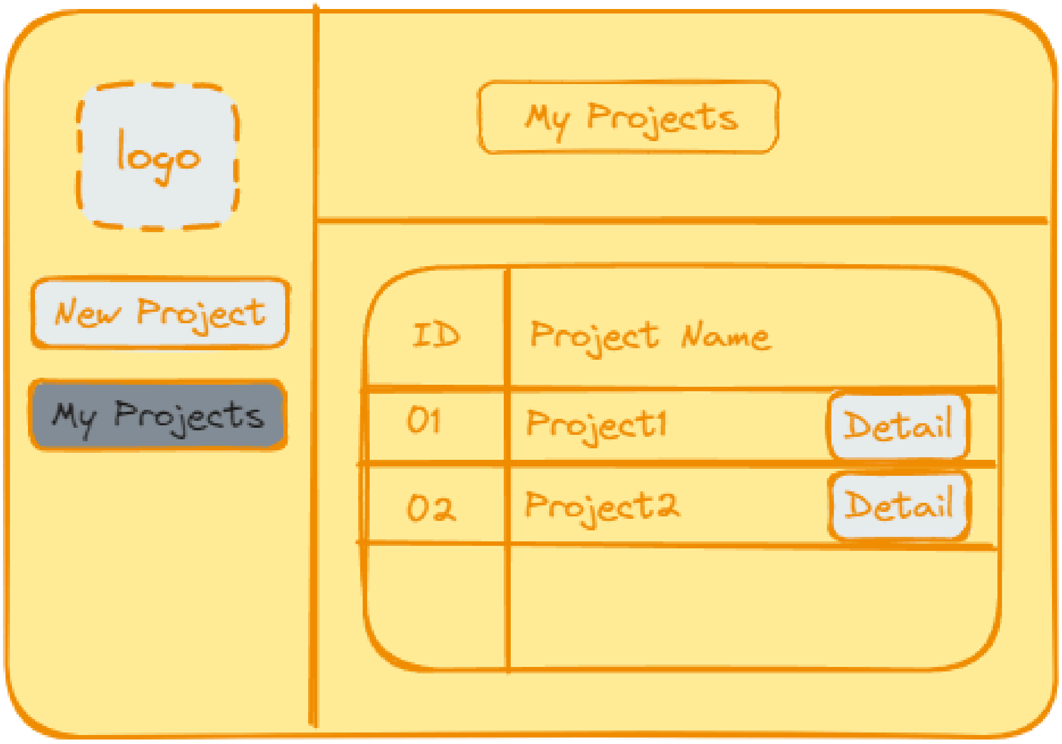
7. Sample Screens



Screen_1: Login Page

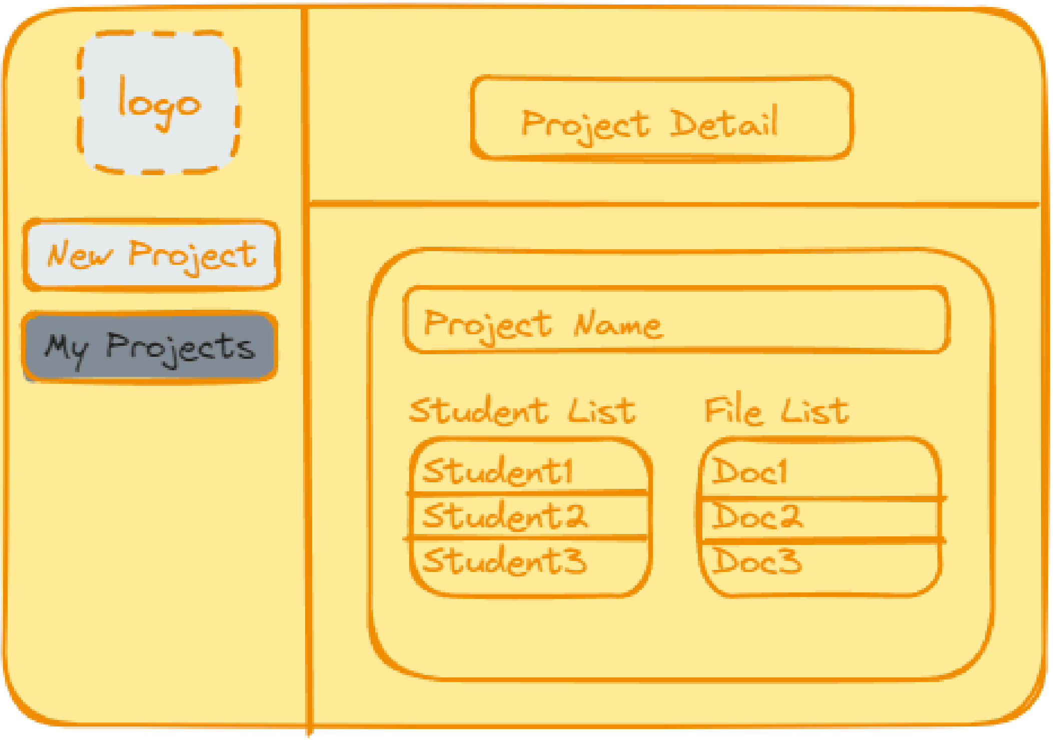


Screen_2: New Project



ID	Project Name	
01	Project1	Detail
02	Project2	Detail

Screen_3: My Projects



Student List	File List
Student1	Doc1
Student2	Doc2
Student3	Doc3

Screen_4: Project Details