Software Requirement Specification for TAC Portal

Name	RITHU SHREE M
Roll no	7376222BT179
Seat no	349
Project ID	29
Problem Statement	BIT- INTERVIEW CANDIDATE DETAILS

1. Introduction

1.1. Purpose:

The purpose of this project is to store the BIT-interview candidate details in a webpage. Admin can view the details of the candidates including selection status. Whenever required they can easily view the details of the onhold candidates and call for the job . The users can access the page at any time using their ID and password.

1.2. Scope of Project:

- This software system will serve as a portal for the Technical Approval Committee (TAC), enabling students to submit their projects and receive their rewards. From an administrative perspective, this system will provide a comprehensive **analytical dashboard for project oversight**.
- Administrators have the ability to approve or reject projects. Once a project is approved, students can schedule an appointment using their accepted PTAC

ID. The system will calculate the number of days between the approval date and the current date. If this duration is less than 30 days, a warning will be displayed indicating that students can claim only 30% of their rewards, which is contingent on the number of days since approval.

2. System Overview:

2.1. Users:

1. **Students**:

They have the ability to submit applications for TAC approval, upload relevant project documents, monitor the status of their application, schedule appointments following approval, and review their TAC interaction history.

2. Admins:

Review submitted TAC applications, approve or reject applications (with remarks), manage appointments, schedule meetings, and access analytical dashboards for project oversight

2.2. Features:

1. Login and registration:

Students can register for an account or login with their existing account

2. TAC Application Submission:

Students can input relevant details regarding their project application including project title, description, objectives, and any necessary attachments. Upon completion, the application is submitted to the admin interface for review and further processing

3. Application Status:

Students can view the current status of their application and also see the history logs in the option Activity

4. Appointment Booking:

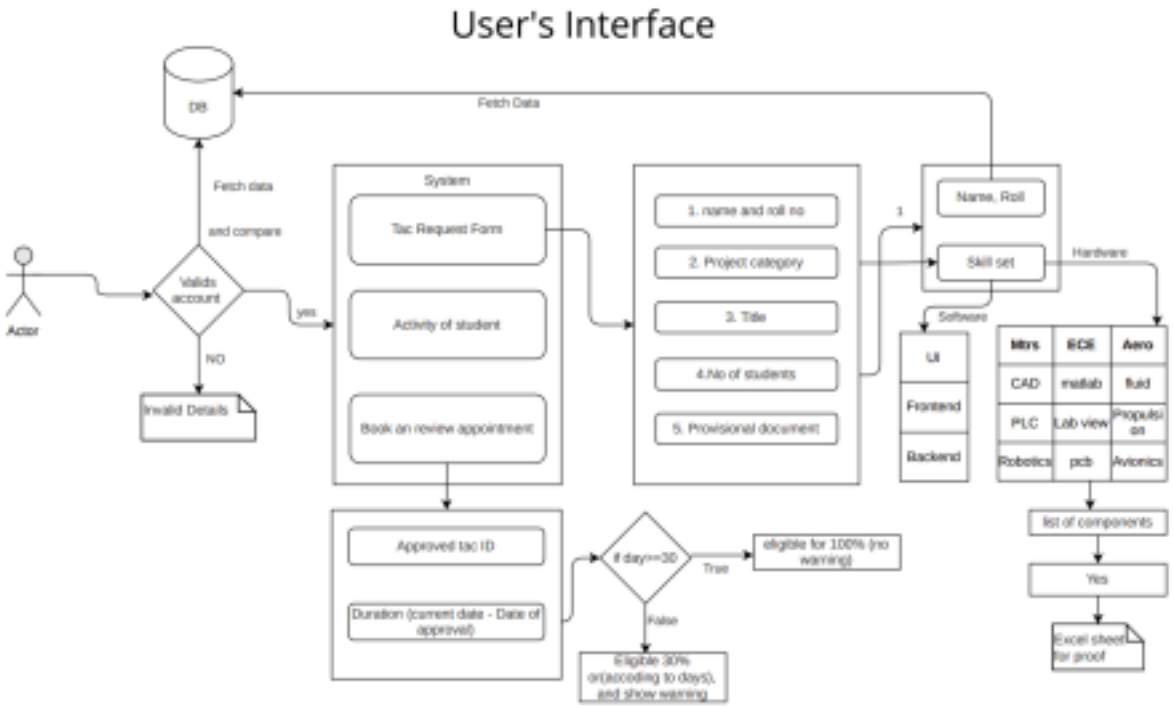
Student with approved TAC ID can request for Project review after completion of 30 days

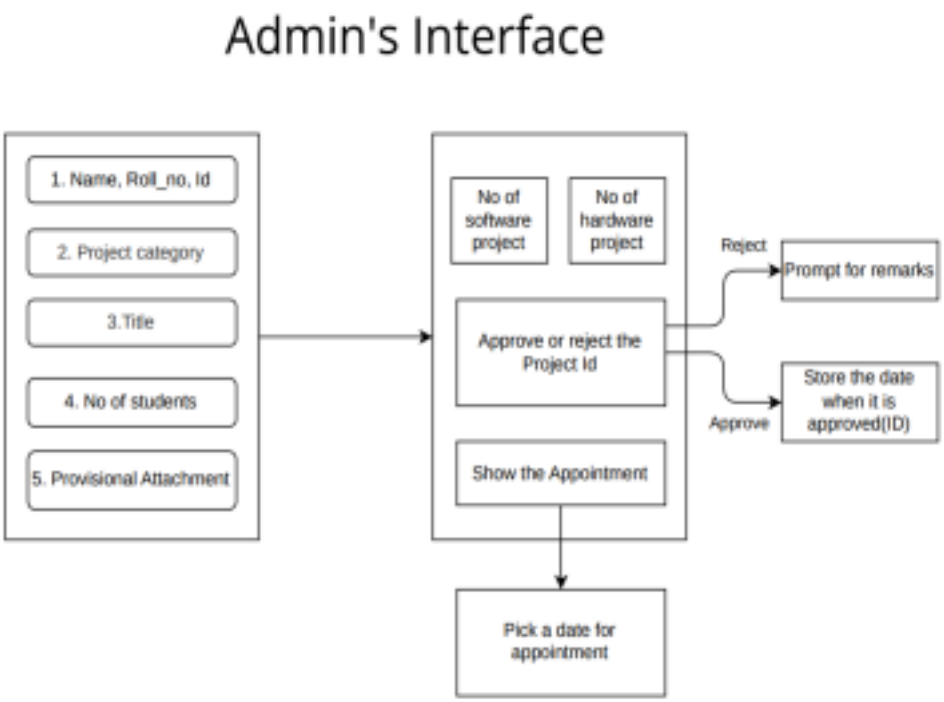
5. Admin Access:

Admin can view all submitted TAC applications in a category of either software or hardware, view application details, approve or reject the application with suitable remarks, schedule meetings.

6. Admin's Analytical Dashboard:

Admin can view the number of applications by category, appointments request and also see the latest log of applications





3. System Requirements Specification:

3.1 Functional Requirements:

- **User Management:**
  - Students can register and login.
  - Admins have access control with an analytical dashboard and dedicated features .
- **TAC Application:**
  - Students can submit applications with appropriate details
  - Application form contains:
    - Title of Project
    - Category of the project
    - Number of students involved
    - Provisional document attachment
- **Application Status:**
  - Students can view the current status of their application
  - If the application is rejected then the remarks is shown

    - Students can also see the logs of their applications
- **Appointment Scheduling (After Approval):**
    - Students with approved TACs can request appointments after completion of 30 days
- **Admin Dashboard:**
    - Admins can view a list of all submitted TAC applications.
    - Applications can be filtered by category (software, hardware).
    - Admins can view details of each application.
    - Admins can approve or reject applications with suitable remarks.
    - Admins can schedule meetings for accepted appointments.
- **Analytics Dashboard:**
    - Admin can view the number of applications by its category
    - Number of appointments is requested based on the category

3.2. Non-Functional Requirements:

- **Performance**: The system must respond to user actions within 2 seconds to ensure efficient usability and must handle a concurrent user load of at least 100 users without significant performance degradation.
- **Security**: User data must be encrypted during transmission and storage, and access to sensitive functionalities should be restricted to authorized admin users through secure authentication mechanisms.
- **Usability**: The user interface should be intuitive and user-friendly, with clear and concise error messages provided to guide users in case of input errors or system failures.
- **Reliability**: The system should be available 24/7 with minimal downtime and should have a backup and recovery mechanism in place to prevent data loss in case of system failures or crashes.

- **Scalability**: The system should be designed to accommodate an increasing number of users and data volume over time, and it should be scalable to support additional features and functionalities as per future requirements.

Backend:

1. Student entity

name	String
email	String
password	Hash code
Roll no	String

2. Tac Details Entity

Roll no	String
Details	Array of Objects category String (drop down) title String numberOfStudents Number additionalStudents Array of Objects pdfPath String (using multer) status String createdAt Date

3. Appointment Entity

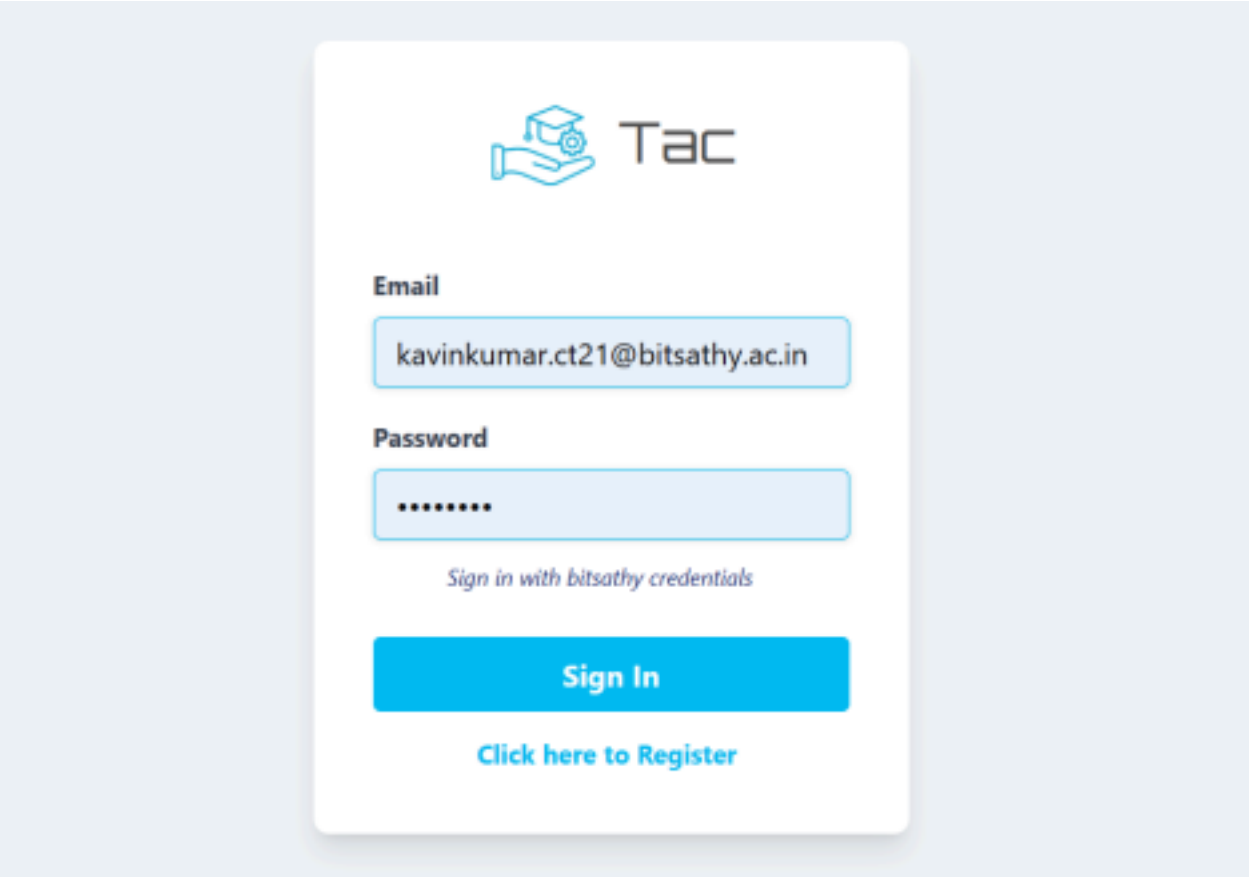
Roll no	String
Tac_appointment	Details[ ] in the Tac entity

Stack:

Front End	Vue Js, Tailwind css
Backend	Node Js, Express
Data Base	MongoDB

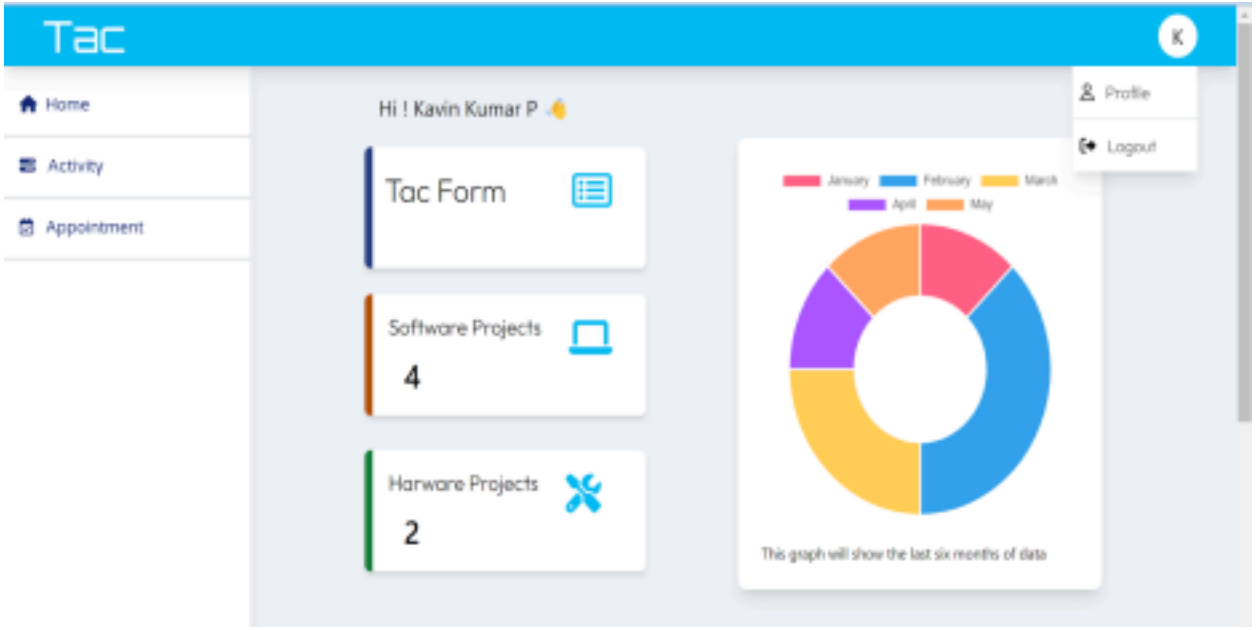
Prototype of the Project:

1. Login form

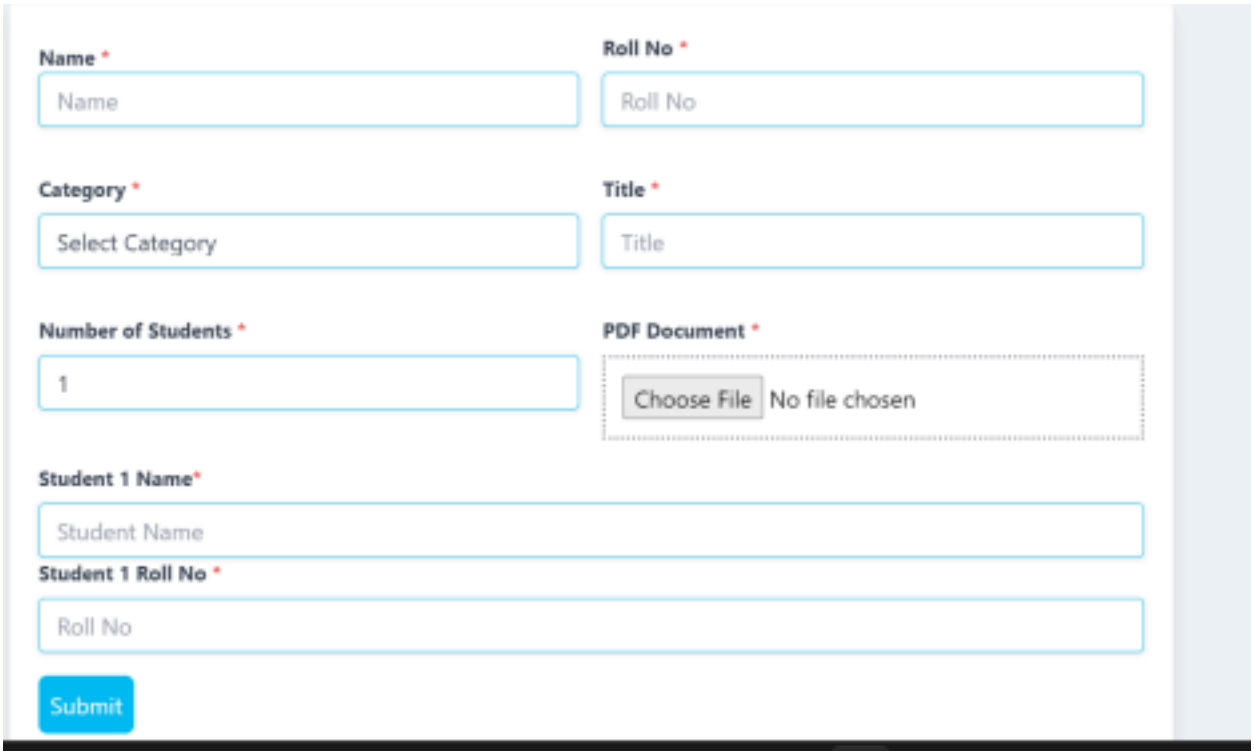


2. Register Form

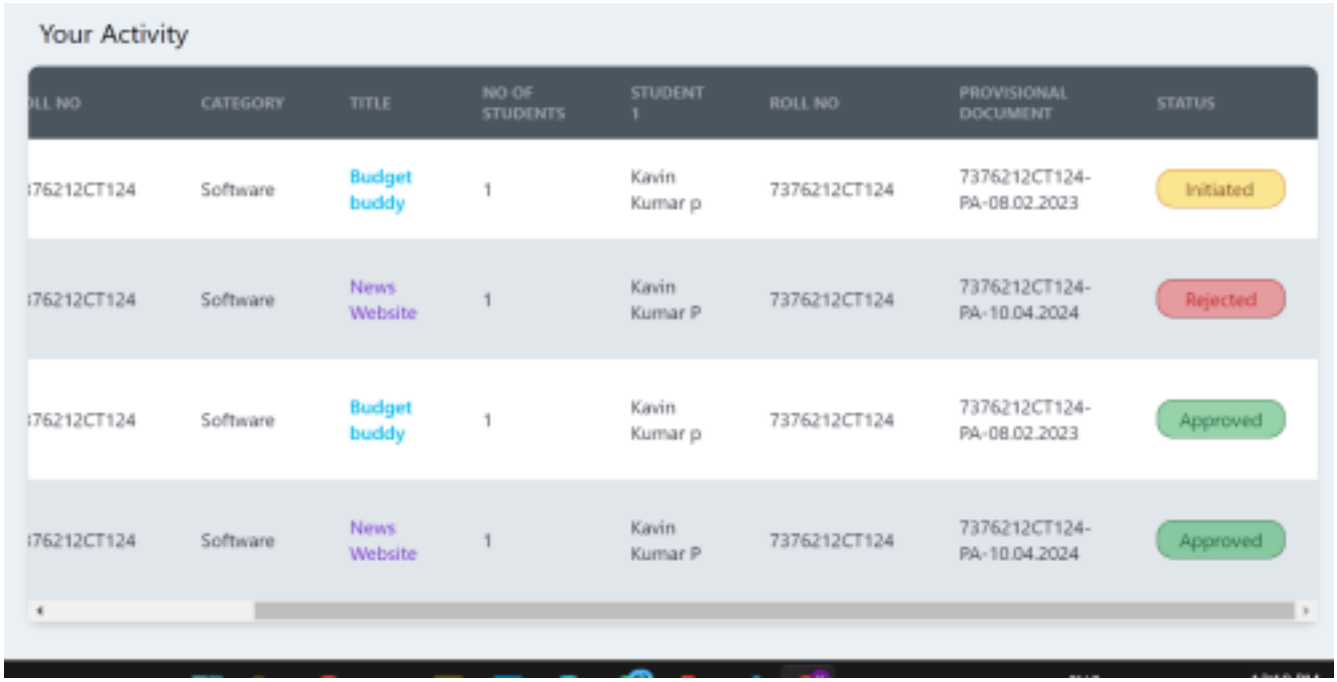
3. Student's view



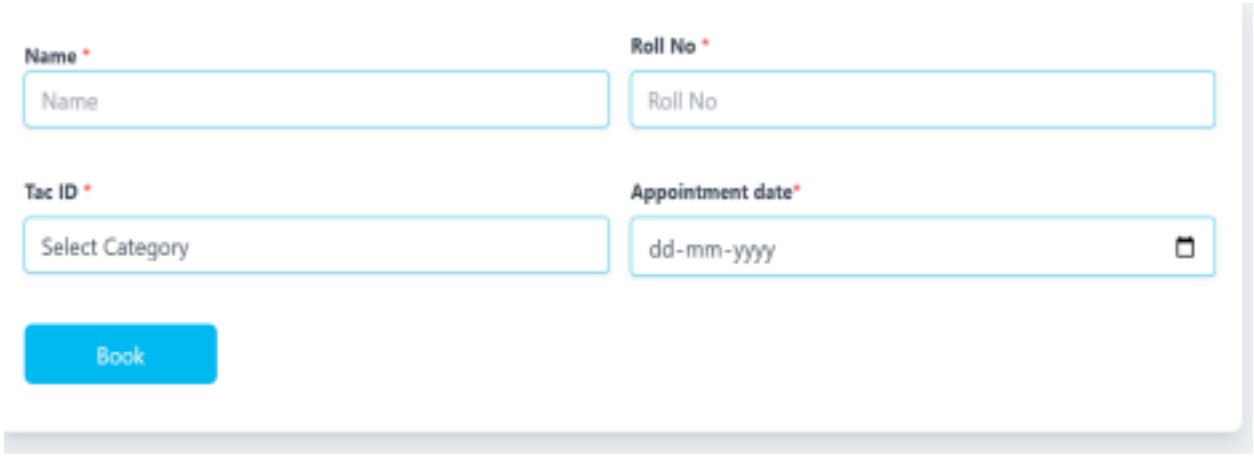
4. Tac form:



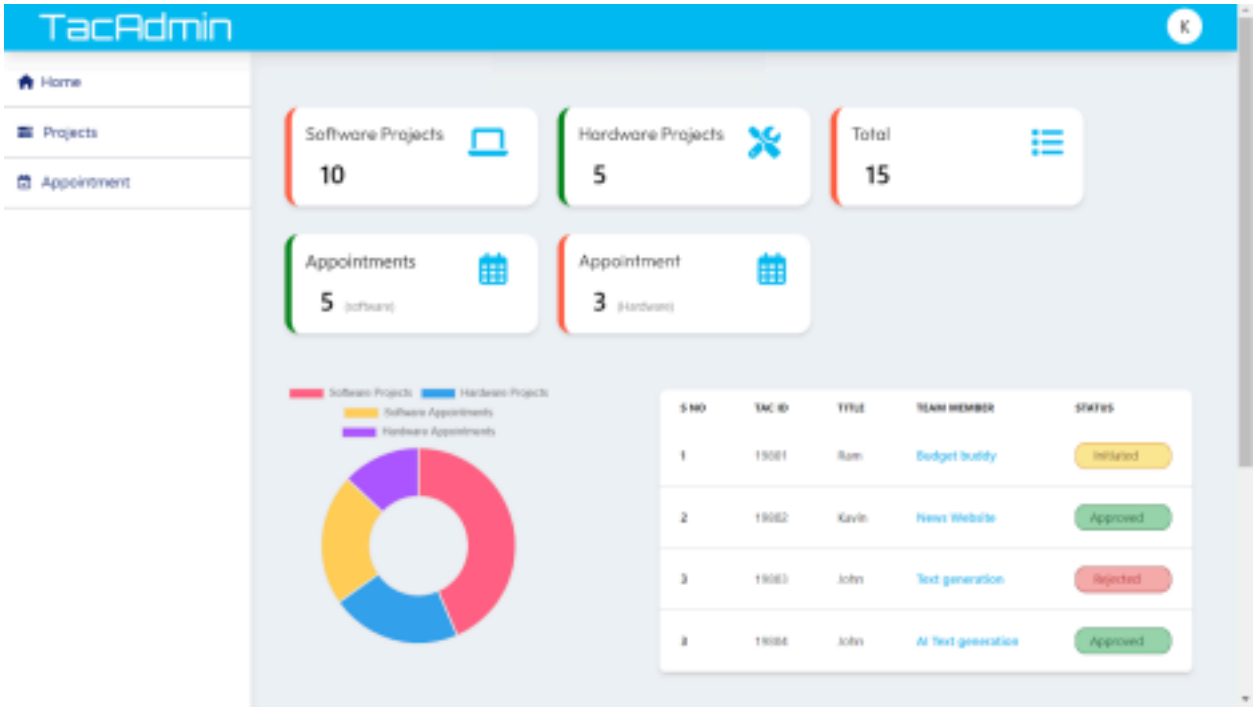
5. Activity page:



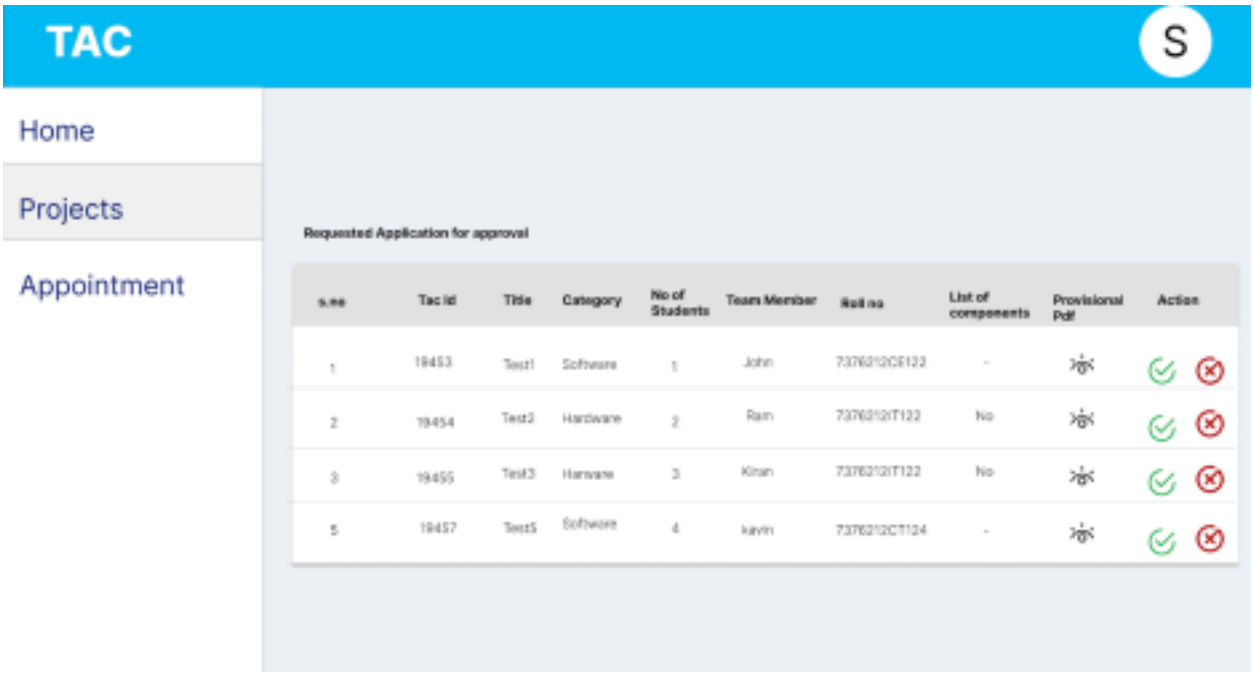
6. Appointment Page:



7. Admin's View:



8. Approval Page



9. Conform the date