SELF INTENSIVE TRAINING ON FULL STACK DEVELOPMENT

Stage -1

(Planning And Requirement Gathering)

Name	ARUN K N
Roll No	7376221EC117
Seat No	395
Project ID	35
Module Name	Open Innovation Category.

TECHNICAL COMPONENTS:

COMPONENT	**MERN STACK**
Backend	React (JS Library for building user interfaces)
Frontend	Node.js with Express.js
Database	MongoDB (NOSQL Database)
API	Open API

1. Introduction

1.1. Purpose:

The purpose of this document is to present a detailed description of the Open Innovation Category. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli.

1.2. Scope of Project:

- This software system will serve as a portal for the Campus (BIT), enabling students to submit their projects and receive their result (Approved or rejected). From an administrative perspective, this system will provide a comprehensive analytical dashboard for project oversight from the students.

- Administrators (Faculty) have the ability to approve or reject projects. Once a project is approved, students can schedule to their project work.

2.System Overview

2.1. Users:

Students: They have the ability to submit applications for their idea, upload relevant project details, like team size, domain, title, abstract, explanation of the project, flow diagram, Er diagram, components for the project.

Admins: Review submitted details, approve or reject applications (with remarks), manage appointments, schedule meetings, and access analytical dashboards for project.

2.2. Features:

1. Login and registration: Students can register for an account or login with their existing account.

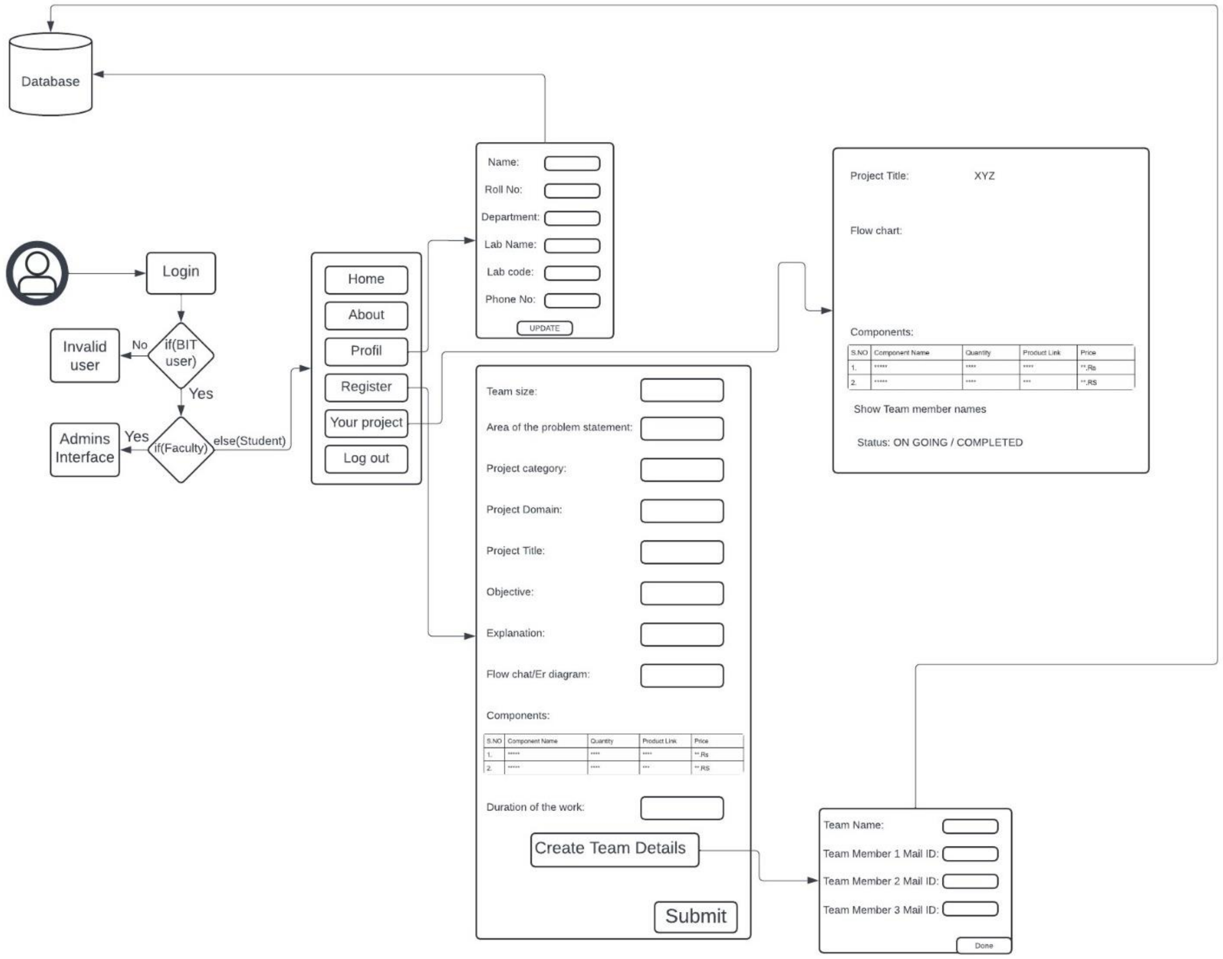
2. Application Submission: Students can input relevant details regarding their project application including team size, team members details, project title, description, objectives, and any necessary attachments. Upon completion, the application is submitted to the admin interface for review and further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

3. Application Status: Students can view the current status of their application and also see the history logs Activity.

4. 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.

5. Admin's Analytical Dashboard: Admin can view the number of applications by category, appointments requested and also see the latest log of applications.

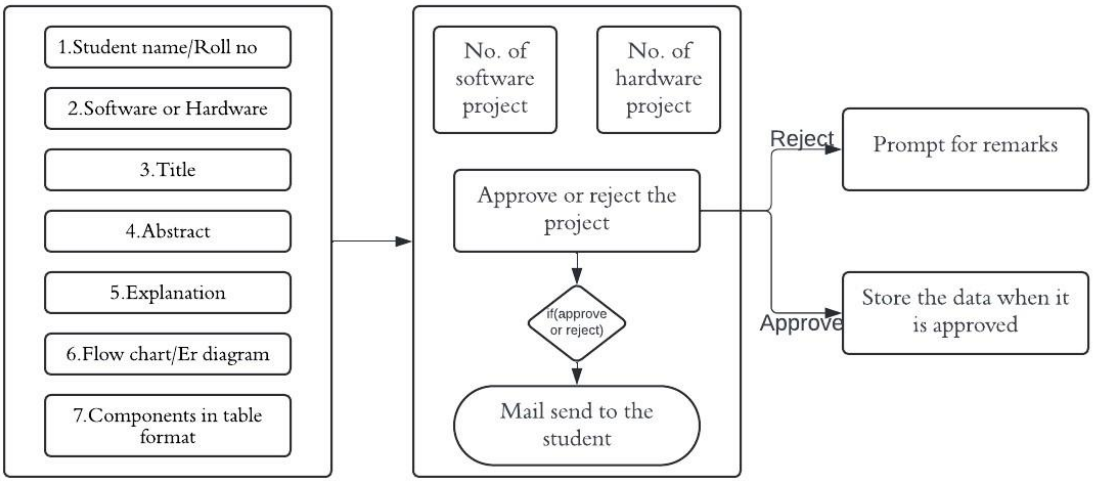
Flow Chart: User interface



S.NO	Component Name	Quantity	Product Link	Price
1.	*****	****	****	**.Rs
2.	*****	****	***	**.RS

S.NO	Component Name	Quantity	Product Link	Price
1.	*****	****	****	**.Rs
2.	*****	****	***	**.RS

Admin’s interface



3. System Requirements Specification:

3.1 Functional Requirements:

- **User Management:**
  - Students can directly login through the google.
  - Admins have access control with an analytical dashboard and dedicated features.
- **Registration Application:**
  - Students can submit applications with appropriate details
  - Application form contains:
    - Area of the problem statement.
    - Project category.
    - Project domain.
    - Project title.
    - Objective.
    - Explanation.
    - Flow chart / Er diagram.
    - Components.
    - Time plane.
- **Application Status:**
  - Students can view the current status of their application in the "Your Project" Page, and also students can see the Project title, flow chat and components list.
  - If the application is rejected then the remarks is shown.
- **Admin Dashboard:**
  - Admins can view a list of all submitted Projects.
  - Applications can be filtered by category (software, hardware).
  - Admins can view details of each application.
  - Admins can approve or reject applications with suitable remarks.

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.