Software Requirement Specification for TAC Portal

Name	PRENEETHA A
Roll no	7376221CS266
Seat no	13
Project ID	7
Problem Statement	Task dashboard

Problem statement:

Admin and user dashboard for task portal to view overdue, active and completed task

1. Introduction:

1.1. Purpose:

The purpose of this document is to present a detailed description of the Tac dashboard. 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 operated.

1.2. Scope of Project:

- The scope of a task dashboard for workforce includes supplying a centralized, actual-time evaluate of all tasks and obligations assigned to the staffs by Hod's. Key functionalities consist of undertaking completed, active, overdue tracking. The dashboard have to facilitate efficient assignment management by using imparting capabilities including development tracking , updates, and sending reminders.
- The purpose is to decorate transparency, responsibility, and performance, ensuring that every one team contributors are aligned and on the right track to meet organizational targets.

2. Project-flow:

2.1 Dependencies:

- Integration with Google OAuth for user authentication.

- Consistent performance and availability of the existing email server.

2.2 Business Context:

The centralized task dashboard system is aimed at enhancing communication clarity and timeliness across BIT, thus boosting organizational efficiency by minimizing scheduling conflicts. Primary stakeholders include HOD’s, faculty and administrative staff

2.3 User personas:

- **User Staff**: To view the tasks and their completed and overdue status.
- **Admin :** Assign and schedule the tasks,Manages tasks of respective staffs.
- **Task**: An activity or piece of work that needs to be completed.
- **Overdue Task**: A task that has not been completed by its due date.
- **Active Task**: A task that is currently in progress and not yet completed.
- **Completed Task**: A task that has been finished.

2.4 Functional Requirements:

- **User Authentication**: Secure login using Google OAuth.
- **Admin**: The ability to assign and monitor the tasks to the users.
- **User:**The ability to view the tasks and their completed tasks.

2.5 Hardware Interfaces

- Not applicable as the system is web-based.

2.6 Software Interfaces

- The system will interact with a backend API to manage data.

- It will use a database to store task and user information.

2.7 Communications Interfaces

- The system will use HTTPS for secure communication between the client and server.

3. Technical Components: Tech Stack - LAMP

Front End -

- HTML
- CSS
- javascript

Back End -

- Linux
- Apache Web Server
- PHP with Laravel framework

Database -

- MYSQL

API -

- REST Ful API

4. Non-Functional Requirements:

4.1. Performance Requirements

- The system should support up to 1000 concurrent users without performance degradation.
- Page load times should not exceed 2 seconds under normal conditions.

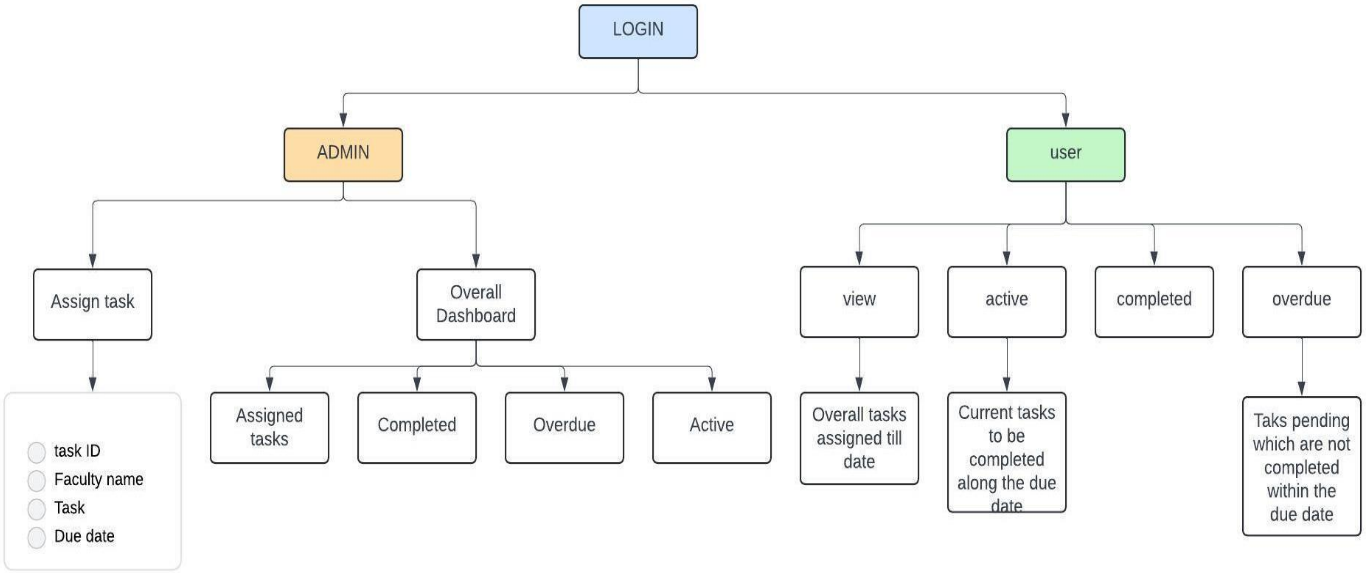
4.2. Security Requirements

- User data must be encrypted in transit and at rest.
- Role-based access control must be implemented to ensure appropriate data access.

4.3. Usability Requirements

- The interface should be intuitive and easy to navigate.
- The system should provide help and documentation for user assistance.

5. FLOWCHART:


LOGIN
ADMIN
user
Assign task
Overall Dashboard
view
active
completed
overdue
task ID
Faculty name
Task
Due date
Assigned tasks
Completed
Overdue
Active
Overall tasks assigned till date
Current tasks to be completed along the due date
Taks pending which are not completed within the due date