SOFTWARE REQUIREMENT SPECIFICATION FOR SKILL TRAINING PORTAL

NAME	LOHITHA B E
ROLL NO	7376222AL163
SEAT NO	6
DEPARTMENT	AIML
PROJECT ID	6
PROBLEM STATEMENT	REPEAT TASK

INTRODUCTION

Purpose

The purpose of the Repeat Task project is to develop a scheduling system that allows the creation and management of recurring tasks, particularly for allocating classes to staff. It ensures that class schedules do not clash and each staff member has a unique, conflict-free timetable. The system provides a user-friendly interface for staff to view their schedules and manage their tasks. Additionally, it offers administrative tools for managing class allocations efficiently.

Scope of the Project

The scope of the Repeat Task project encompasses developing a comprehensive scheduling system to manage recurring tasks, specifically focusing on class allocations for educational institutions. The system includes user authentication for staff and administrators, class creation and management, and conflict-free task allocation. Staff members can view their schedules through a user-friendly dashboard, while administrators can efficiently manage class schedules and allocations. Optional features like notification systems for reminders may be incorporated. The project aims to ensure scalability, security, and usability, providing a reliable tool for streamlining class scheduling and enhancing organizational efficiency.

1.Create a system for repeat tasks:

Tasks can repeat daily, weekly, or monthly.

2.Allocate classes to staff:

Ensure no scheduling conflicts or duplicates.

3.User login:

Staff members can log in to view their class schedules.

4.Class management:

Admins can manage class schedules and allocations.

Functional Requirements:

User Management:

Staff can register and login to the system.

Staff can view and update their profile information.

Task Management:

Tasks should not clash with existing tasks of the staff or other staff members in the same department.

Class Allocation:

Admins can allocate classes to staff members.

Each department and staff member should have unique classes without duplicates or clashes.

Class Details:

Staff can view details of their allocated classes upon login.

Details should include class name, date, time, location, and any additional notes.

Non-functional Requirements:

Performance:

The system should handle concurrent user requests efficiently.

Response time for task creation and class allocation should be minimal.

Security:

User authentication and data transmission should be encrypted.

Role-based access control to ensure staff only access their own data.

Scalability:

The system should be scalable to accommodate a growing number of staff members and tasks.

Database design should allow for easy scaling.

Usability:

The user interface should be intuitive and easy to navigate. Error messages should be informative and user-friendly.

User Dashboard (Functionalities):

Login:

Staff can register or login with their credentials.

Profile Management:

Staff can view and update their profile information.

Class Allocation:

View allocated classes with details.

Notifications:

Receive notifications for upcoming tasks or class changes.

User Dashboard (Components):

Dashboard:

Overview of upcoming tasks and classes.

Profile Page:

View and edit personal information.

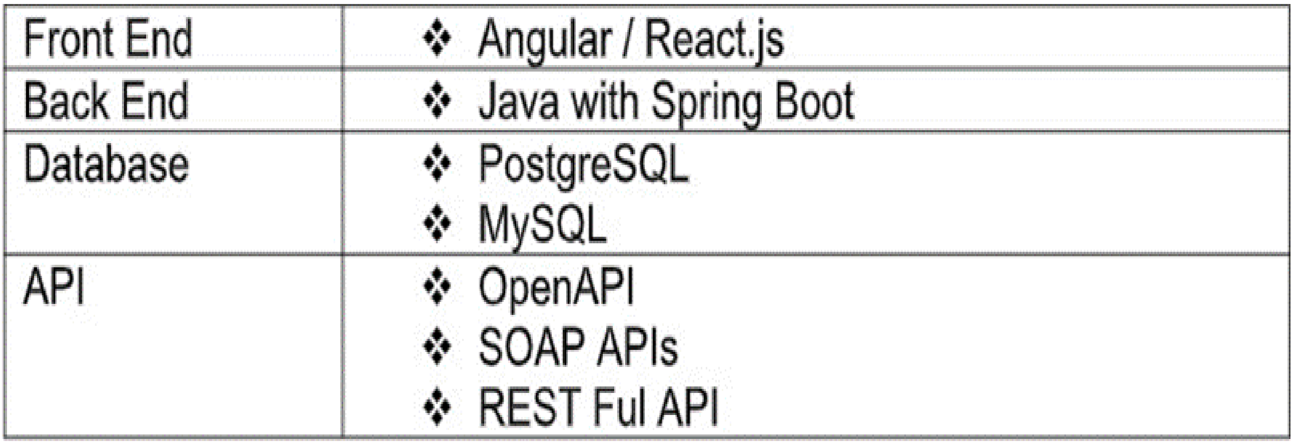
Class Details Page:

View details of allocated classes.

Notification Center:

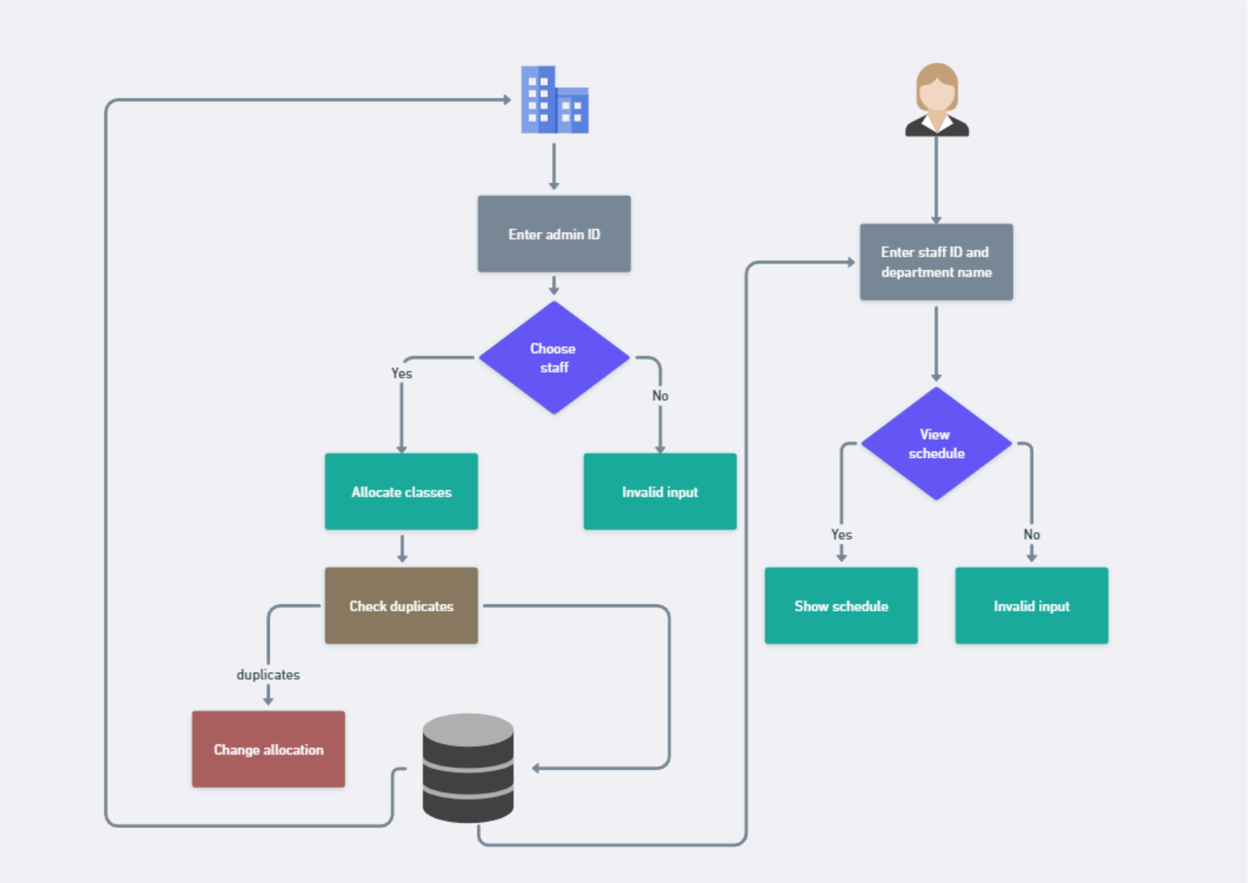
Manage notifications settings and view recent notifications.

Technology Stack:



Front End	❖ Angular / React.js
Back End	❖ Java with Spring Boot
Database	❖ PostgreSQL ❖ MySQL
API	❖ OpenAPI ❖ SOAP APIs ❖ REST Ful API

FLOWCHART


Enter admin ID
Choose staff
Yes
No
Allocate classes
Invalid input
Check duplicates
duplicates
Change allocation
Enter staff ID and department name
View schedule
Yes
No
Show schedule
Invalid input