

## SOFTWARE REQUIREMENT SPECIFICATION FOR REPEAT TASKS

NAME	MAHATHI A
ROLL NO	7376221CD123
PROJECT ID	6
PROBLEM STSTATEMENT	REPEAT TASKS

### INTRODUCTION

The repeat task system aims to streamline task management by offering users a reliable platform for scheduling and monitoring recurring tasks. This system will be equipped with interactive dashboards that provide real-time insights and controls for both end-users and administrators. Users can set up and manage their recurring tasks with ease, while administrators have the tools to oversee user activities, ensure compliance with task schedules, and configure system-wide settings.

### SCOPE OF THE PROJECT

1. **User Task Management:** Develop features for users to create, edit, and delete recurring tasks with customizable frequencies (daily, weekly, monthly, etc.).
2. **User Dashboard:** Design an interactive dashboard for users to view and manage their tasks, including notifications for upcoming tasks.
3. **Admin Dashboard:** Create an admin interface to oversee all user activities, manage user accounts, and configure global settings.
4. **Authentication and Authorization:** Implement a secure login system with role-based access control to differentiate between regular users and administrators.
5. **Database Integration:** Use MongoDB to store user data, task details, and system configurations, ensuring data integrity and scalability.
6. **API Development:** Build RESTful APIs using Express.js to handle task operations, user management, and notification services.
7. **Real-Time Notifications:** Implement real-time notification functionality to alert users about their tasks through email and in-app messages.
8. **Responsive Design:** Ensure the front-end, built with Angular, is responsive and user-friendly across various devices and screen sizes.

9. **Performance Optimization:** Optimize the application for performance and scalability to handle a large number of concurrent users.
10. **Reporting and Analytics:** Provide analytics and reporting tools for administrators to monitor task completion rates, user engagement, and system performance.

## **SYSTEM OVERVIEW**

### **1. FACULTY DASHBOARD**

- **Faculty Dashboard:** Provides faculty members with an interface to create, view, edit, and delete tasks. Displays upcoming tasks, task statuses, and notifications. Key features include:
- **Task Creation and Management:** Faculty can create tasks with specific details and set recurrence patterns.
- **Task Overview:** A comprehensive view of all tasks, including due dates and statuses.
- **Notifications:** Real-time alerts for upcoming tasks and deadlines.

### **2. HOD DASHBOARD**

- **HOD Dashboard:** Offers HODs tools to manage faculty members, view system analytics, and oversee task management. Key features include:
- **User Management:** HODs can add, remove, and update faculty accounts.
- **Task Oversight:** Monitor all tasks created by faculty and ensure compliance with schedules.
- **Analytics and Reporting:** Generate reports on task completion rates, faculty activity, and overall system performance

## **FEATURES OF THE PROJECT**

### **Faculty (Users)**

1. **Task Creation and Management:**
  - **Create Tasks:** Faculty can create new tasks, specifying details such as task name, description, priority, and due dates.
  - **Recurrence Settings:** Set tasks to recur daily, weekly, monthly, or at custom intervals.
  - **Edit and Delete Tasks:** Modify existing tasks or remove them as needed.
2. **Task Overview:**
  - **Dashboard View:** A comprehensive view of all tasks, including upcoming tasks, overdue tasks, and completed tasks.
  - **Calendar Integration:** Visualize tasks on a calendar to easily manage and track task deadlines.
  - **Task Filtering:** Filter tasks by status, priority, and due date.
3. **Notifications and Reminders:**
  - **Real-Time Notifications:** Receive alerts for upcoming tasks and deadlines through in-app notifications and email reminders.
  - **Custom Reminder Settings:** Faculty can set custom reminders for tasks to receive notifications at preferred times.

4. **User Profile Management:**
  - **Profile Settings:** Update personal information, change passwords, and manage notification preferences.

## **HOD (Admin)**

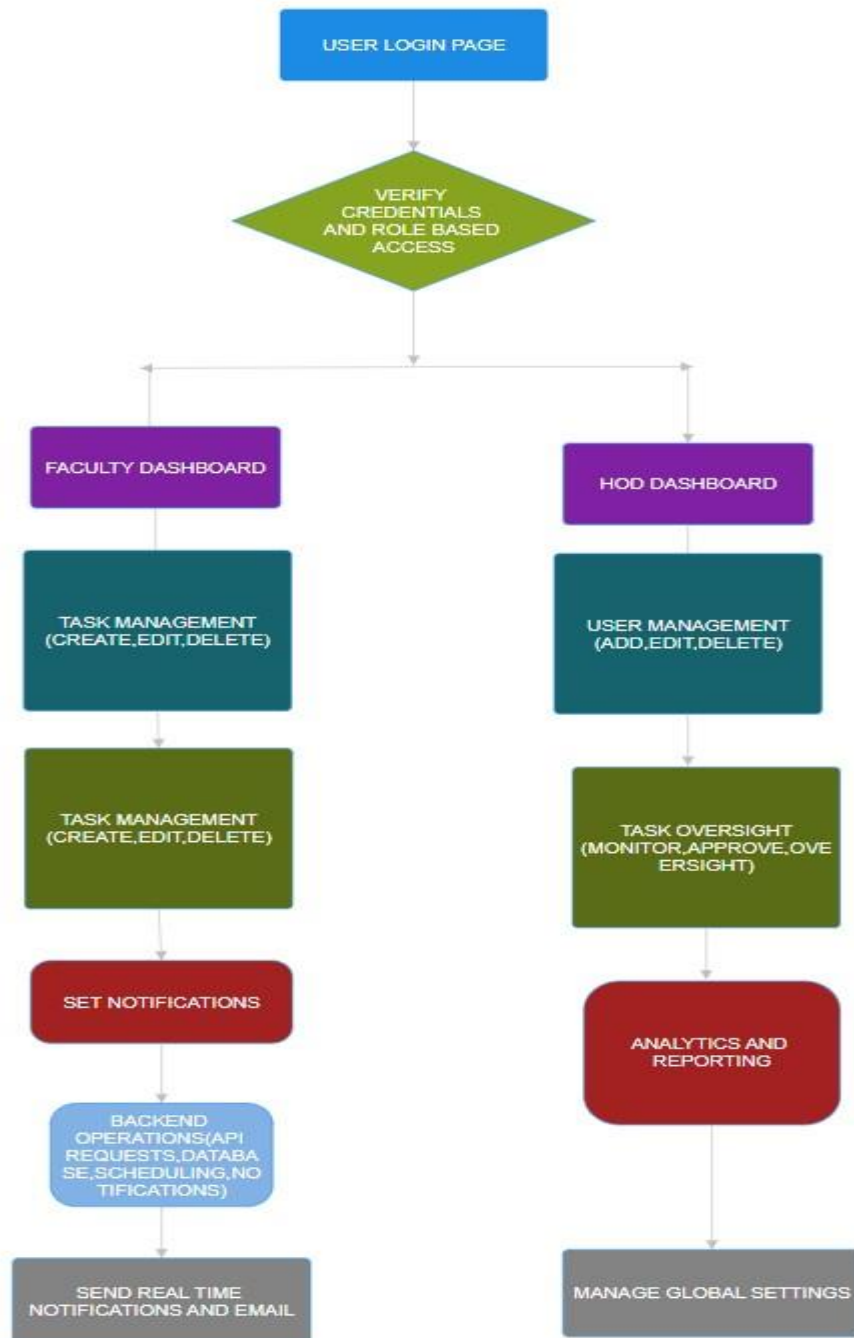
1. **User Management:**
  - **Manage Faculty Accounts:** Add new faculty members, update existing profiles, and deactivate or delete accounts.
  - **Role Assignment:** Assign and manage roles to ensure appropriate access levels for faculty members.
2. **Task Oversight:**
  - **Monitor Tasks:** View and oversee all tasks created by faculty members to ensure compliance with schedules.
  - **Approve or Modify Tasks:** Approve critical tasks or make necessary adjustments to task details and deadlines.
3. **Analytics and Reporting:**
  - **Task Completion Rates:** Generate reports on task completion rates for individual faculty members and the department as a whole.
  - **Activity Logs:** View logs of faculty activities, including task creation, modification, and completion.
  - **Performance Metrics:** Analyse data to identify trends, bottlenecks, and areas for improvement.
4. **System Configuration:**
  - **Global Settings:** Configure system-wide settings, including notification templates, recurrence patterns, and default task priorities.
  - **Access Control:** Define and manage access permissions to ensure data security and privacy.
5. **Real-Time Notifications:**
  - **Admin Alerts:** Receive notifications about critical tasks, system updates, and important faculty activities.
  - **System Health Monitoring:** Monitor the health and performance of the system to ensure smooth operation.

## **General Features**

1. **Authentication and Authorization:**
  - **Secure Login:** Ensure secure access with username and password authentication through college email id.
  - **Role-Based Access Control:** Implement different access levels for faculty and HODs to protect sensitive information and functionalities.
2. **Responsive Design:**
  - **Cross-Device Compatibility:** Ensure the application is responsive and user-friendly across various devices and screen sizes.
  - **Intuitive User Interface:** Design a clean and easy-to-navigate interface for both faculty and HODs.
3. **Database Management:**
  - **Efficient Data Storage:** Use MongoDB to store user data, task details, and system configurations efficiently.

- **Data Integrity and Backup:** Ensure data integrity and implement regular backups to prevent data loss.
4. **Performance Optimization:**
- **Scalability:** Design the system to handle a large number of concurrent users and tasks.
  - **Fast Response Times:** Optimize backend processes to ensure quick and reliable performance.

## PROJECT FLOW



## **FUNCTIONAL REQUIREMENTS**

- **USER:**
  1. User Authentication
  2. Task Management
  3. Task Overview
  4. Notifications and Reminders
  5. Profile Management
- **ADMIN:**
  1. Admin Authentication
  2. User Management
  3. Task Oversight
  4. Analytics and Reporting
  5. System Configuration
- **GENERAL**
  1. Authentication and Authorization
  2. API Development
  3. Database Management
  4. Real-Time Notifications
  5. Responsive Design
  6. Performance Optimization
  7. Security

## **NON FUNCTIONAL REQUIREMENT**

1. Performance
2. Reliability
3. Usability
4. Security
5. Maintainability
6. Portability
7. Backup and Recovery
8. Compliance
9. Performance Monitoring

## **STACK**

Frontend	Angular (JavaScript framework)
Backend	Node js & Express js
Database	Mongo DB