BANNARI AMMAN INSTITUTE OF TECHNOLOGY

An Autonomous Institution Affiliated to Anna University - Chennai, Accredited by NAAC with A+ Grade

Sathyamangalam - 638401 Erode District, Tamil Nadu, India

Stay Ahead

Student Name: MITHUNAVANAN S

Seat No: 442

Project ID: 6

Project title: Repeat Task

Technical Components

Component	Tech Stack
Backend	Node.js with Express.js
Frontend	React
Database	MongoDB
API	RESTful API

Implementation Timeline

Phase	Deadline	Status	Notes
Stage 1	03/06/2024	Under review	Planning and Requirement gathering
Stage 2		Under review	Design and Prototyping
Stage 3		Under review	DB Designing
Stage 4		In progress	Backend Implementation
Stage 5		Not started	Testing & Implementation

PROBLEM STATEMENT:

The advent of web-based applications has transformed the way individuals manage their tasks and schedules. However, existing task management applications often lack the capability to handle repetitive tasks efficiently. This project aims to address this limitation by developing a web application using the MERN (MongoDB, Express, React, Node.js) stack that enables users to create tasks with repetition intervals, ensuring efficient task management and scheduling.

1.User Authentication: Many task management applications lack robust user authentication mechanisms, leading to security vulnerabilities and unauthorized access. This project will implement user registration, login, and account management functionalities to ensure secure access to the application.

2.Task Creation: Existing task management applications often provide limited options for task creation, failing to accommodate repetitive tasks effectively. This project will empower users to create tasks with specific details, such as task name, description, start date, and repetition intervals (daily, weekly, monthly).

3.Task Scheduling: Managing repetitive tasks manually can be time-consuming and prone to errors. This project will automate the task scheduling process, allowing users to schedule tasks to repeat at specified intervals. By leveraging backend scheduling mechanisms, tasks will be executed automatically, enhancing productivity and efficiency.

4.Notifications: Users often miss important tasks and deadlines due to a lack of timely reminders. This project will integrate notification functionalities to alert users about upcoming tasks based on their schedule. By providing notifications via email or in-app alerts, users will stay informed and proactive in managing their tasks effectively.

PROJECT-FLOW:

Purpose:

To develop a web application that enables users to create, manage, and receive notifications for tasks that repeat at specified intervals (daily, weekly, monthly), leveraging the capabilities of the MERN stack to ensure robust performance and a seamless user experience.

Scope:

The project will cover the following functionalities:

- User registration, login, and authentication.
- Task creation with specific details and repetition intervals.
- Automated task scheduling and execution.
- Notification system to alert users of upcoming tasks.
- User interface for managing tasks.

Business Context:

Effective task management is crucial for productivity and efficiency in both personal and professional settings. Current task management solutions often lack robust support for repetitive tasks, leading to missed deadlines and decreased productivity. This application aims to fill this gap by providing a comprehensive tool for scheduling and managing repeating tasks.

Consideration:

Effective task management is crucial for productivity and efficiency in both personal and professional settings. Current task management solutions often lack robust support for repetitive tasks, leading to missed deadlines and decreased productivity. This application aims to fill this gap by providing a comprehensive tool for scheduling and managing repeating tasks.

Dependencies:

- Security: Ensuring secure user authentication and data protection.
- Scalability: Designing the system to handle a growing number of users and tasks.
- Usability: Creating an intuitive user interface for easy task management.
- Performance: Ensuring the application performs efficiently, even with frequent task scheduling.

User personas:

- **Student:** Needs an up-to-date schedule to effectively plan activities.
- **Faculty:** Requires the ability to send out schedule updates and notices efficiently.
- **Admin Staff:** Manages system operations, resolves conflicts, and approves mail requests.

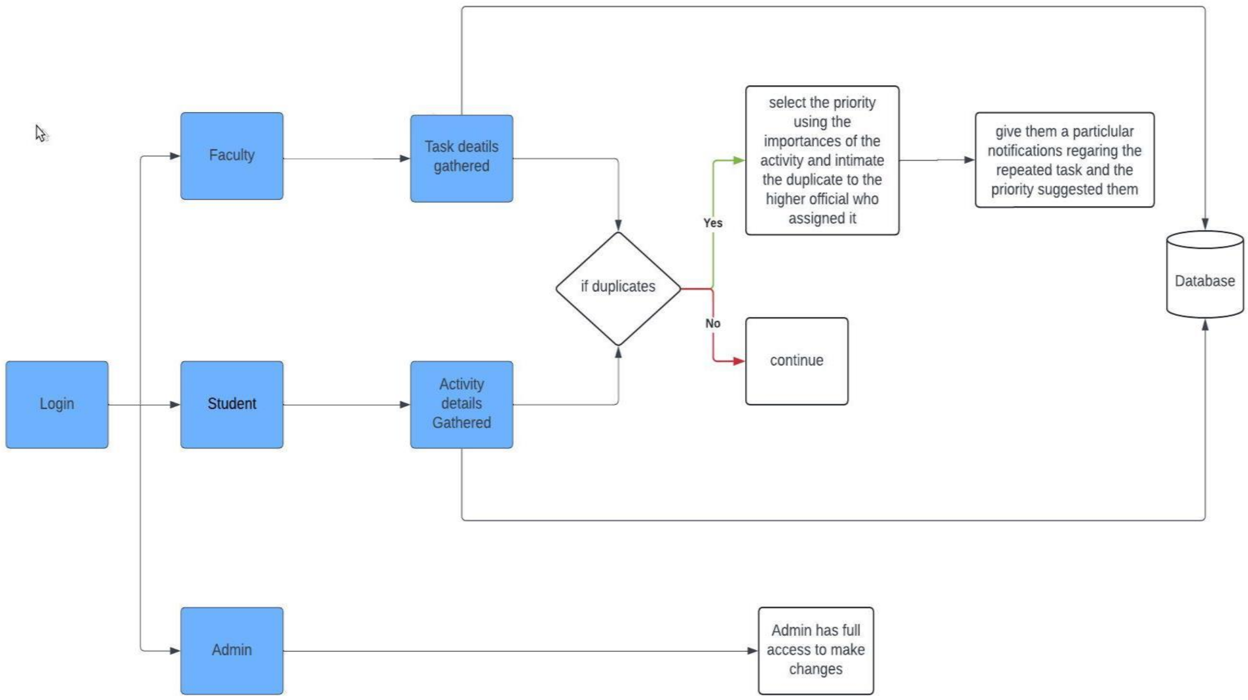
User Stories:

- As a student, I want to view a unified schedule of my classes and events to organize my day effectively.
- As a faculty member, I need to ensure my communications reach students without conflicting with their other scheduled activities.

Functional Requirements:

- **User Authentication:** Secure login using Google OAuth.
- **Mailer Request Form:** Users input mail content, scheduling time, category, and recipients.
- **Conflict Resolution:** Automatic detection of scheduling conflicts with options for adjustment.
- **Dynamic Dashboard:** Real-time schedule viewing and interaction.
- **Priority Algorithm:** Automated prioritization of communications based on rules.

FLOW CHART:


Faculty
Task deatils gathered
select the priority using the importances of the activity and intimate the duplicate to the higher official who assigned it
give them a particular notifications regaring the repeated task and the priority suggested them
Yes
if duplicates
No
continue
Database
Login
Student
Activity details Gathered
Admin
Admin has full access to make changes