COLLEGE CODE : 9623

COLLEGE NAME : Amrita College of Engineering and Technology

DEPARTMENT : Computer Science and Engineering

STUDENT NM-ID : BD135BD8FDA28CFB7D24092B09DCA85C

ROLL NO : 962323104035

DATE : 12/09/2025

Completed the project named as

Phase 1 TECHNOLOGY PROJECT

NAME : CHAT APPLICATION UI

SUBMITTED BY,
NAME : Hari Krishnan R
MOBILE NO : 9385610623

PROJECT PLAN

Phase 1 – Problem Understanding & Requirements

1.1 Problem Statement

In today's fast-paced environment, effec ve task management is crucial for collabora on and produc vity. Many student organiza ons, small teams, and clubs s ll rely on informal methods such as WhatsApp messages, spreadsheets, or verbal communica on to manage tasks. While these methods are convenient, they lack structure, accountability, and transparency.

For example, when mul ple people work on a college event, it becomes difficult to track who is responsible for what task, whether the task is completed, and if deadlines are being met. As a result, delays occur, tasks are forgo en, and miscommunica on increases.

To address this problem, we propose the development of a Task Management Web Applica on built using a React.js frontend and a Node.js REST API backend. The applica on will provide features such as user authen ca on, task crea on and assignment, progress tracking, and dashboards for be er visibility. This solu on will streamline collabora on, improve accountability, and ensure mely task comple on.

1.2 Users & Stakeholders

Primary Users

1. Regular Users (Members) – They are responsible for receiving assigned tasks and upda ng their status.
2. Task Creators – Individuals who create and assign tasks to themselves or others.
3. Administrators/Moderators – Users with addi onal privileges to update or delete any task and oversee project progress.

Stakeholders

1. Club Leaders / Project Managers – Need insights into progress and accountability.
2. Team Members – Want a simple, distrac on-free way to see their responsibili es.
3. Faculty/External Mentors – May require access to task summaries and progress updates.

Each stakeholder has different levels of involvement. For instance, a regular team member only interacts with their assigned tasks, while an administrator has control over the en re workflow.

1.3 User Stories

The applica on will follow the Agile methodology of defining user stories. These represent features from the perspec ve of end-users:

1. As a member, I want to log in securely so that I can access my tasks.
2. As a member, I want to view my assigned tasks so that I don't forget deadlines.
3. As a member, I want to update the status of my tasks so that others can see my progress.
4. As a task creator, I want to assign tasks to specific users so that responsibili es are clear.
5. As an admin, I want to update or delete tasks so that the system stays clean and accurate.
6. As an admin, I want to view the dashboard of all tasks so I can monitor overall progress.
7. As a user, I want to receive deadline reminders so I don't miss important work.
8. As a user, I want the system to be mobile-friendly so that I can use it on the go.

Priori za on of Stories:

- Must-Have: Login/Signup, Create Task, Assign Task, Update Status.
- Should-Have: Delete Task, Dashboard Summary.
- Nice-to-Have: Deadline No fica ons, Search/Filter Tasks.

1.4 MVP Features

For the Minimum Viable Product (MVP), the following features will be implemented:

Feature	Descrip on	Priority
User Authen ca on	Secure login/signup using JWT	High
Task Crea on	Create a task with tle, descrip on, assignee, due date	High
Task Assignment	Assign tasks to users from the registered list	High
Task Status Update	Mark tasks as "To Do", "In Progress", or "Done"	High
Dashboard	View tasks assigned to me and tasks I created	Medium

Admin Controls	Admins can delete or update any task	Medium

This MVP ensures that the system delivers essen al value immediately while allowing room for future enhancements such as no fica ons and repor ng.

1.5 Wireframes / API Endpoint List Wireframes (Text Representa on):

1. Login Page

```
-------------------------
|  Login Form
|
| [Email] [Password]   |
| [Login Bu on]       |
-------------------------
```

2. Dashboard

```
---------------------------------------
| My Tasks        | Assigned Tasks |
---------------------------------------
| Task 1: Pending | Task 5: Done   |
| Task 2: In Prog | Task 6: To Do  |
---------------------------------------
[Create Task Bu on]
```

3. Create Task Form

```
-------------------------
|  Create New Task     |
| [Title]             |
| [Descrip on]        |
| [Assignee Dropdown]  |
```

| [Due Date Picker] |

| [Save Bu on] |

API Endpoints (with Purpose):

- POST /auth/signup → Register a new user.
- POST /auth/login → Login and get JWT token.
- GET /tasks → Fetch all tasks assigned to logged-in user.
- POST /tasks → Create new task.
- PUT /tasks/:id → Update task details/status.
- DELETE /tasks/:id → Delete a task.

1.6 Acceptance Criteria

The following acceptance criteria ensure the requirements are testable and verifiable:

1. A user can register with a valid email and password.
2. A er logging in, a JWT token must be generated and stored.
3. Users without a token cannot access /tasks.
4. Crea ng a task requires tle, status, assignee, due date.
5. A user should only see tasks assigned to them or created by them.
6. Admins can view all tasks and delete/update them.
7. Status op ons must be restricted to: To Do, In Progress, Done.
8. UI should render correctly on desktop and mobile browsers.
9. API responses should follow JSON format with proper error messages.

Phase 1 Summary :

This phase thoroughly documents the problem, users, requirements, MVP, wireframes, endpoints, and acceptance criteria. It lays a strong founda on for moving into design and implementa on.