

## 🔗 MERN Stack Assignment: Project & Task Management System

### ◇ Project Overview

Develop a Project & Task Management System where an Admin can manage multiple projects and assign tasks to Team Members. Team Members can update task statuses and comment on tasks.

### ◇ Features to Implement

#### 1 User Authentication & Authorization (JWT)

Admin can: Create projects (visible to all users).  
Assign tasks to team members.  
Set task priority.

Team Members can: View all projects.  
Select a project and see all tasks.  
Filter tasks (assigned to me / all).  
Update task status (Pending, In Progress, Done).  
Comment on tasks.

#### 2 Project Management

Admin can create projects.  
All users (Admin & Team Members) can see all projects.  
Clicking on a project will open its task list.  
Admin can add tasks inside the project.

#### 3 Task Management

Admin assigns tasks to a specific user.  
Each task contains: Title, Description, Assigned User, Status, Priority.  
Tasks are visible inside the respective project page.  
Filters & Sorting: Filter by user (Only my tasks / All tasks).  
Sort by priority, status, creation date.

#### 4 Comments System

All team members can comment on any task.  
Store comment history with timestamps.

#### 5 API Rate Limiting & Security

Protect API routes using: JWT-based authentication (HTTP-only cookie).

Role-based authorization (Admin, Team Member).  
Rate-limiting middleware to prevent abuse.

## 6 Deployment

Frontend: Netlify / Vercel.

Backend: Render.

### ◇ Tech Stack

Frontend (React.js)

React with Context API or Redux Toolkit for state management.

React Router for navigation.

Axios for API requests.

TailwindCSS or Material UI for UI design.

Backend (Node.js, Express.js)

Express.js for API handling.

JWT (JSON Web Token) for authentication.

Express-validator for input validation.

Rate-limiting middleware to prevent API abuse.

Database (MongoDB)

Mongoose ORM for database modeling.

MongoDB Atlas for cloud database hosting.

### ◇ Submission Guidelines

Submit a GitHub repository with: README (setup instructions, API documentation, features) and with proper commits(if possible).

Backend API documentation (Postman).

Live deployment links (Vercel/Netlify + Render).

### 📌 Evaluation Criteria

- ☒ Authentication & Security Measures
- ☒ Database Design & Query Optimization
- ☒ Project & Task Management Features
- ☒ UI/UX & State Management
- ☒ Deployment & API Documentation