Task for Software Development Interns: Building a Task Management System With Advance Features

Overview

As a software development intern, your task is to design and develop a **Task Management System** using modern development practices. The application will allow users to create, update, delete, and manage tasks, providing insights into task progress and deadlines. The project is divided into three weeks with clear deliverables.

The final submission must include:

1. A **recorded video** demonstrating the project running.
2. A **GitHub public repository link** containing the code.

Week 1: Backend Development

1. **Set Up the Environment**
    - Choose your tech stack:
        - Backend: **Node.js with Express.js** (preferred) or any backend framework.
        - Database: **MongoDB** (preferred) or **MySQL/PostgreSQL**.
    - Initialize the project:
        - Set up a package manager (npm/yarn).
        - Install necessary dependencies.
2. **Build API Endpoints**
    - Create a REST API with the following endpoints:
        - **POST /tasks**: Add a new task.
        - **GET /tasks**: Fetch all tasks.
        - **GET /tasks/:id**: Fetch a single task by ID.
        - **PUT /tasks/:id**: Update a task.
        - **DELETE /tasks/:id**: Delete a task.
3. **Database Integration**
    - Design and implement a database schema for tasks with the following fields:
        - Title
        - Description
        - Status (e.g., Pending, In Progress, Completed)
        - Due Date

4. **Validation and Error Handling**
    - Validate incoming data using middleware (e.g., `express-validator` or `Joi`).
    - Handle errors gracefully with proper status codes and messages.

Week 2: Frontend Development

1. **Set Up the Frontend**
    - Use **React.js** or **Vue.js** for the frontend.
    - Initialize the project and structure folders for components, services, and styles.
2. **Build the User Interface**
    - Develop the following components:
        - **Task List**: Display all tasks fetched from the backend.
        - **Task Form**: Allow users to create and edit tasks.
        - **Task Details**: Show detailed information about a selected task.
3. **API Integration**
    - Use **Axios** or the Fetch API to communicate with the backend.
    - Implement CRUD functionality on the frontend.
4. **Styling and Responsiveness**
    - Style the application using CSS frameworks (e.g., **Tailwind CSS**, **Bootstrap**) or plain CSS.
    - Ensure the UI is responsive and works on different screen sizes.

Week 3: Advanced Features and Final Polishing

1. **Authentication**
    - Implement a simple authentication system (optional):
        - **Register** and **Login** endpoints.
        - Use JWT (JSON Web Tokens) for user sessions.
2. **Search and Filters**
    - Add a search bar to find tasks by title or description.
    - Provide filters to display tasks by status (Pending, In Progress, Completed).

3. **Task Progress Tracking**
    - ○ Implement a progress bar or visual indicator showing the percentage of completed tasks.
4. **Optimization and Testing**
    - ○ Optimize API calls to improve performance.
    - ○ Write basic unit tests for the backend and frontend (e.g., using Jest or Mocha).
5. **Final Submission**
    - ○ Test the application thoroughly.
    - ○ Fix any bugs or usability issues.

Submission Details

1. Deadline: 28th May, 2025
2. You will have to submit all of these tasks collectively om your console before deadline.
3. Submission Format:
    - ○ **Recorded video** demonstrating the project.
    - ○ **GitHub public repository link** with code and a README file.

Note

- ● Try to solve any errors or challenges independently to build problem-solving skills.
- ● If an issue persists, consult your **team leader** for guidance.

Good luck, and happy coding!