MERN Stack Take-Home Assignment: Task Management Tool

Objective:

Develop a comprehensive task management application that allows users to manage their tasks effectively, with features such as user authentication, task management, collaboration tools, and more.

Requirements:

1. **Frontend (React):**
    - **User Interface:**
        - **Authentication Pages:** Create pages for user login and registration.
        - **Dashboard:** Display all tasks with the ability to add, update, delete, and view details.
        - **Task Filters and Search:** Implement filtering options for tasks and a search bar.
        - **Task Prioritization:** Allow users to set priorities for tasks (High, Medium, Low).
        - **Collaboration Features:** Enable task assignment to other users and support commenting on tasks.
        - **Calendar Integration:** Incorporate a calendar view for tasks
        - **File Attachments:** Allow users to attach and manage files for each task.
        - **Time Tracking:** Provide a feature to track time spent on tasks with reporting capabilities.
2. **Backend (Node.js + Express):**
    - **APIs:**
        - **User Authentication:** Implement endpoints for user registration and login, utilizing JWT for session management.
        - **Task Management:** Create RESTful APIs for managing tasks (CRUD operations).
        - **Notifications:** Set up email notifications for task assignments and updates.
        - **Role-Based Access Control:** Implement middleware to enforce different access levels based on user roles.
        - **API Documentation:** Document all endpoints using Swagger or a similar tool.
3. **Database (MongoDB):**
    - **Schemas:**
        - **User Schema:** Fields for user credentials and role information.
        - **Task Schema:** Include task title, description, priority, status, due date, and user assignments.
        - **Comment Schema:** Store comments linked to tasks and users.
        - **Time Log Schema:** Maintain records of time logged against tasks.
4. **Security:**
    - Implement best practices like password hashing with bcrypt, securing routes with JWT, and sanitizing user inputs to prevent injection attacks.

Deliverables:

- Source code in a Git repository.
- README.md with detailed setup instructions, project overview, and a guide on how to use the application.
- (Optional) Link to a live demo on a cloud platform.

Evaluation Criteria:

- **Functionality:** The application must meet all the listed requirements.
- **Code Quality:** Code should be clean, well-documented, and follow best practices.
- **Security:** The application should securely handle user data and interactions.
- **User Experience:** The application should be intuitive and easy to navigate.
- **Documentation:** Documentation should be clear and detailed, enabling easy setup and usage.