Objective

The goal of this assignment is to evaluate your frontend and backend development skills, along with your ability to work with containerization, API design, database management, and CI/CD pipelines. The estimated time to complete the assignment is 2-3 days however you may take a maximum of one week from the date of receiving this assignment.

Assignment Overview

You are required to develop a task management web application with the following features:

Core Requirements:

1. **User Authentication & Authorization**
   - Implement signup/login functionality using JWT-based authentication.
   - Users should be able to create an account, log in, and manage their sessions securely.
2. **Task Management**
   - Users can Create, Read, Update, and Delete (CRUD) tasks. (Example: Product Inventory Hub to edit products)
   - Users should only be able to access and modify their own tasks.
3. **Frontend Development**
   - Build a responsive web application using React, Vue, or Angular.
   - Use Redux, Vuex, or Context API for state management.
   - Implement form validation and error handling.
4. **Backend Development**
   - Use Node.js (Express.js) or Python (Django/FastAPI) to develop the backend API.
   - Implement RESTful API endpoints for authentication and task management.
   - Ensure proper request validation and error handling.
5. **Database Management**
   - Use PostgreSQL, MySQL, or MongoDB to store user and task data.
   - Use ORM such as Sequelize, Prisma, or Mongoose for database interactions.
   - Add database seed scripts to create the DB with tables
6. **Containerization & DevOps**
   - Dockerize the application using Docker and Docker Compose.
   - Provide a Dockerfile for the backend and frontend.
   - Set up a CI/CD pipeline using GitHub Actions, GitLab CI/CD, or Jenkins to automate builds and tests.

Bonus (Optional but Preferred)

- Implement Task Notifications (via WebSockets, email, or push notifications).
- Write Unit and Integration Tests using Jest, Mocha, or PyTest.
- Deploy the application on a cloud platform (AWS/GCP/Azure) with a CI/CD pipeline.
- Use Kubernetes for container orchestration.

Submission Guidelines:

- Submit the source code via a GitHub repository (ensure it is public or provide access).
- Include a README.md with the following details:
    - Setup instructions.
    - API documentation.
    - Any assumptions or decisions made.
- Ensure the application is fully functional and tested before submission.

Evaluation Criteria:

1. Code Quality & Best Practices (Clean, modular, maintainable code)
2. Frontend Implementation (UI/UX, state management, responsiveness)
3. Backend API Design (Security, efficiency, error handling)
4. Database Schema & Queries (Optimization, indexing, ORM usage)
5. Containerization & CI/CD (Proper Dockerization, automated workflows)
6. Testing & Documentation (Unit tests, API documentation, setup instructions)

Questions?

If you have any questions, feel free to reach out!