Full-Stack Developer Assignment

Objective

Evaluate the candidate's ability to build a full-stack application by integrating frontend, backend, and database components. Docker is to be used for deployment, emphasizing best practices in project structuring and implementation.

Task Overview

1. **Frontend Development**
    - Implement routing for three pages.
    - Output the frontend build using a Dockerfile with Webpack.
2. **Backend Development**
    - Use Golang or Java-Springboot.
    - Utilize the provided SQL script for database setup and mock data.
    - Create APIs to serve data for the three frontend pages.
    - Write unit test (optional)
    - Optimized services latency (optional)
    - Optimized database schema (optional)
    - Including services stress test report such as locust, k6(optional)
3. **Deployment with Docker**
    - Develop a `docker-compose.yml` file to integrate the frontend, backend, and database.

Resources Provided

- **Design**: design.jpeg
- **Database Schema (SQL script)**: schema.sql
- **Mock Data (SQL script)**: mock.zip
- **HTML and CSS Files**: web.zip

Link: Download Link

Expectations

- **Project Structure**: Follow scalable and maintainable best practices.
- **Backend APIs**: Implement modular, secure APIs with proper error handling.
- **Frontend Integration**: Seamlessly connect the frontend with backend APIs.
- **Dockerization**: Ensure efficient Dockerfiles and smooth service integration using Docker Compose.

Deliverables

1. A functional full-stack application running via Docker, with three frontend pages.
2. A `README.md` with setup instructions, project structure overview, and backend API documentation.

Timeline

Complete the project within 2 weeks and submit via Git with the repository link.