Binalyze Coding Challenge

VueJS and Golang Web Application with SQLite

Objective:

Develop a sophisticated web application using VueJS for the frontend and Golang for the backend. The application will interact with an SQLite database, which keeps consumption data of running processes such as memory and CPU usage in varied intervals. The frontend will display this data in a dynamic and interactive manner.

Task Specifications:

1. **Backend Development (Golang):**
    - **Version Requirement:** Golang version 1.20 or higher.
    - **Framework:** Use the echo/gin framework.
    - **Database Integration:**
        - Integrate an SQLite database for data storage.
        - Implement a mechanism to insert process resource consumption data into the database in varied intervals.
    - **API Endpoints:**
        - `GET /api/v1/data`: Retrieves current data from the database.
        - `WebSocket /api/v1/ws`: Establishes a WebSocket connection for real-time data updates.
2. **Frontend Development (VueJS):**
    - **Version Requirement:** Vue 3.
    - **User Interface Challenges:**
        - **Dynamic Data Display:** Develop a dashboard or similar interface to display the data from the SQLite database. Include interactive elements like charts or graphs, updating in real-time.
        - **Real-Time Data Sync:** Use WebSockets to synchronize the data displayed on the frontend with the latest entries in the database.
        - **User Interactions:** Allow users to filter, sort, and explore the data in various forms.

        - **Responsive Design:** Ensure the application is fully responsive across different devices and screen sizes.
    - **VueJS Advanced Features:**
        - Utilize Vuex for efficient state management, especially handling the WebSocket connections and real-time data updates.
        - Implement comprehensive error handling and user feedback mechanisms.
        - Cover the frontend with appropriate unit and integration tests.
3. **Containerization:**
    - Provide Dockerfiles for both frontend and backend.
    - Create a `docker-compose.yml` file for container orchestration.
    - Ensure the application is runnable with `docker-compose up`.
4. **Documentation and Delivery:**
    - Write a detailed `README.md` file with setup and running instructions.
    - Document the code thoroughly, explaining complex functionalities and design decisions.
5. **GitHub Repository:**
    - Host the project in a private GitHub repository.
    - Include all source code, Docker configurations, and documentation.

Evaluation Focus:

- Effective integration of the SQLite database with the Golang backend.
- Implementation of process resource consumption data and insertion into the database.
- Real-time data synchronization and visualization on the frontend.
- Quality and functionality of the user interface, including responsiveness and interactivity.
- Overall code quality, including organization, documentation, and testing.

This task aims to assess the candidate's ability to create a complex, real-time web application using VueJS and Golang, with a focus on backend database integration, dynamic data handling, and creating an engaging user experience.