CactusTech

Golang & Elixir Developer Challenge

Challenge Overview

Objective: Develop a Real-time Metrics Dashboard

The goal of this challenge is to build a real-time dashboard for monitoring server metrics. You will create a microservices architecture with a service written in Golang and another in Elixir, both containerized using Docker.

1. Project Setup:

   - Initialize a new project with a Golang service and an Elixir service.
   - Dockerize both services ensuring communication between them.

2. Golang Service:

   - Simulate real-time data generation for server metrics.
   - Create REST API endpoints to access the metrics.
   - Use goroutines for concurrent data handling.

3. Elixir Service:

   - Use Elixir and Phoenix to develop a real-time backend service.
   - Implement Phoenix Channels for data subscription and broadcasting.

4. Database Integration:

   - Design a SQL schema for metrics data.

   - Integrate the database with Golang to store metrics.

5. Frontend Development:

   - Develop a front-end interface in JavaScript to display metrics.

   - Implement real-time data streaming using WebSockets.

6. Testing:

   - Write integration tests for the Golang API.

   - Test Phoenix Channel communication in Elixir.

   - Conduct front-end testing for the data display functionality.

7. Documentation:

   - Provide setup and run instructions in a README file.

   - Document the project's structure and API endpoints.

Deliverables:

- Source code repository.

- Dockerfiles and Docker-Compose configurations.

- Test suite for backend and frontend.

- Documentation in the form of a README.md file.

Evaluation Criteria:

- Functionality and real-time data display.

- Clean code and proper documentation.

- Efficient use of concurrency in Golang and Elixir.

- Effective database schema design and integration.

- Real-time communication with Phoenix Channels and WebSockets.

- Docker setup for development environment.

- Comprehensive tests for all parts of the application.