Dec-2024

**SRE home Task**

**Task: Create a Docker Compose Application with MySQL and Flask**

**Bonus : Run the app in K8s locally ( minikube/Rancher)**

**Objective**:  
Your task is to create a Docker Compose application consisting of two services:

1. A **MySQL database** to store data.
2. A **Flask-based web application** to fetch data from the MySQL database and display it on a web page.

**Requirements:**

1. **MySQL Service**:
   * Use the official MySQL Docker image.
   * Initialize the database with a sample table (employees) using an SQL script.
   * The employees table should contain the following columns:
     + id (INT, Primary Key, Auto Increment)
     + name (VARCHAR)
     + role (VARCHAR)
   * Insert at least 3 sample rows into the employees table.
2. **Flask Web Application**:
   * Create a simple Flask app that connects to the MySQL database.
   * Fetch data from the employees table.
   * Display the data in an HTML table format on a webpage (e.g., at /).
3. **Docker Compose**:
   * Create a docker-compose.yml file to define both services.
   * The Flask app should start only after the MySQL service is ready.
4. **Directory Structure**: Your project directory should include the following files:

csharp

Copy code

docker-compose-mysql-webapp/

├── docker-compose.yml

├── init-db/

│ └── init.sql

├── webapp/

│ ├── Dockerfile

│ ├── app.py

│ ├── requirements.txt

└── README.md

**Deliverables:**

1. A complete project directory containing:
   * docker-compose.yml
   * SQL initialization script (init.sql)
   * Flask application code (app.py) and its dependencies (requirements.txt and Dockerfile).
2. A README.md file explaining:
   * How to build and run the application using Docker Compose.
   * The functionality of the application.

**Evaluation Criteria:**

1. **Correctness**:
   * Does the application start successfully with docker-compose up?
   * Does the Flask app fetch and display the data from the MySQL database?
2. **Code Quality**:
   * Is the code well-structured, readable, and modular?
3. **Documentation**:
   * Is the README.md file clear and easy to follow?
4. **Bonus Points**:
   * The Flask app gracefully handles database connection errors.
   * The services follow best practices, such as using environment variables for configuration.

**Submission Instructions:**

1. Package your solution in a ZIP file or upload it to a Git repository.
2. Provide clear instructions in the README.md file for testing the solution.

This task is designed to assess your practical knowledge of Docker, Flask, and MySQL integration. Good luck!