

Instructions:

- Please complete the following tasks using the provided technologies: Golang, PostgreSQL, and Docker.
- Create your own GitHub account or any other repositories.
- Once you have finished, please provide a link to your GitHub repository branch. Make sure the repository is publicly accessible.

Tasks:

Task 1: Database Setup

- Set up a PostgreSQL database using Docker. You can use any official PostgreSQL Docker image.
- Create a table named employees with the following columns:
 - id (serial, primary key)
 - first_name (text)
 - last_name (text)
 - email (text)
 - hire_date (date)
- Using migration to create the requirement table is a plus

Task 2: Golang API

- Create a Golang RESTful API with the following endpoints:
 - GET /employees: Retrieve a list of all employees from the employees table.
 - GET /employees/{id}: Retrieve an employee by their id.
 - POST /employees: Create a new employee record in the database.
 - PUT /employees/{id}: Update an employee record by their id.
 - DELETE /employees/{id}: Delete an employee record by their id.
 - Implement appropriate error handling and validation for these endpoints.
- Provide API documentation using OpenAPI spec is a plus

Task 3: Unit Testing

- Write unit tests for at least two of the API endpoints.
- Use a testing framework of your choice (e.g., testing package in Golang).

- Ensure good test coverage and provide a brief explanation of your testing strategy.

Task 4: Documentation

- Document how to run your Golang application using Docker.
- Include information on how to set up and run the PostgreSQL container.
- Provide a brief description of the API endpoints and their expected input/output.

Task 5: Docker Compose

- Create a docker-compose.yml file that sets up both the Golang API container and the PostgreSQL container.
- Include any necessary environment variables for the API container to connect to the PostgreSQL database.
- Ensure that the API container waits for the PostgreSQL container to be ready before starting.

Submission:

- Commit your changes.
- Push the branch to your GitHub repository.
- Provide a link to your GitHub repository branch for review.

Notes:

- You are free to use any Golang web framework or libraries of your choice.
- Keep your code clean, well-structured, and maintainable.
- The focus is not just on completing the tasks but also on the quality and organization of your code.
- You may use any additional libraries or tools you find necessary to accomplish the tasks.
- This assessment is designed to evaluate your skills in Golang, Docker, and PostgreSQL, as well as your ability to create a functional and documented application.