Java, SpringBoot, OpenAPI, and Docker Technical Challenge:

Timeframe - Please return within 72 hours.

Please upload your finished product & send us the link to access it.

Overview

In this technical challenge, you will build a SpringBoot web application that integrates with a PostgreSQL database and uses Docker to run in a container. The application should have a REST API that allows users to perform CRUD (create, read, update, delete) operations on a data model representing a simple e-commerce platform. The REST API should be defined using OpenAPI.

Requirements

1. The application should be built using SpringBoot and Java 21.
2. The application should integrate with a PostgreSQL database using Spring Data JPA.
3. The application should have a REST API that allows users to perform the following operations:

Product API Calls:

- GET /api/products - Get a list of all products
- GET /api/products/{id} - Get a product by ID
- POST /api/products - Create a new product
- PUT /api/products/{id} - Update an existing product
- DELETE /api/products/{id} - Delete a product by ID

Order API Calls:

- GET /api/orders - Get a list of all orders
- GET /api/orders/{id} - Get an order by ID
- POST /api/orders - Create a new order
- PUT /api/orders/{id} - Update an existing order
- DELETE /api/orders/{id} - Delete an order by ID

OrderItem API Calls:

- GET /api/order-items - Get a list of all order items
- GET /api/order-items/{id} - Get an order item by ID
- POST /api/order-items - Create a new order item
- PUT /api/order-items/{id} - Update an existing order item
- DELETE /api/order-items/{id} - Delete an order item by ID

4. The REST API should be defined using OpenAPI 3.0.
5. The application should use Docker to run in a container. You should provide a Dockerfile and a docker-compose.yml file to run the application.

6. The application should be fully tested using JUnit and Mockito. You should provide a set of automated tests that cover the application's functionality.
7. The application should handle errors gracefully and return appropriate HTTP status codes.
8. The application should be well-documented, with clear instructions on how to build, run, and test the application.

Evaluation Criteria

We will evaluate your submission based on the following criteria:

1. Quality of code: the code should be well-structured, easy to read and maintain, and follow best practices for Java and SpringBoot development.
2. Functionality: the application should meet the requirements outlined above and function correctly.
3. Use of OpenAPI: the REST API should be defined using OpenAPI 3.0 and adhere to best practices for API design.
4. Use of Docker: the application should use Docker to run in a container.
5. Testing: the application should have a comprehensive set of automated tests that cover the application's functionality.
6. Documentation: the application should be well-documented, with clear instructions on how to build, run, and test the application.

Submission Guidelines

Please submit your solution as a compressed archive (e.g., zip or tar.gz) containing the following:

1. The source code for the application.
2. A README file with clear instructions on how to build, run, and test the application.
3. A Dockerfile and a docker-compose.yml file to run the application.