Challenge

Containerize RESTful Services and Database by Using Docker

Implementation Environment

- You must have access to GitLab.
- Install git to be able to clone and push code to the repository.
- You must be familiar with forking and cloning a git repository.
- You must have Docker installed on your machine.
- Steps to install Docker
  - Install Docker Desktop
  - Register on Docker Hub so that images can be pulled

Authentication of a Customer

The customers of an ecommerce application should be registered to become the users of the application and must be authenticated at login.

Create a Spring Boot application with an entity class `Customer` with `customerId, customerName, customerPassword, and customerPhoneNo.`

Implement authentication using JWT. Dockerize the MySQL database and the Spring Boot application.

Note : Use the solution from the previous Sprint practice session.

CHALLENGE

Instructions for the Challenge

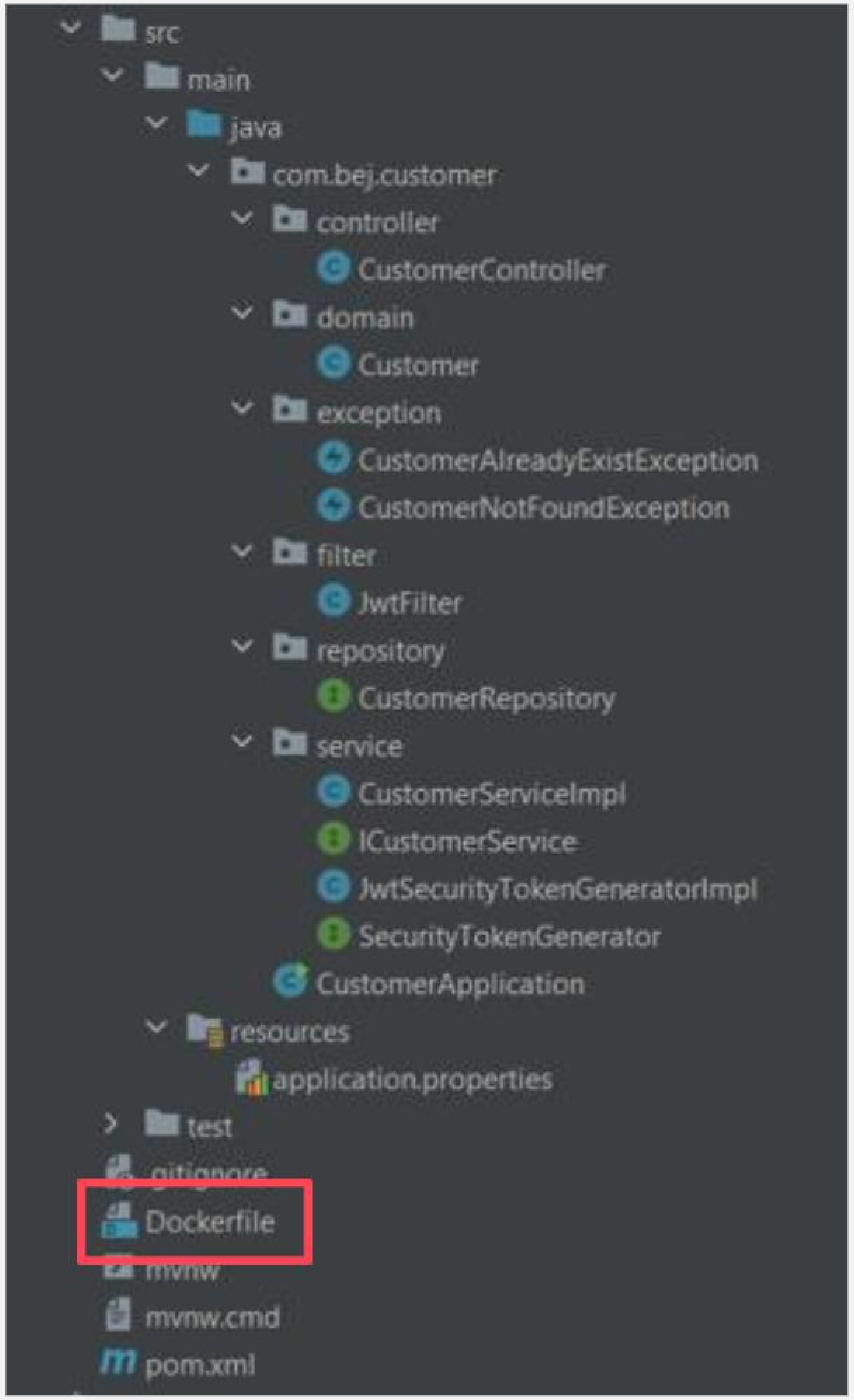
- There is no boilerplate for the challenge.
- Clone the previous practice as instructed and Dockerize the application using a Dockerfile.

Tasks

- Start the Docker Desktop.
- Execute the Docker commands on PowerShell or command prompt.
- Create a Docker network named `customer-network.`
- Pull the MySQL image from Docker hub.
- Run the MySQL container with the name, `mysqlservice,` on the `customer-network.`

Tasks (contd..)

- Clone the repository of your previous practice solution that you created earlier: **BEJ_C3_S1_SpringBoot_JWT_PC_1.**
- Create a `Dockerfile` at the root of your application.
- Provide the Docker configurations in the `Dockerfile.`
- In the `application.properties` file, change the url of MySQL from `localhost` to `mysqlservice`, which is the name of the MySQL container.



Tasks (contd..)

- Run `mvn clean compile package` command to build the JAR file of the application that will be used to build the Docker image of the boot application.
- Build the customer Spring Boot application image at the root of the application with the name of the `customer-app:v1`.
- Run the `customer-app:v1` as a Docker container on the `customer-network` where the MySQL container is running.
- Test the output on Postman.

Submission Instructions

- Create a Git repository named **BEJ_C3_S2_Containerize_REST_API_And_Database_MC_1.**
- Push the solution to the repository and submit for review.