Determine the geographic distance between two postcodes in the UK. This project utilizes the following technologies:

* Java 17
* Spring Boot 3.4
* Maven
* MySQL
* Docker

**Starting up the service:**

This service already uses a Docker container with a database connection and necessary configurations. Follow these steps to build and run the service:

1. **Build the project**: mvn clean install or mvn install
2. **Build the Docker image**: docker-compose build
3. **Run the application**: docker-compose up

Use Postman to send requests and receive responses from the MySQL database.

Example:

In Authorization [ Auth Type: Bearer Token, Token generated from KEYCLOAK for authentication]

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Set a parameter in postman and execute service url in postman:

A screenshot of a computer

Description automatically generated

The code and project structure easy to read and maintain it:

A screenshot of a computer

Description automatically generated

**Project Structure:**

pom.xml – Used for Maven build and to download additional dependencies.

Dockerfile – For dockerizing the Spring Boot application.

docker-compose.yml – Defines the services such as database version, port, and other configurations if needed.

**Packages:**

* com.dhl.geographic – This package contains the main Spring Boot application.
* com.dhl.geographic.config – Any additional configuration class files can be created in this package.
* com.dhl.geographic.model – Used for creating entity files based on the database tables and mapping table relationships.
* com.dhl.geographic.repository – Used for JPA (CRUD operations), HQL, or native queries to get data from entities.
* com.dhl.geographic.service – This package contains business logic and other applied logics.
* com.dhl.geographic.controller – Contains REST controllers and handles validation of request data.
* com.dhl.geographic.dto – Contains immutable data transfer objects for requesting and responding to data.

In the test folder, integration or unit tests are applied to check performance and logic.