**Instructions to Run the Application**

**1. Backend (Java Spring Boot Application)**

1. **Clone or Download the Repository**:
   * If you have the source code stored in a Git repository, clone it:

bash

git clone <repository-url>

* + If you have a .zip file, unzip it into a directory of your choice.

1. **Open the Project in Your IDE**:
   * If you are using **Eclipse**:
     1. Open Eclipse and choose your workspace.
     2. Go to File > Import > Existing Maven Projects.
     3. Select the unzipped project folder and click Finish.
   * If you are using **IntelliJ IDEA**:
     1. Open IntelliJ and select Open Project.
     2. Navigate to the project folder and select it.
2. **Configure MySQL Database**:
   * Create a MySQL database named task\_management (or the name specified in your application.properties or application.yml):

Sql

CREATE DATABASE task\_management;

* + Configure the database connection details in src/main/resources/application.properties:

properties

spring.datasource.url=jdbc:mysql://localhost:3306/task\_management

spring.datasource.username=root

spring.datasource.password=yourpassword

spring.jpa.hibernate.ddl-auto=update

1. **Run the Spring Boot Application**:
   * In Eclipse, right-click the main application class (e.g., TaskManagementApplication.java) and choose Run As > Spring Boot App.
   * In IntelliJ, click the Run button or right-click the main class and select Run 'TaskManagementApplication'.

The backend Spring Boot application should now be running on <http://localhost:5454/>.

1. **Test the API**:
   * You can use **Postman**, **cURL**, or any other tool to send requests to the API.
   * Example endpoints might include:
     1. POST <http://localhost:5454/api/login> for login requests.
     2. GET <http://localhost:5454/api/tasks> to all fetch task details.
   * Ensure that the API endpoints and request methods match your project’s requirements.

**2. Database Configuration**

Make sure the database schema is correctly set up by running the appropriate SQL scripts or allowing the application to automatically create tables based on the JPA entities.

**Troubleshooting**

1. **Database Connection Issues**:
   * Ensure that the MySQL server is running and that the database credentials in application.properties are correct.
   * Ensure that the MySQL JDBC Driver is included in the project dependencies.
2. **Port Conflicts**:
   * If port 5454 is already in use, change the port in the application.properties file:

properties

server.port=5454

1. **Testing**:
   * Since there is no frontend, use Postman or any REST client to test the various endpoints.
   * You can also check logs in the IDE's console to verify that the backend services are working as expected.