Detailed, step-by-step, build/installation/initialization/launch instructions

- **Prerequisites:**

- Install the latest version of JDK.
- Install Node.js and npm for managing frontend dependencies.
- Set up a MySQL database and note down the connection details.

STEPS:

- **Part 1: Spring Boot Backend**

1. Create a Spring Boot Project:

   Utilize Spring Initializer or your preferred IDE to initiate a new Spring Boot project. Include dependencies like Spring Web, Spring Data JPA, and MySQL.

2. Configure MySQL:

   Set up the application.properties or application.yml file with your MySQL database connection details.

3. Create JPA Entity and Repository:

   Define the Employee entity with JPA annotations. Develop a JPA repository for basic CRUD operations.

4. Build Get All Employee REST API:

   Implement a REST API endpoint to retrieve all employees.

5. Build Create and Update Employee REST APIs:

Develop REST API endpoints for creating and updating employees.

6. Build Get Employee by ID and Delete Employee REST APIs:

   Implement REST API endpoints to fetch a single employee by ID and delete an employee.

7. Test Your Backend:

   Run your Spring Boot application and test the APIs using tools such as Postman.

Part 2: React Frontend

1. Create React Application:

   Use create-react-app or your preferred method to initialize a new React application.

2. Add Bootstrap 5:

   Integrate Bootstrap 5 into your React app by installing it via npm.

3. Configure Routing:

   Set up routing using react-router-dom to navigate between different components.

4. Create ListEmployeeComponent:

   Build a component to display the list of employees using the useState hook.

5. Create AddEmployeeComponent:

   Develop a form component for adding new employees.

6. Connect React to Backend:

   Use fetch or axios to establish communication between your React app and the Spring Boot backend APIs.

7. Testing the Frontend:

   - Run the React application to ensure proper communication with the backend.
   - Build and Launch

8. Build the Spring Boot JAR:

   Use your IDE or Maven to build the Spring Boot project, resulting in a JAR file.

9. Run the Spring Boot Application:

   Execute the JAR file or run the application from your IDE.

10. Build and Launch React App:

    Use npm to build your React app (npm run build) and configure your Spring Boot application to serve the static files.

Access the Application:

Open a web browser and navigate to the specified URL (usually http://localhost:8080) to access and use the application.