**Employee Management Application Documentation**

**Project Overview**

This project is a Full Stack Employee Management Application built using:

* **Backend:** Spring Boot with JPA
* **Frontend:** HTML, CSS, JS with Thymeleaf
* **Database:** MySQL

**Getting Started**

**1. Configure Database**

* Update application.properties file in src/main/resources/ with your MySQL credentials:

spring.datasource.url=jdbc:mysql://localhost:3306/employee\_db

spring.datasource.username=root

spring.datasource.password=your-password

spring.jpa.hibernate.ddl-auto=update

* Create the employee\_db database in MySQL:

**2. Build and Run Application**

* Using Maven:
* Application runs at: http://localhost:8080

**API Endpoints**

**1. Employee APIs**

* **Get All Employees:** GET /employees
  + Response:
* **Get Employee by ID:** GET /employees/{id}
  + Response:
* **Add Employee:** POST /employees
  + Request Body:
  + Response:
* **Update Employee:** PUT /employees/{id}
  + Request Body:
  + Response:
* **Delete Employee:** DELETE /employees/{id}
  + Response: 204 No Content

**Validation and Error Handling**

**1. Employee Validation**

* **Email:** Must be unique, valid, and properly formatted.
* **firstName, lastName, department:** Cannot be empty or null.

**2. Error Handling**

* EmailAlreadyExistsException - When email already exists.
* ResourceNotFoundException - When employee is not found.
* MethodArgumentNotValidException - When validation fails.

**Swagger Documentation**

**1. Add Swagger Dependency**

**2. Configure Swagger in application.properties**

**3. Access Swagger UI**

* Run the application and go to:

**Unit Testing**

**1. Run Tests**

**2. Unit Tests Coverage**

* **Controller Layer:** EmployeeControllerTest.java
* **Service Layer:** EmployeeServiceImplTest.java
* **Repository Layer:** EmployeeRepositoryTest.java