

Employee Management System

A full-stack Employee Management web application built with Spring Boot, JPA, MySQL, and Thymeleaf, integrated with Swagger UI for API documentation.

Project Structure

EmployeeManagement/

```
├── src/
│   ├── main/
│   │   ├── java/com/example/EmployeeManagement/
│   │   │   ├── config/SwaggerConfig.java
│   │   │   ├── controller/EmployeeController.java
│   │   │   ├── entity/Employee.java
│   │   │   ├── repository/EmployeeRepository.java
│   │   │   ├── service/EmployeeService.java
│   │   │   └── EmployeeManagementApplication.java
│   │   ├── resources/
│   │   │   ├── static/css/style.css
│   │   │   ├── templates/list.html
│   │   │   ├── templates/add.html
│   │   │   ├── templates/view.html
│   │   │   └── application.properties
│   └── pom.xml
```

Technologies Used

- Java 21
- Spring Boot 3.4.5
- Spring Data JPA
- Thymeleaf

- MySQL
 - Swagger (OpenAPI 3)
 - HTML/CSS/JS
-

Database

MySQL Database: employeedb

Entity: Employee

Field Name Type

id Long

firstName String

lastName String

email String

CREATE DATABASE employeedb;

USE employeedb;

```
CREATE TABLE employee (  
    id BIGINT PRIMARY KEY AUTO_INCREMENT,  
    first_name VARCHAR(255),  
    last_name VARCHAR(255),  
    email VARCHAR(255)  
);
```

Setup Instructions

1. Clone the repository:

```
git clone https://github.com/yourusername/EmployeeManagement.git
```

2. Configure MySQL in application.properties:

```
spring.datasource.url=jdbc:mysql://localhost:3306/employeedb
```

```
spring.datasource.username=root
```

```
spring.datasource.password=yourpassword
```

spring.jpa.hibernate.ddl-auto=update

3. Run the application using IntelliJ or:

```
./mvnw spring-boot:run
```

4. Visit:


- App: <http://localhost:8080/users>
- Swagger: <http://localhost:8080/swagger-ui.html>

Features

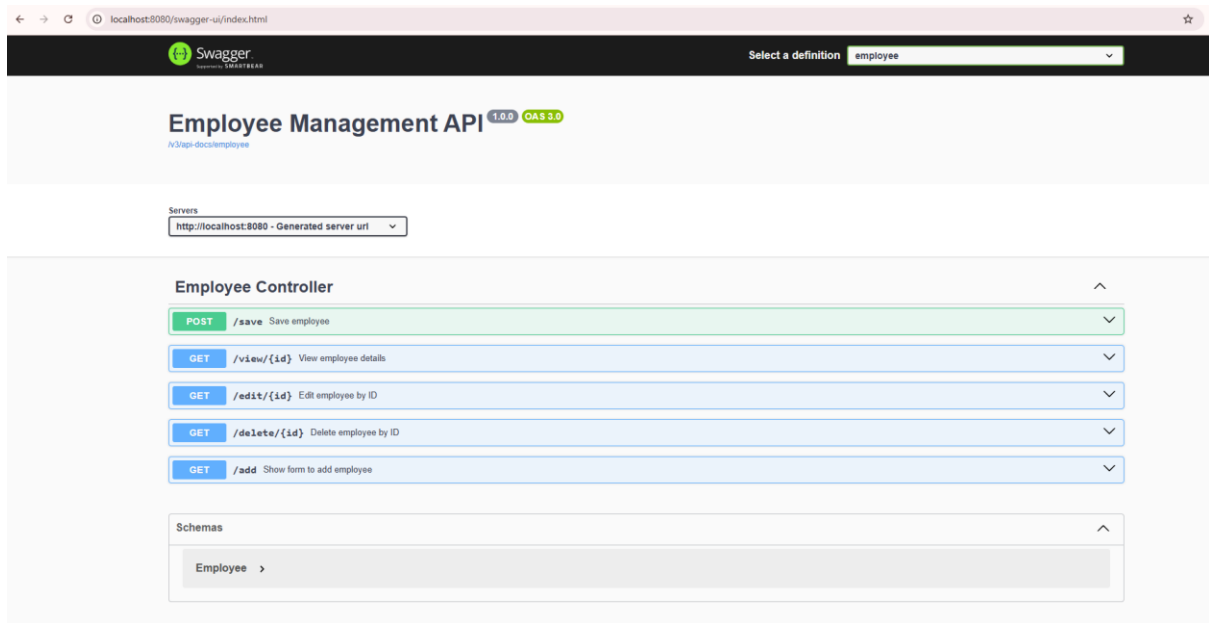
- Add new employees
- View employee list
- Update employee details
- Delete employee records
- View employee details (popup-style)
- Full API documentation via Swagger UI

API Endpoints

Method	Endpoint	Description
GET	/api/employees	Get all employees
GET	/api/employees/{id}	Get employee by ID
POST	/api/employees	Add new employee
PUT	/api/employees/{id}	Update employee
DELETE	/api/employees/{id}	Delete employee

 Swagger UI Snapshots

1. Swagger Homepage



Author

- Name: Sneha Kotak
- GitHub: skotak