Riya Chandrakantbhai Bhimani

Email: riyabhimani.me@gmail.com

Project-2: Employee Management System

- **Abstract :**
- The **Employee Management System** is a web application built using Spring Boot that simplifies the management of employee information for organizations. It allows HR departments to efficiently add, edit, and delete employee records, including details like names, emails, and departments. The system utilizes Spring Data JPA for database interactions with MySQL and features a layered architecture for maintainability. With a user-friendly interface powered by Thymeleaf, the application enhances productivity by automating record-keeping tasks, ultimately improving organizational efficiency and data accuracy.

Project-2: Employee Management System

- **Objective** :

•**Streamline Management**: Centralize employee information for easy access and updates.
•**Enhance Data Accuracy**: Minimize errors through structured data entry.
•**Improve User Experience**: Create an intuitive interface for better navigation.
•**Quick Information Access**: Enable rapid retrieval of employee details.
•**Support CRUD Operations**: Implement basic Create, Read, Update, and Delete functionalities.
•**Promote Data Security**: Protect sensitive employee information with access controls.
•**Enable Reporting**: Provide tools for generating employee data reports.
•**Integration Capability**: Facilitate potential integration with other HR systems.

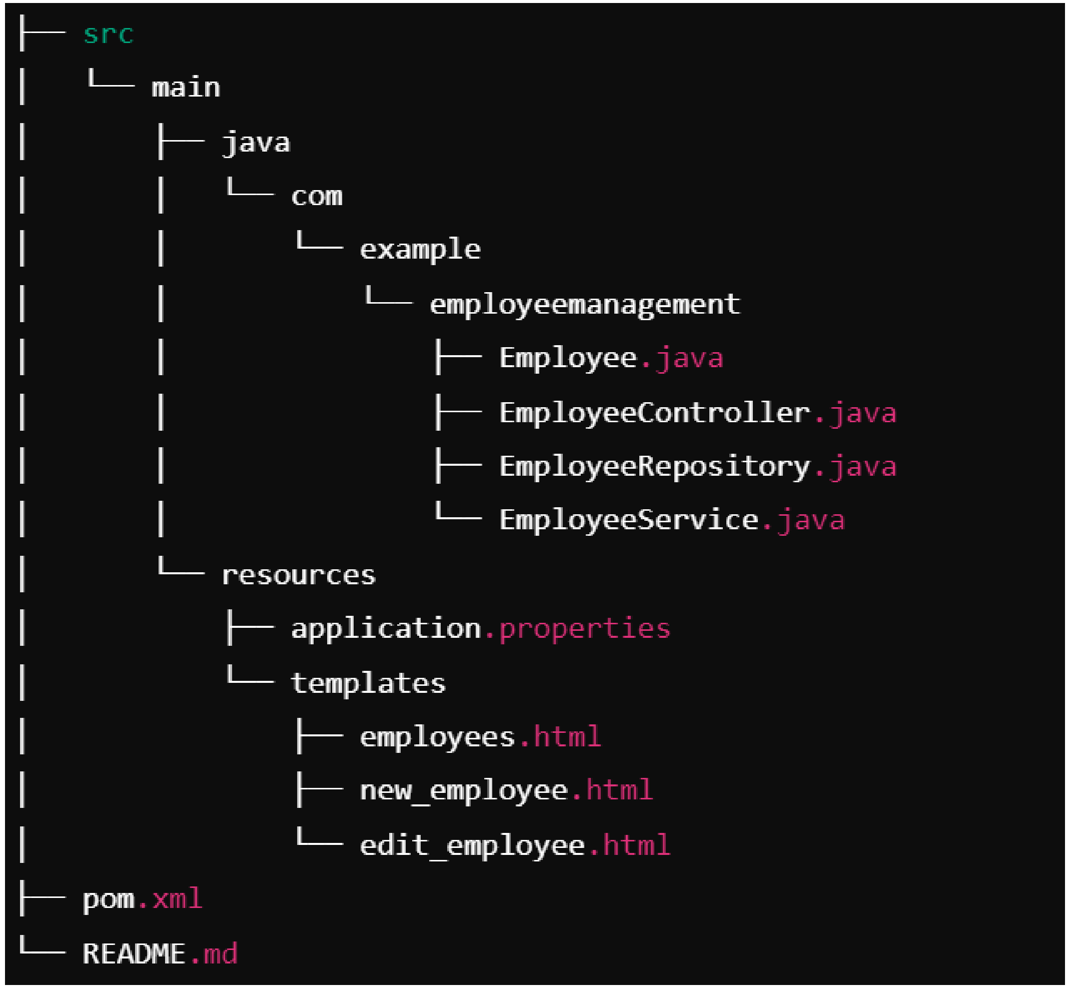
Project-2: Employee Management System

- **Methodology:**

  - **Requirements**: Gather system needs.
  - **Design**: Plan architecture.
  - **Tech**: Use Spring Boot, MySQL, Thymeleaf.
  - **Implementation**: Develop features.
  - **Testing**: Ensure functionality.
  - **Deployment**: Deploy application.
  - **Training**: Guide users.
  - **Maintenance**: Provide support.

Project-2: Employee Management System

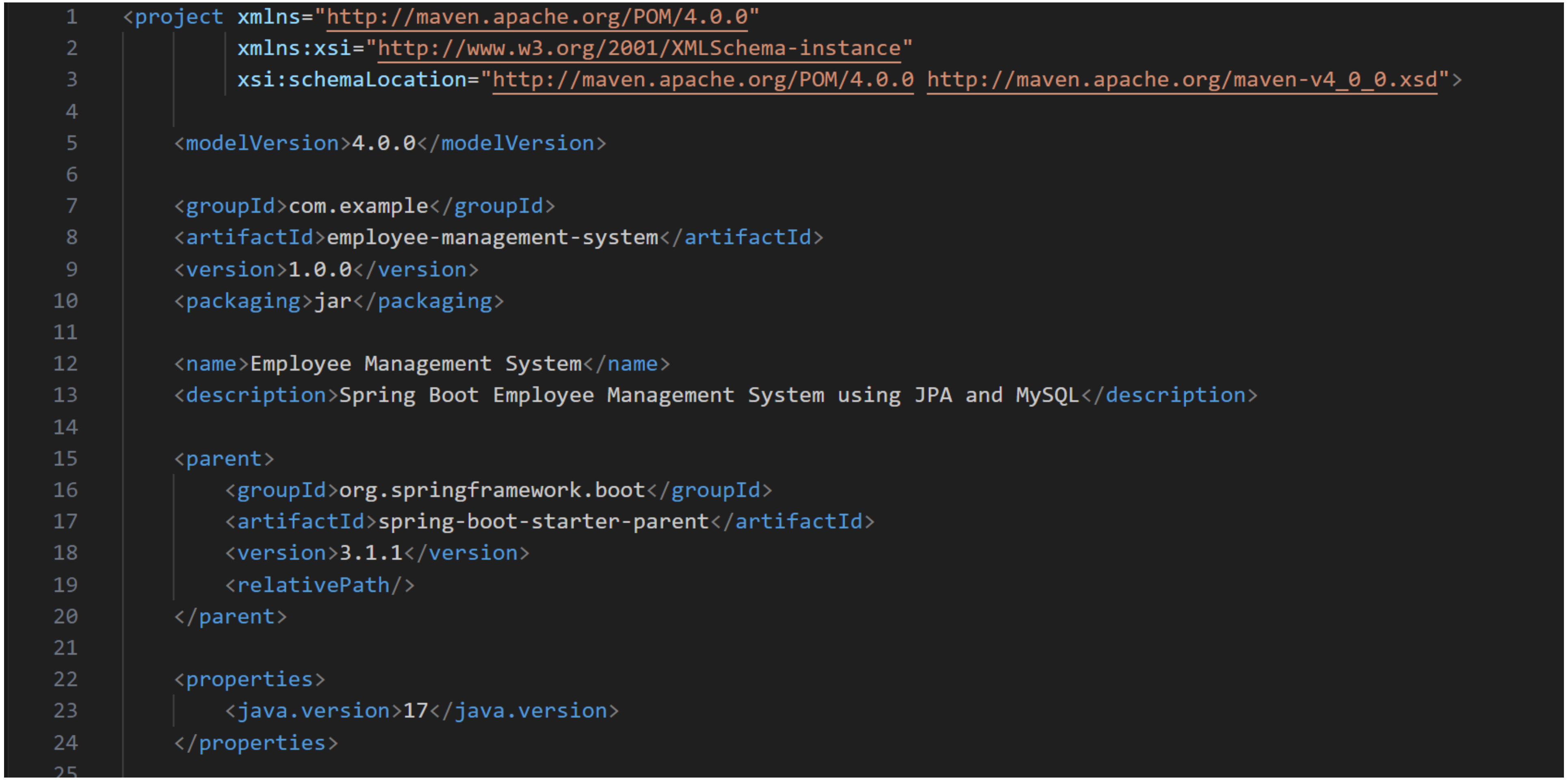
Project Structure :



```
├── src
│   └── main
│       ├── java
│       │   └── com
│       │       └── example
│       │           └── employeemanagement
│       │               ├── Employee.java
│       │               ├── EmployeeController.java
│       │               ├── EmployeeRepository.java
│       │               └── EmployeeService.java
│       └── resources
│           ├── application.properties
│           └── templates
│               ├── employees.html
│               ├── new_employee.html
│               └── edit_employee.html
├── pom.xml
└── README.md
```

Project-2: Employee Management System

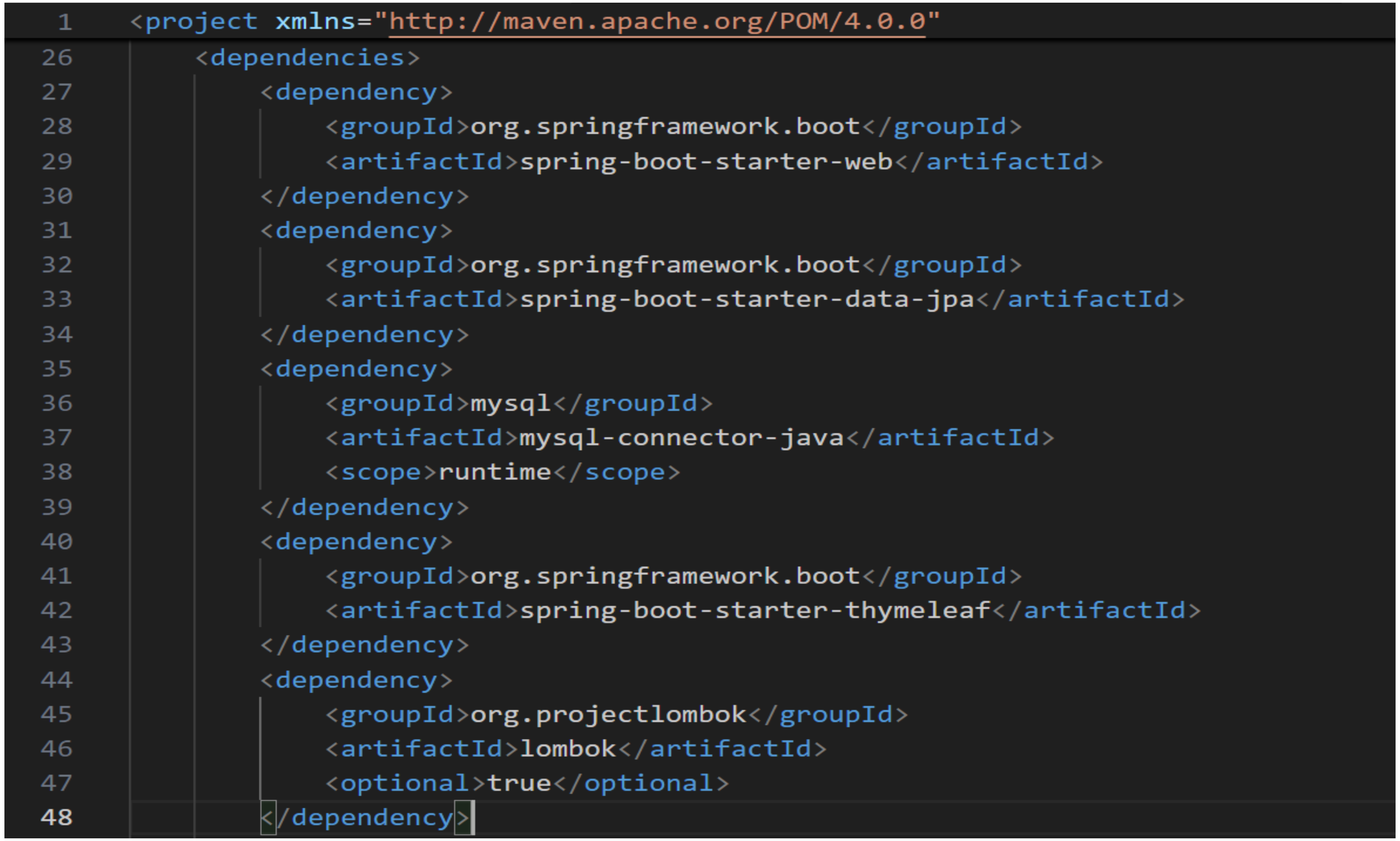
pom.xml :



```xml
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">

    <modelVersion>4.0.0</modelVersion>

    <groupId>com.example</groupId>
    <artifactId>employee-management-system</artifactId>
    <version>1.0.0</version>
    <packaging>jar</packaging>

    <name>Employee Management System</name>
    <description>Spring Boot Employee Management System using JPA and MySQL</description>

    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>3.1.1</version>
        <relativePath/>
    </parent>

    <properties>
        <java.version>17</java.version>
    </properties>
```

Project-2: Employee Management System

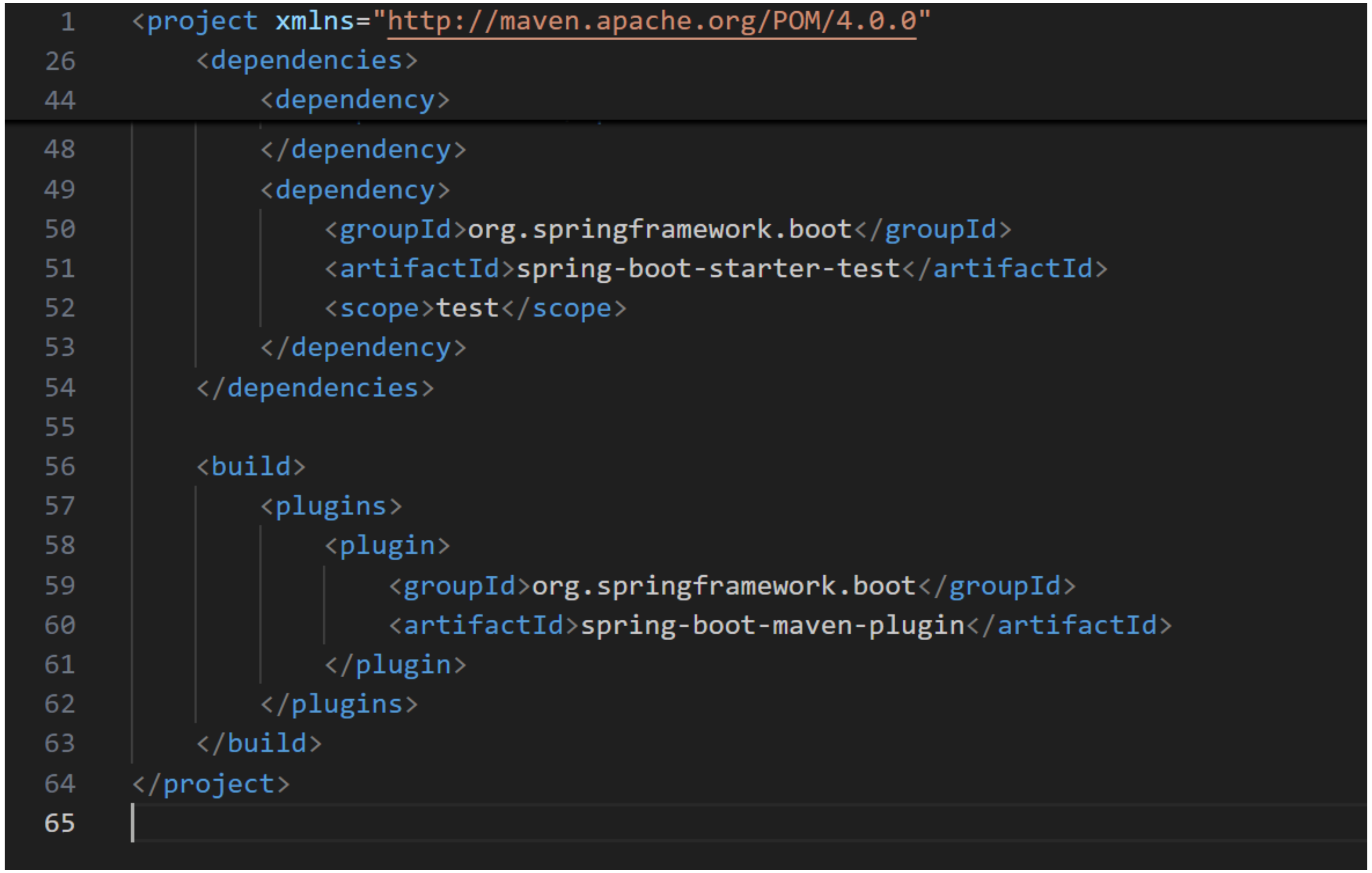
pom.xml :



```xml
<project xmlns="http://maven.apache.org/POM/4.0.0"
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-data-jpa</artifactId>
        </dependency>
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <scope>runtime</scope>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-thymeleaf</artifactId>
        </dependency>
        <dependency>
            <groupId>org.projectlombok</groupId>
            <artifactId>lombok</artifactId>
            <optional>true</optional>
        </dependency>
```

Project-2: Employee Management System

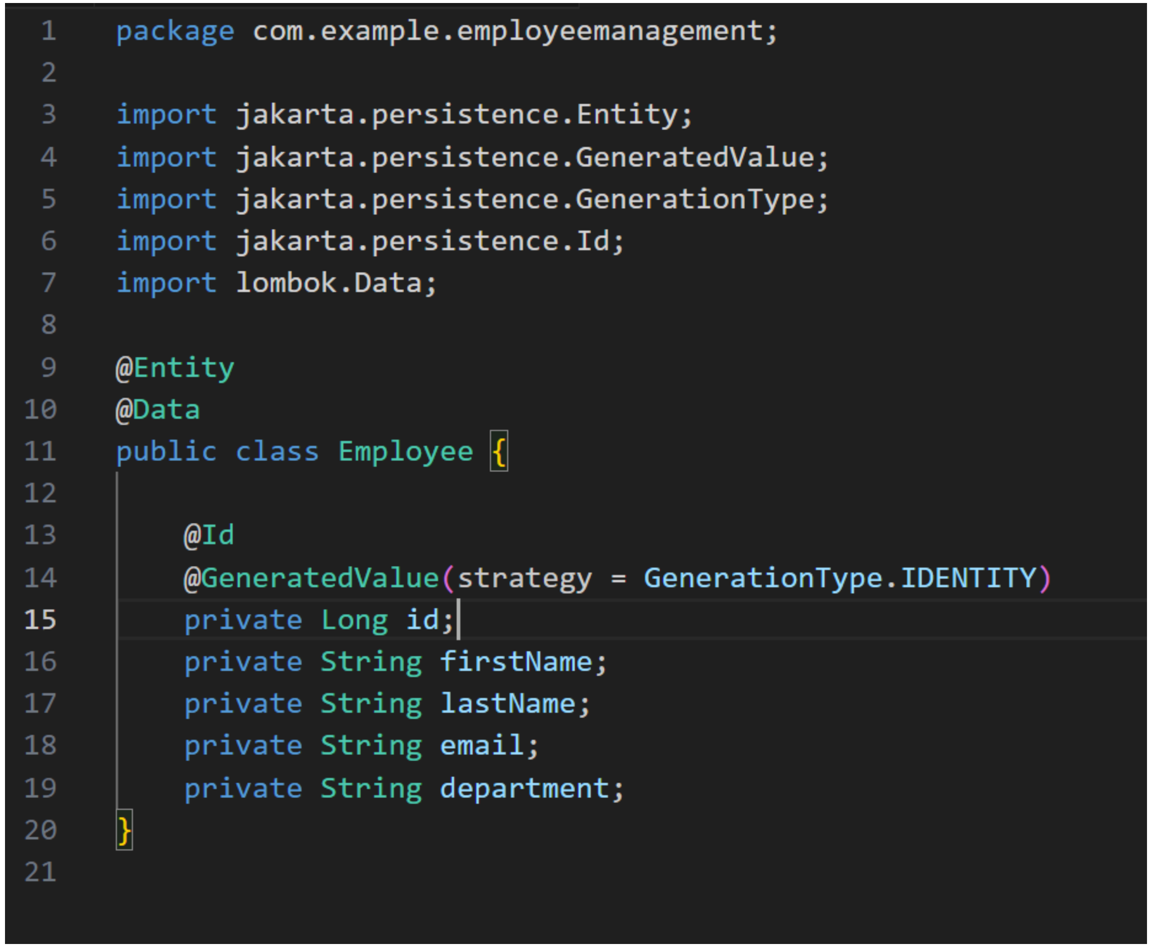
pom.xml :



```xml
<project xmlns="http://maven.apache.org/POM/4.0.0"
    <dependencies>
        <dependency>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>
</project>
```

Project-2: Employee Management System

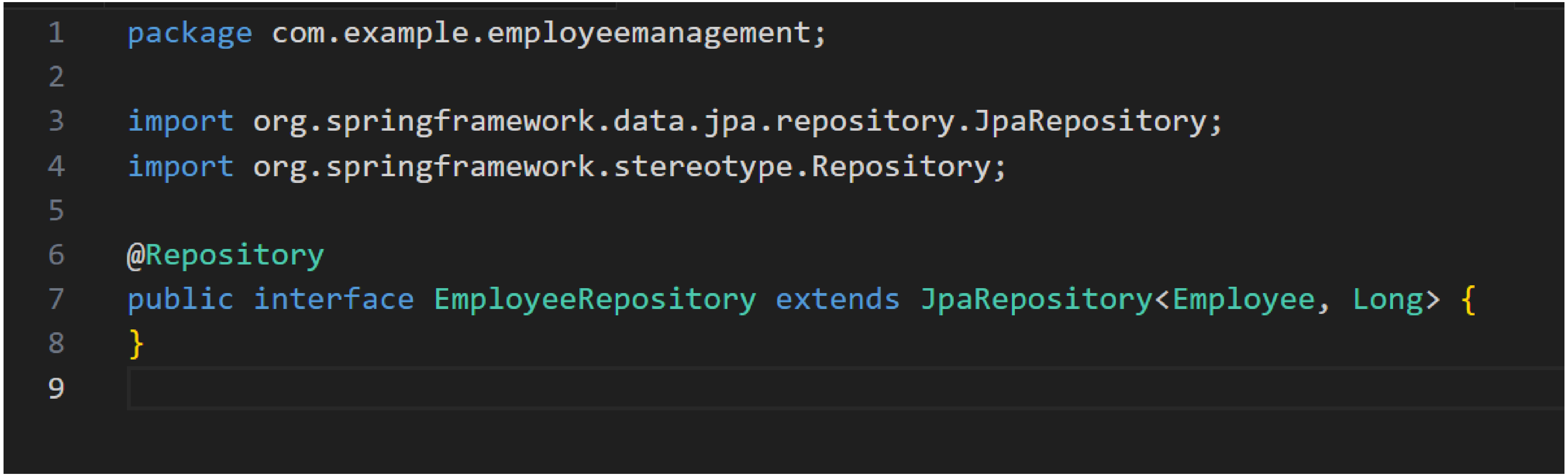
Employee.java



```java
package com.example.employeemanagement;

import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import lombok.Data;

@Entity
@Data
public class Employee {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String firstName;
    private String lastName;
    private String email;
    private String department;
}

```

Project-2: Employee Management System

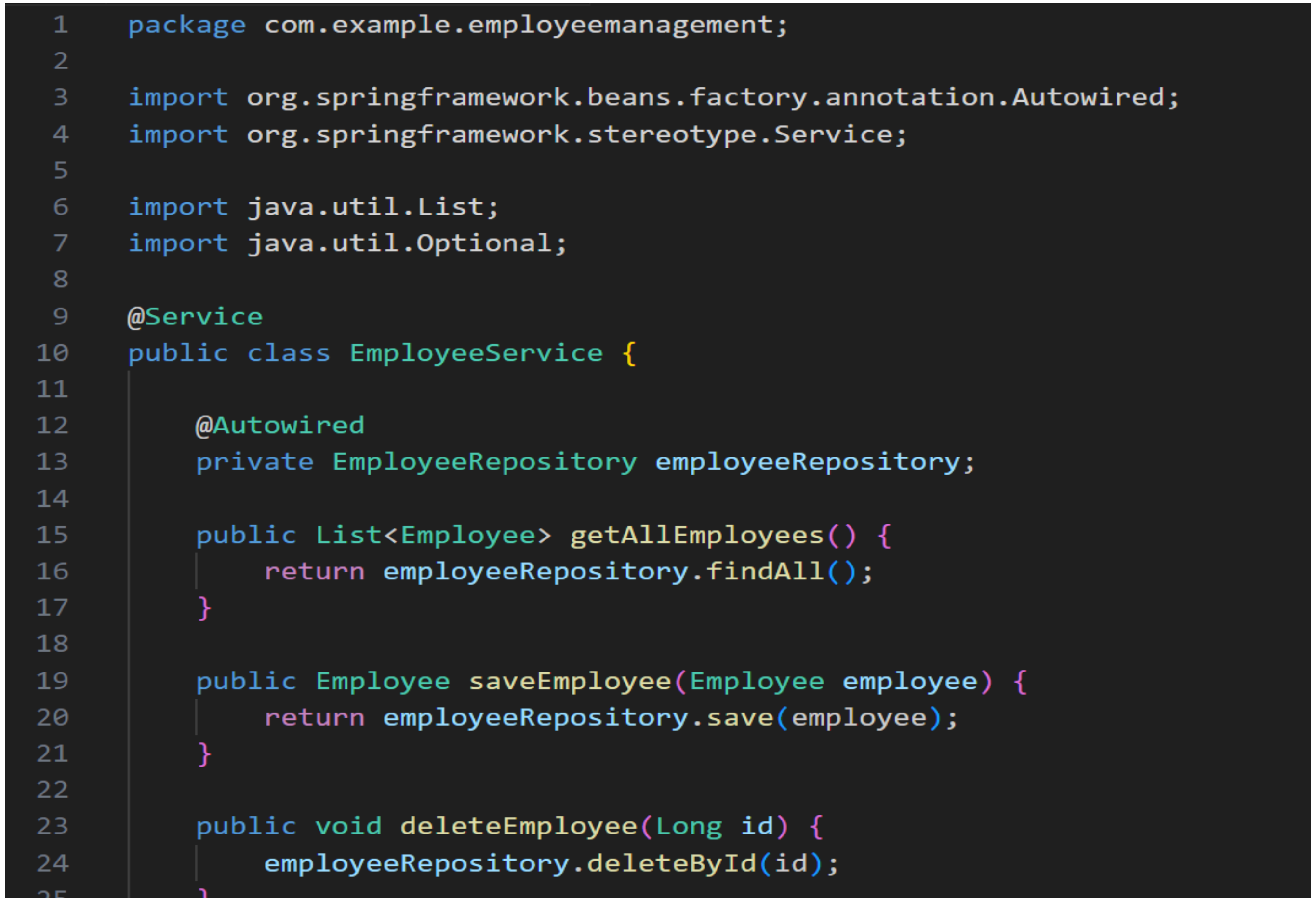
EmployeeRepository.java



```java
package com.example.employeemanagement;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

@Repository
public interface EmployeeRepository extends JpaRepository<Employee, Long> {
}

```

Project-2: Employee Management System

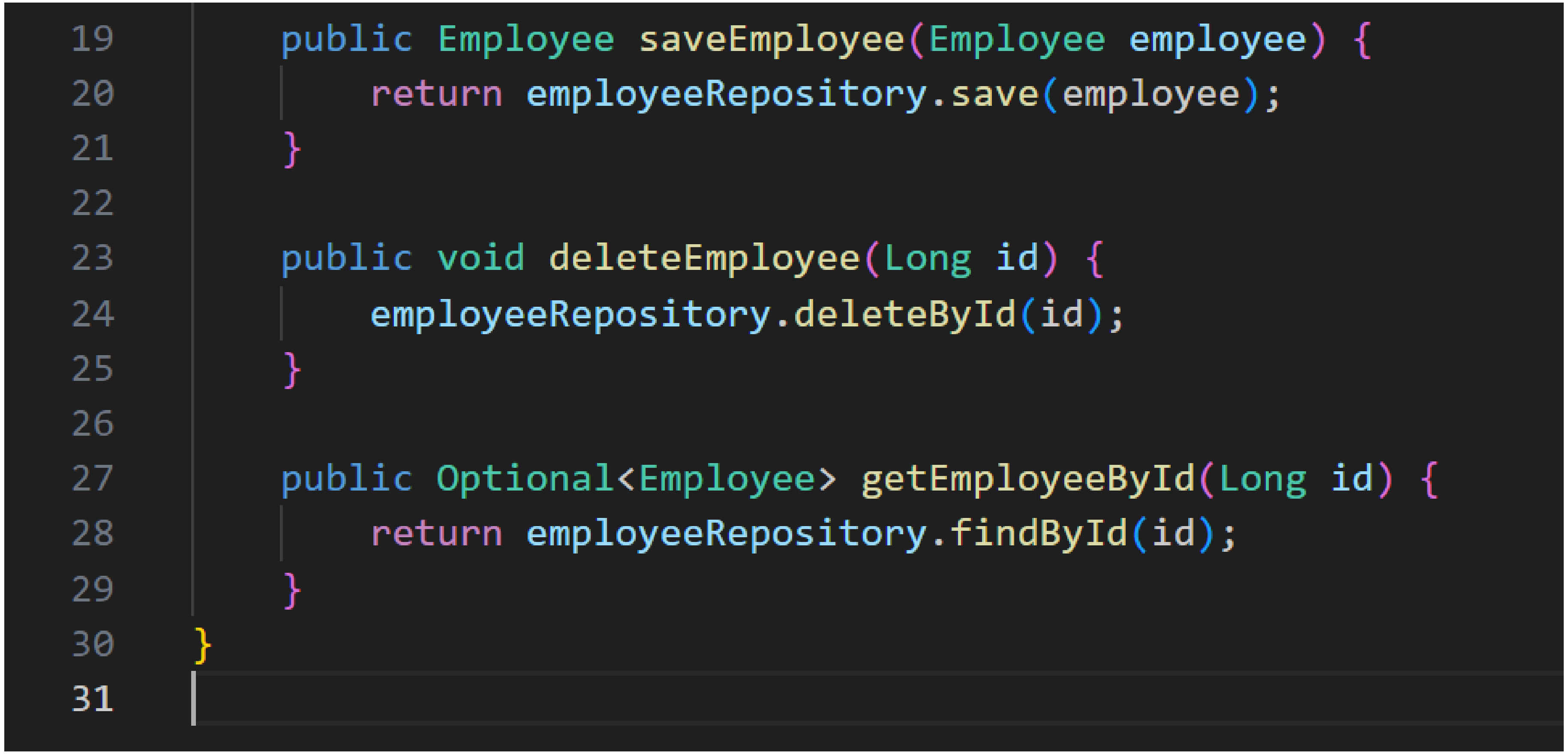
EmployeeService.java



```java
package com.example.employeemanagement;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;
import java.util.Optional;

@Service
public class EmployeeService {

    @Autowired
    private EmployeeRepository employeeRepository;

    public List<Employee> getAllEmployees() {
        return employeeRepository.findAll();
    }

    public Employee saveEmployee(Employee employee) {
        return employeeRepository.save(employee);
    }

    public void deleteEmployee(Long id) {
        employeeRepository.deleteById(id);
    }
```

Project-2: Employee Management System

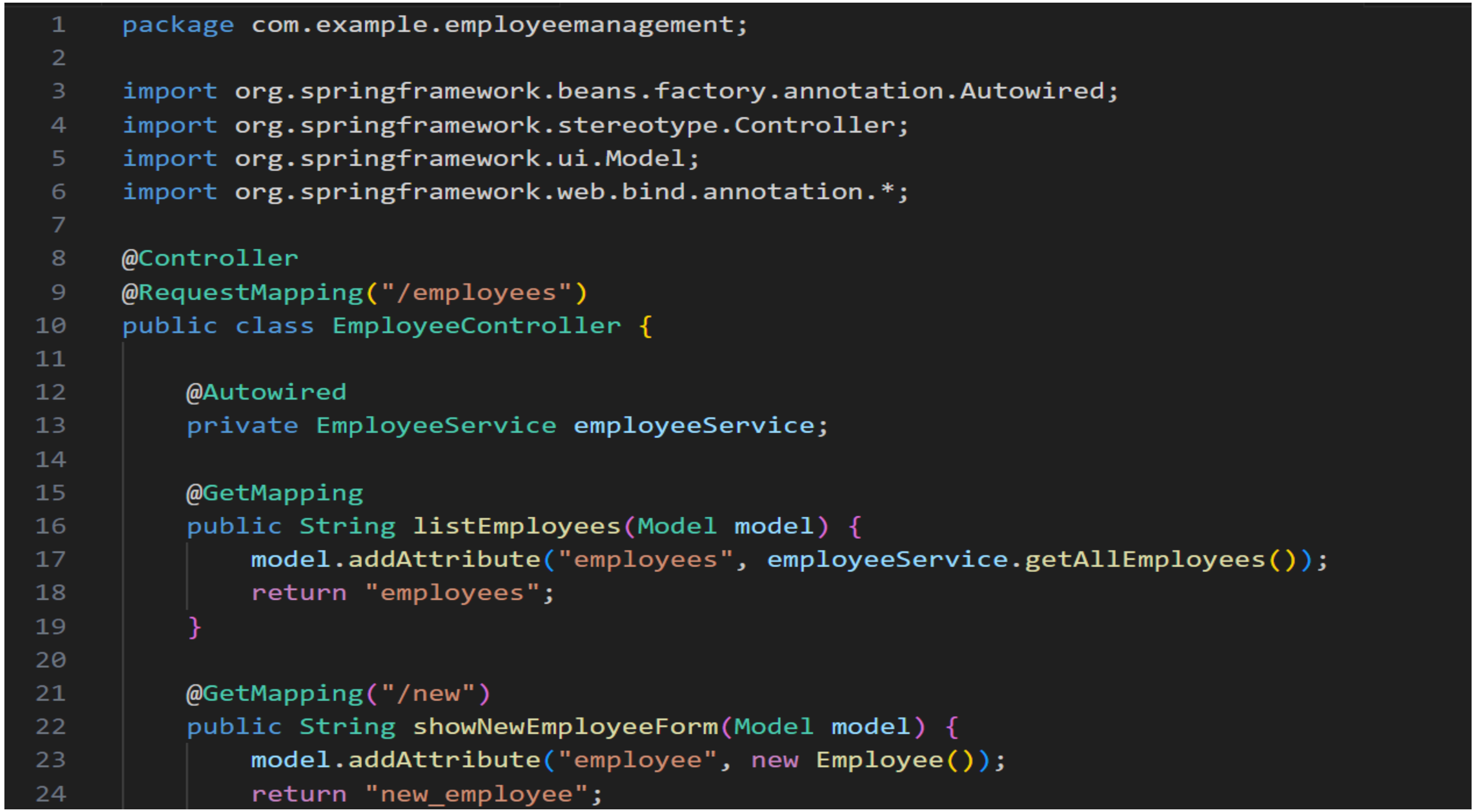
EmployeeService.java



```java
    public Employee saveEmployee(Employee employee) {
        return employeeRepository.save(employee);
    }

    public void deleteEmployee(Long id) {
        employeeRepository.deleteById(id);
    }

    public Optional<Employee> getEmployeeById(Long id) {
        return employeeRepository.findById(id);
    }
}

```

Project-2: Employee Management System

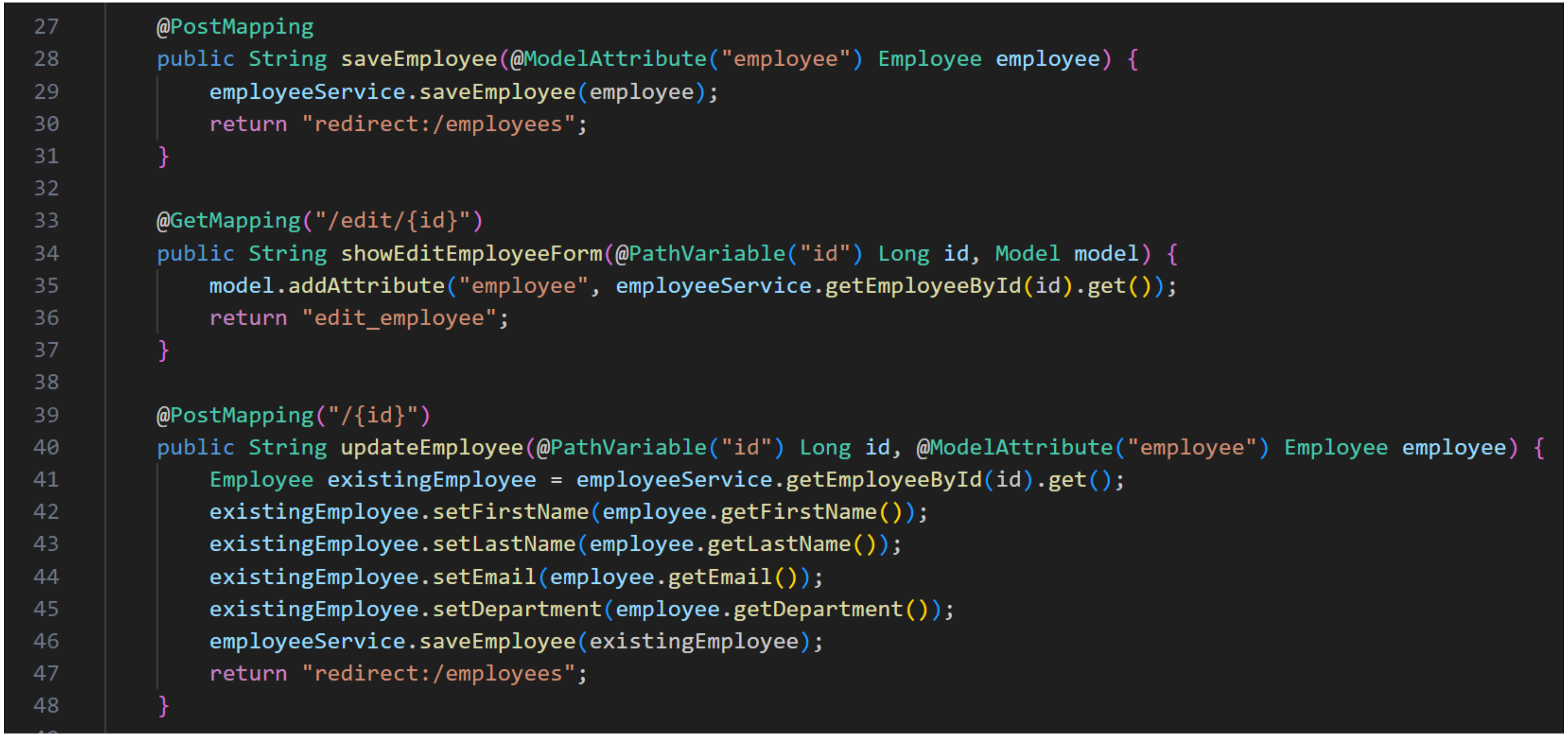
EmployeeController.java



```java
package com.example.employeemanagement;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.*;

@Controller
@RequestMapping("/employees")
public class EmployeeController {

    @Autowired
    private EmployeeService employeeService;

    @GetMapping
    public String listEmployees(Model model) {
        model.addAttribute("employees", employeeService.getAllEmployees());
        return "employees";
    }

    @GetMapping("/new")
    public String showNewEmployeeForm(Model model) {
        model.addAttribute("employee", new Employee());
        return "new_employee";
```

Project-2: Employee Management System

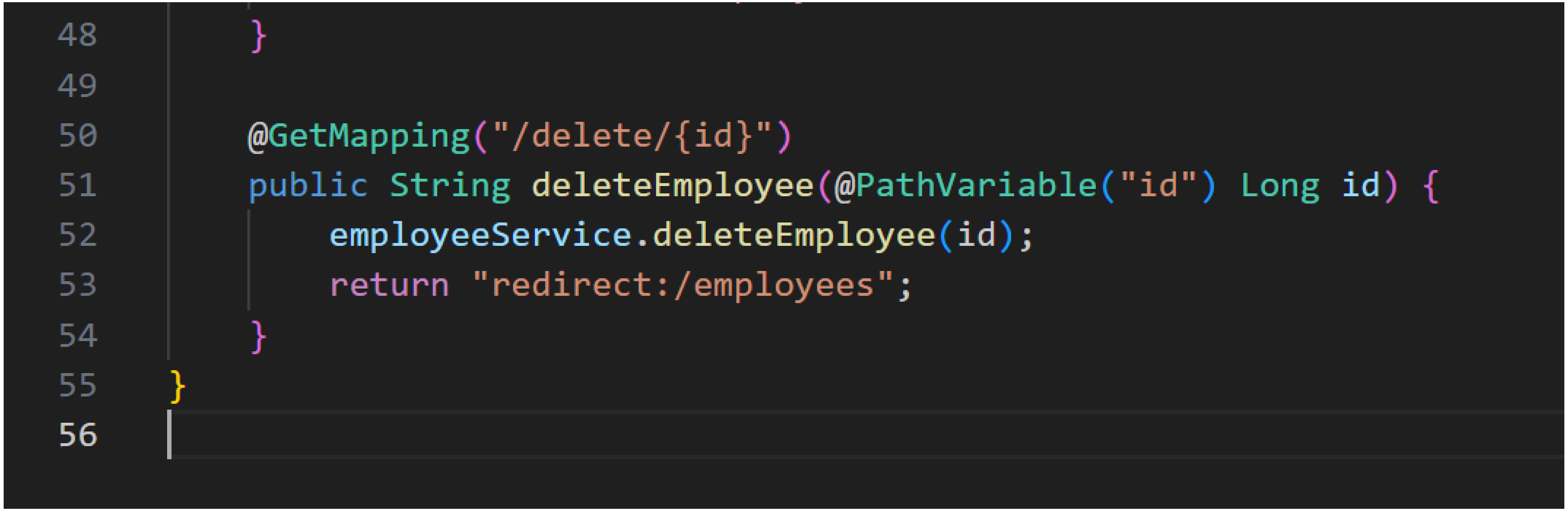
EmployeeController.java



```java
    @PostMapping
    public String saveEmployee(@ModelAttribute("employee") Employee employee) {
        employeeService.saveEmployee(employee);
        return "redirect:/employees";
    }

    @GetMapping("/edit/{id}")
    public String showEditEmployeeForm(@PathVariable("id") Long id, Model model) {
        model.addAttribute("employee", employeeService.getEmployeeById(id).get());
        return "edit_employee";
    }

    @PostMapping("/{id}")
    public String updateEmployee(@PathVariable("id") Long id, @ModelAttribute("employee") Employee employee) {
        Employee existingEmployee = employeeService.getEmployeeById(id).get();
        existingEmployee.setFirstName(employee.getFirstName());
        existingEmployee.setLastName(employee.getLastName());
        existingEmployee.setEmail(employee.getEmail());
        existingEmployee.setDepartment(employee.getDepartment());
        employeeService.saveEmployee(existingEmployee);
        return "redirect:/employees";
    }
```

Project-2: Employee Management System

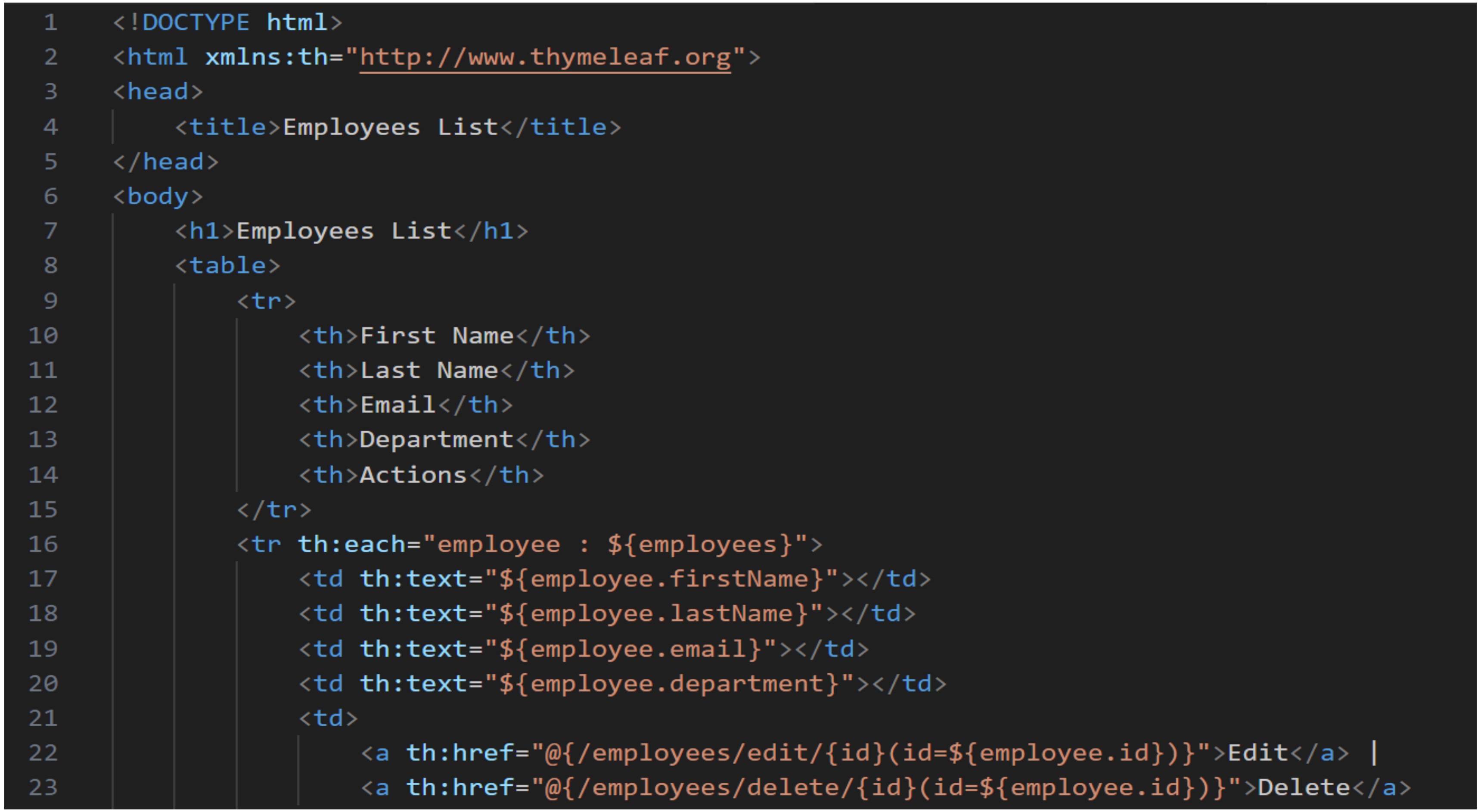
EmployeeController.java



```java
    }

    @GetMapping("/delete/{id}")
    public String deleteEmployee(@PathVariable("id") Long id) {
        employeeService.deleteEmployee(id);
        return "redirect:/employees";
    }
}
```

Project-2: Employee Management System

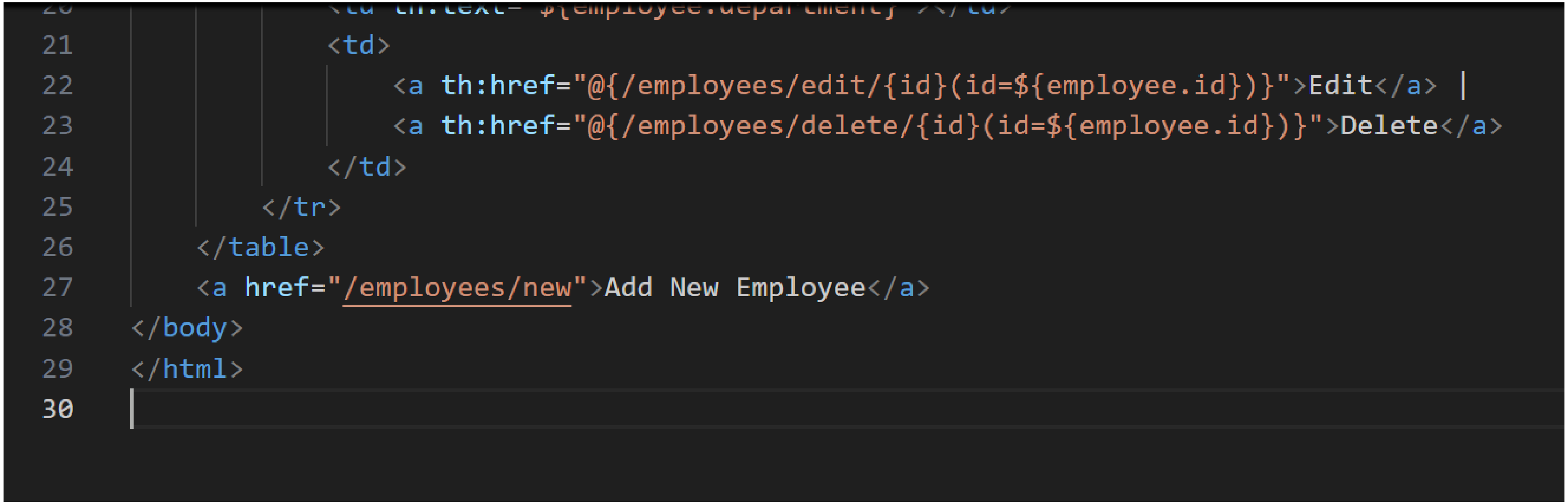
Employees.html



```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <title>Employees List</title>
</head>
<body>
    <h1>Employees List</h1>
    <table>
        <tr>
            <th>First Name</th>
            <th>Last Name</th>
            <th>Email</th>
            <th>Department</th>
            <th>Actions</th>
        </tr>
        <tr th:each="employee : ${employees}">
            <td th:text="${employee.firstName}"></td>
            <td th:text="${employee.lastName}"></td>
            <td th:text="${employee.email}"></td>
            <td th:text="${employee.department}"></td>
            <td>
                <a th:href="@{/employees/edit/{id}(id=${employee.id})}">Edit</a> |
                <a th:href="@{/employees/delete/{id}(id=${employee.id})}">Delete</a>
```

Project-2: Employee Management System

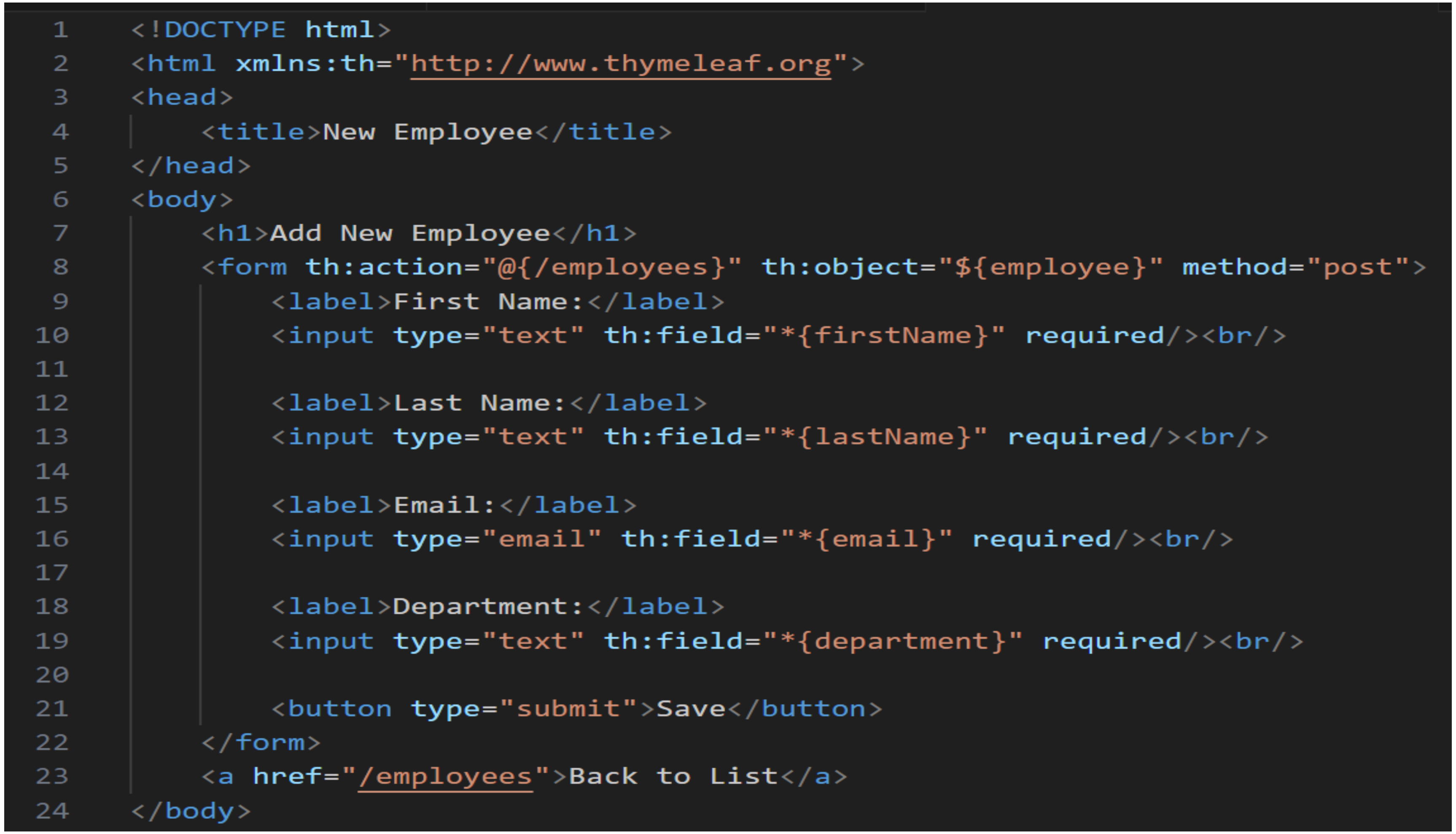
Employees.html



```
            <td th:text="${employee.department}"></td>
            <td>
                <a th:href="@{/employees/edit/{id}(id=${employee.id})}">Edit</a> |
                <a th:href="@{/employees/delete/{id}(id=${employee.id})}">Delete</a>
            </td>
        </tr>
    </table>
    <a href="/employees/new">Add New Employee</a>
</body>
</html>

```

Project-2: Employee Management System

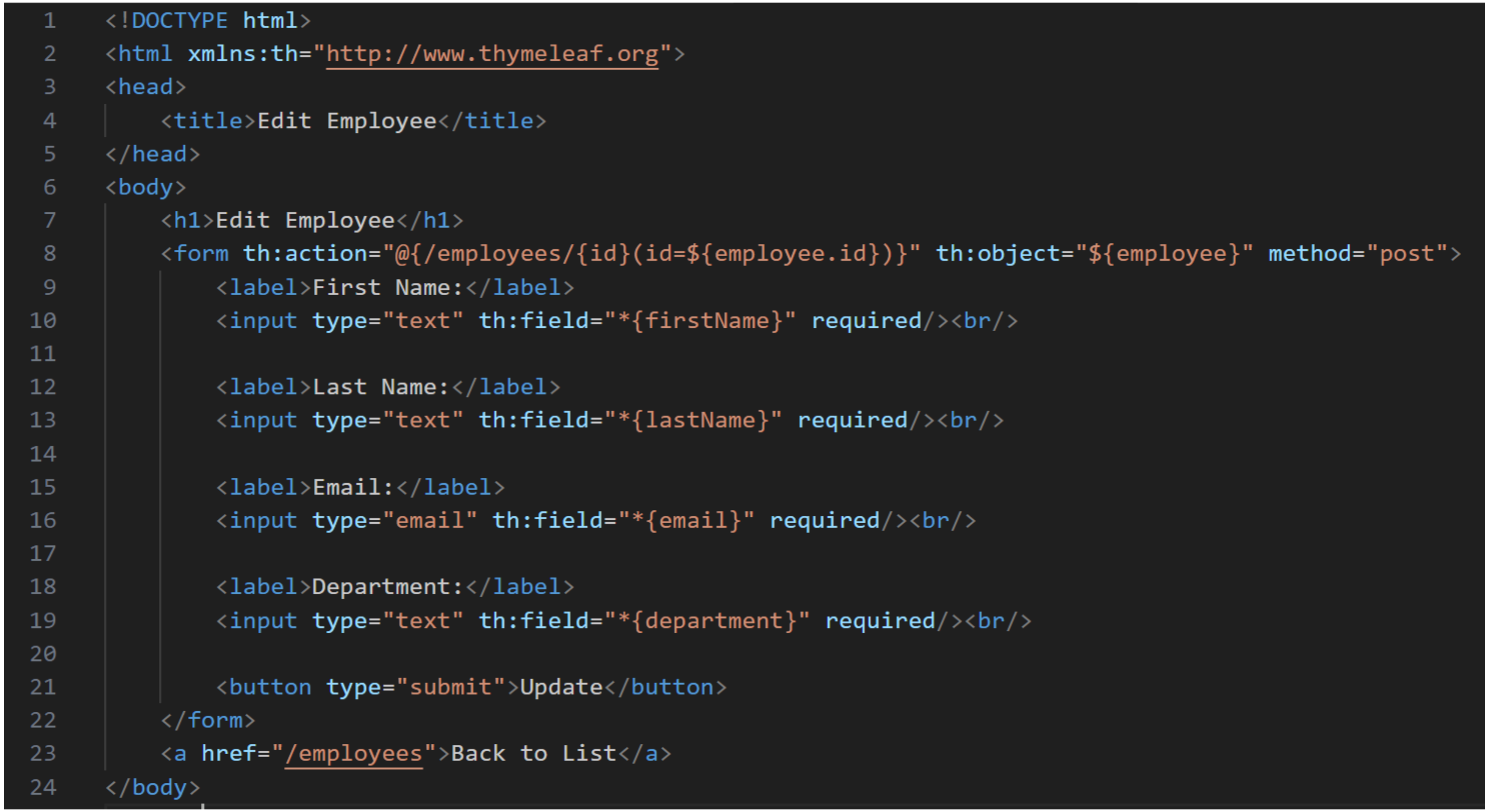
New_employees.html



```html
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <title>New Employee</title>
</head>
<body>
    <h1>Add New Employee</h1>
    <form th:action="@{/employees}" th:object="${employee}" method="post">
        <label>First Name:</label>
        <input type="text" th:field="*{firstName}" required/><br/>

        <label>Last Name:</label>
        <input type="text" th:field="*{lastName}" required/><br/>

        <label>Email:</label>
        <input type="email" th:field="*{email}" required/><br/>

        <label>Department:</label>
        <input type="text" th:field="*{department}" required/><br/>

        <button type="submit">Save</button>
    </form>
    <a href="/employees">Back to List</a>
</body>
```

Project-2: Employee Management System

edit_employees.html



```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <title>Edit Employee</title>
</head>
<body>
    <h1>Edit Employee</h1>
    <form th:action="@{/employees/{id}(id=${employee.id})}" th:object="${employee}" method="post">
        <label>First Name:</label>
        <input type="text" th:field="*{firstName}" required/><br/>

        <label>Last Name:</label>
        <input type="text" th:field="*{lastName}" required/><br/>

        <label>Email:</label>
        <input type="email" th:field="*{email}" required/><br/>

        <label>Department:</label>
        <input type="text" th:field="*{department}" required/><br/>

        <button type="submit">Update</button>
    </form>
    <a href="/employees">Back to List</a>
</body>
```

Project-2: Employee Management System

- **Conclusion:**

- The Employee Management System effectively simplifies the management of employee information by utilizing Spring Boot, MySQL, and Thymeleaf. It offers a user-friendly interface for efficient CRUD operations, enhances data accuracy, and secures sensitive information. The application improves organizational efficiency and supports better decision-making through accessible employee data, showcasing the value of web applications in optimizing HR processes.