**Pom.xml**

<?xml version="1.0" encoding="UTF-8"?>  
<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 https://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
 <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>3.4.5</version>  
 <relativePath/> <!-- lookup parent from repository -->  
 </parent>  
 <groupId>com.example</groupId>  
 <artifactId>taskdemo</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <name>taskdemo</name>  
 <description>Demo project for Spring Boot</description>  
 <url/>  
 <licenses>  
 <license/>  
 </licenses>  
 <developers>  
 <developer/>  
 </developers>  
 <scm>  
 <connection/>  
 <developerConnection/>  
 <tag/>  
 <url/>  
 </scm>  
 <properties>  
 <java.version>17</java.version>  
 </properties>  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-data-jpa</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-data-mongodb</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </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>

**Employee.java**

package com.example.taskdemo.model;  
  
import org.springframework.data.annotation.Id;  
import org.springframework.data.mongodb.core.mapping.Document;  
  
@Document(collection = "employees")  
public class Employee {  
  
 @Id  
 private String id;  
 private String name;  
 private String position;  
 private double salary;  
  
 // Constructors  
 public Employee() {}  
  
 public Employee(String name, String position, double salary) {  
 this.name = name;  
 this.position = position;  
 this.salary = salary;  
 }  
  
 // Getters and Setters  
 public String getId() {  
 return id;  
 }  
  
 public void setId(String id) {  
 this.id = id;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public String getPosition() {  
 return position;  
 }  
  
 public void setPosition(String position) {  
 this.position = position;  
 }  
  
 public double getSalary() {  
 return salary;  
 }  
  
 public void setSalary(double salary) {  
 this.salary = salary;  
 }  
}

**EmployeeRepositry.java**

package com.example.taskdemo.repository;  
  
import com.example.taskdemo.model.Employee;  
import org.springframework.data.mongodb.repository.MongoRepository;  
  
public interface EmployeeRepository extends MongoRepository<Employee, String> {  
}

**EmployeeController.java**

package com.example.taskdemo.controller;  
  
import com.example.taskdemo.model.Employee;  
import com.example.taskdemo.repository.EmployeeRepository;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.\*;  
  
import java.util.List;  
import java.util.Optional;  
  
@RestController  
@RequestMapping("/api/employees")  
@CrossOrigin(origins = "\*") // For frontend CORS access  
public class EmployeeController {  
  
 @Autowired  
 private EmployeeRepository repository;  
  
 @GetMapping  
 public List<Employee> getAllEmployees() {  
 return repository.findAll();  
 }  
  
 @GetMapping("/{id}")  
 public Optional<Employee> getEmployeeById(@PathVariable String id) {  
 return repository.findById(id);  
 }  
  
 @PostMapping  
 public Employee addEmployee(@RequestBody Employee employee) {  
 return repository.save(employee);  
 }  
  
 @PutMapping("/{id}")  
 public Employee updateEmployee(@PathVariable String id, @RequestBody Employee updatedEmployee) {  
 updatedEmployee.setId(id);  
 return repository.save(updatedEmployee);  
 }  
  
 @DeleteMapping("/{id}")  
 public void deleteEmployee(@PathVariable String id) {  
 repository.deleteById(id);  
 }  
}

**Index.html**

<!DOCTYPE html>  
<html xmlns:th="http://www.thymeleaf.org">  
<head>  
 <meta charset="UTF-8">  
 <title>Employee Form</title>  
</head>  
<body>  
<h1>Enter Employee Details</h1>  
  
<form th:action="@{/save}" th:object="${employee}" method="post">  
 <label for="employeeId">Employee ID:</label>  
 <input type="text" id="employeeId" th:field="\*{employeeId}" /><br/><br/>  
  
 <label for="name">Name:</label>  
 <input type="text" id="name" th:field="\*{name}" /><br/><br/>  
  
 <label for="email">Email:</label>  
 <input type="email" id="email" th:field="\*{email}" /><br/><br/>  
  
 <label for="location">Location:</label>  
 <input type="text" id="location" th:field="\*{location}" /><br/><br/>  
  
 <button type="submit">Submit</button>  
</form>  
</body>  
</html>

**displayAll.html**

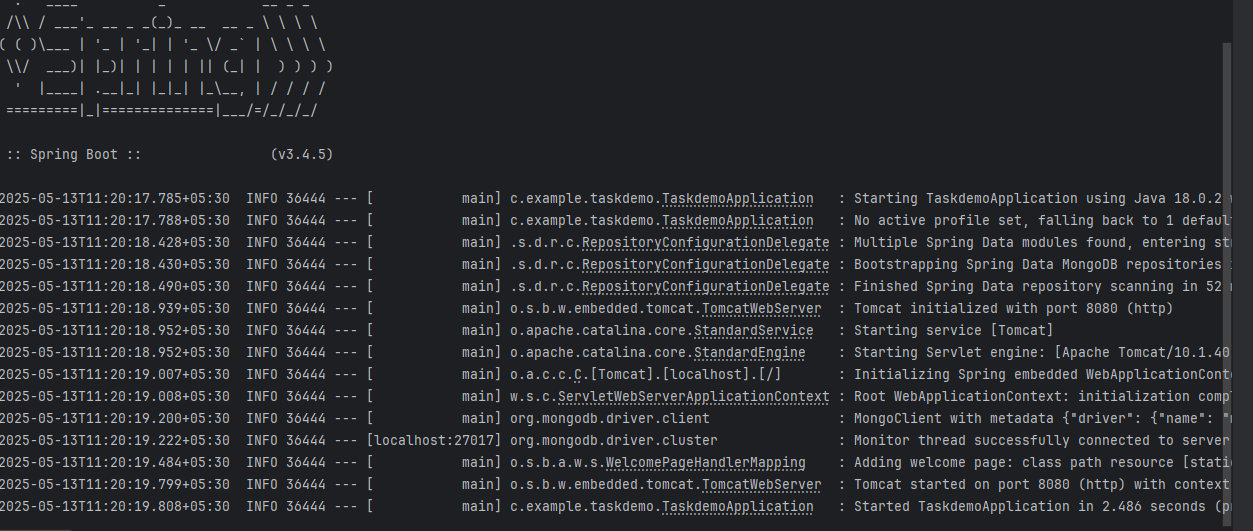
<!DOCTYPE html>  
<html xmlns:th="http://www.thymeleaf.org">  
<head>  
 <title>All Employees</title>  
</head>  
<body>  
<h1>Employee List</h1>  
<table border="1">  
 <tr>  
 <th>Employee ID</th><th>Name</th><th>Email</th><th>Location</th>  
 </tr>  
 <tr th:each="employee : ${employees}">  
 <td th:text="${employee.employeeId}"></td>  
 <td th:text="${employee.name}"></td>  
 <td th:text="${employee.email}"></td>  
 <td th:text="${employee.location}"></td>  
 </tr>  
</table>  
</body>  
</html>

**DisplayEmployee.html**

<!DOCTYPE html>  
<html xmlns:th="http://www.thymeleaf.org">  
<head>  
 <title>Employee Details</title>  
</head>  
<body>  
<h1>Employee Details</h1>  
<p><strong>Employee ID:</strong> <span th:text="${employee.employeeId}"></span></p>  
<p><strong>Name:</strong> <span th:text="${employee.name}"></span></p>  
<p><strong>Email:</strong> <span th:text="${employee.email}"></span></p>  
<p><strong>Location:</strong> <span th:text="${employee.location}"></span></p>  
</body>  
</html>

**EmployeeNotFound.html**

<!DOCTYPE html>  
<html xmlns:th="http://www.thymeleaf.org">  
<head>  
 <title>Employee Details</title>  
</head>  
<body>  
<h1>Employee Details</h1>  
<p><strong>Employee ID:</strong> <span th:text="${employee.employeeId}"></span></p>  
<p><strong>Name:</strong> <span th:text="${employee.name}"></span></p>  
<p><strong>Email:</strong> <span th:text="${employee.email}"></span></p>  
<p><strong>Location:</strong> <span th:text="${employee.location}"></span></p>  
</body>  
</html

****