22kq1a1219(IT)

S.chandini

(Mini project)

Abstract

The Employee Management System (EMS) is a comprehensive software application designed to manage employee information within an organization effectively. It automates various HR functions, from hiring and onboarding to performance tracking and payroll management. The EMS aims to improve operational efficiency, ensure data accuracy, and provide robust reporting capabilities to support strategic decision-making

Technologies Used

Java SE (Standard Edition): Core language for developing the application.

Spring Framework: For dependency injection, transaction management, and building RESTful web services.

Hibernate: For ORM (Object-Relational Mapping) to manage database operations.

MySQL: Database for storing employee information.

Apache Tomcat: Web server for deploying the application.

Maven: For project management and build automation.

Gathering:

\*\*Employee information Managaement:

Add, update, and delete employee records.

View detailed employee profiles.

\*\*Attendance and Leave Management:

Record daily attendance.

Apply for and approve leaves.

\*\* Payroll Management:

Calculate and manage employee salaries.

Generate pay slips and manage deductions.

\*\*Performance Management:

Manage access control to various features

The project

**//Appilication.java**

**package** edu.pace.obs1;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

**public** **class** Obs1Application {

**public** **static** **void** main(String[] args) {

SpringApplication.*run*(Obs1Application.**class**, args);

}

}

**//controller Package**

**package** edu.pace.obs1.controller;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.stereotype.Controller;

**import** org.springframework.ui.Model;

**import** org.springframework.web.bind.annotation.GetMapping;

**import** org.springframework.web.bind.annotation.ModelAttribute;

**import** org.springframework.web.bind.annotation.PathVariable;

**import** org.springframework.web.bind.annotation.PostMapping;

**import** edu.pace.obs1.model.Employee;

**import** edu.pace.obs1.services.EmployeeService;

@Controller

**public** **class** EmployeeController {

@Autowired

**private** EmployeeService employeeService;

// display list of employees

@GetMapping("/")

**public** String viewHomePage(Model model) {

model.addAttribute("listEmployees", employeeService.getAllEmployees());

**return** "index";

}

@GetMapping("/showNewEmployeeForm")

**public** String showNewEmployeeForm(Model model) {

// create model attribute to bind form data

Employee employee = **new** Employee();

model.addAttribute("employee", employee);

**return** "new\_employee";

}

@PostMapping("/saveEmployee")

**public** String saveEmployee(@ModelAttribute("employee") Employee employee) {

// save employee to database

employeeService.saveEmployee(employee);

**return** "redirect:/";

}

@GetMapping("/showFormForUpdate/{id}")

**public** String showFormForUpdate(@PathVariable(value = "id") **long** id, Model model) {

// get employee from the service

Employee employee = employeeService.getEmployeeById(id);

// set employee as a model attribute to pre-populate the form

model.addAttribute("employee", employee);

**return** "update\_employee";

}

@GetMapping("/deleteEmployee/{id}")

**public** String deleteEmployee(@PathVariable(value = "id") **long** id) {

// call delete employee method

**this**.employeeService.deleteEmployeeById(id);

**return** "redirect:/";

}

}

**//Moddel Package**

**package** edu.pace.obs1.model;

**import** jakarta.persistence.\*;

@Entity

@Table(name = "emp")

**public** **class** Employee {

@Id

@GeneratedValue(strategy = GenerationType.***IDENTITY***)

**private** **long** id;

@Column(name = "first\_name")

**private** String firstName;

@Column(name = "last\_name")

**private** String lastName;

@Column(name = "email")

**private** String email;

**public** **long** getId() {

**return** id;

}

**public** **void** setId(**long** id) {

**this**.id = id;

}

**public** String getFirstName() {

**return** firstName;

}

**public** **void** setFirstName(String firstName) {

**this**.firstName = firstName;

}

**public** String getLastName() {

**return** lastName;

}

**public** **void** setLastName(String lastName) {

**this**.lastName = lastName;

}

**public** String getEmail() {

**return** email;

}

**public** **void** setEmail(String email) {

**this**.email = email;

}

}

**//Repository Package**

**package** edu.pace.obs1.repository;

**import** org.springframework.data.jpa.repository.

JpaRepository;

**import** org.springframework.stereotype.Repository;

**import** edu.pace.obs1.model.Employee;

@Repository

**public** **interface** EmployeeRepository

**extends** JpaRepository<Employee, Long>{

}

**//services package**

**package** edu.pace.obs1.services;

**import** java.util.List;

**import** edu.pace.obs1.model.Employee;

**public** **interface** EmployeeService {

List <Employee> getAllEmployees();

**void** saveEmployee(Employee employee);

Employee getEmployeeById(**long** id);

**void** deleteEmployeeById(**long** id);

}

**package** edu.pace.obs1.services;

**import** java.util.List;

**import** java.util.Optional;

**//services Package**

**import** org.springframework.beans.factory.

annotation.

@Autowired;

**import** org.springframework.stereotype.Service;

**import** edu.pace.obs1.model.Employee;

**import** edu.pace.obs1.repository.EmployeeRepository;

@Service

**public** **class** EmployeeServiceImpl **implements** EmployeeService {

@Autowired

**private** EmployeeRepository employeeRepository;

@Override

**public** List < Employee > getAllEmployees() {

**return** employeeRepository.findAll();

}

@Override

**public** **void** saveEmployee(Employee employee) {

**this**.employeeRepository.save(employee);

}

@Override

**public** Employee getEmployeeById(**long** id) {

Optional < Employee > optional =

employeeRepository.findById(id);

Employee employee = **null**;

**if** (optional.isPresent()) {

employee = optional.get();

} **else** {

**throw** **new** RuntimeException(" Employee not found for id :: " + id);

}

**return** employee;

}

@Override

**public** **void** deleteEmployeeById(**long** id) {

**this**.employeeRepository.deleteById(id);

}

}

**Index.html**

<!DOCTYPE html>

<html lang=*"en"* xmlns:th=*"http://www.thymeleaf.org"*>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Employee Management System</title>

<link rel=*"stylesheet"* href=*"https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"* integrity=*"sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"* crossorigin=*"anonymous"*>

</head>

<body >

<div class=*"container my-2"*>

<hr><hr> <h1 align =*"center"* ><font face=*"algerian"*size=*"8"* color=*"red"*>Employees Data</font></h1><hr><hr>

<a th:href=*"@{/showNewEmployeeForm}"* class=*"btn btn-primary btn-sm mb-3"*> Add Employee </a>

<table border=*"1"* class=*"table table-striped table-responsive-md"*>

<thead>

<tr>

<th><u>Employee First Name</th>

<th><u>Employee Last Name</th>

<th><u>Employee Email</th>

<th> Actions </th>

</tr>

</thead>

<tbody>

<tr th:each=*"employee : ${listEmployees}"*>

<td th:text=*"${employee.firstName}"*></td>

<td th:text=*"${employee.lastName}"*></td>

<td th:text=*"${employee.email}"*></td>

<td> <a th:href=*"@{/showFormForUpdate/{id}(id=${employee.id})}"* class=*"btn btn-primary"*>Update</a>

<a th:href=*"@{/deleteEmployee/{id}(id=${employee.id})}"* class=*"btn btn-danger"*>Delete</a>

</td>

</tr>

</tbody>

</table>

</div>

</body>

</html>

**New.html**

<!DOCTYPE html>

<html lang=*"en"* xmlns:th=*"http://www.thymeleaf.org"*>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Employee Management System</title>

<link rel=*"stylesheet"* href=*"https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"* integrity=*"sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"* crossorigin=*"anonymous"*>

</head>

<body >

<div class=*"container"*>

<h1 align=*"center"* ><font face=*"stensil"*><u>Employee Management System</font></h1>

<hr><hr>

<h2>Save Employee</h2>

<form action=*"#"* th:action=*"@{/saveEmployee}"* th:object=*"${employee}"* method=*"POST"*>

<input type=*"text"* th:field=*"\*{firstName}"* placeholder=*"Employee First Name"* class=*"form-control mb-4 col-4"*>

<input type=*"text"* th:field=*"\*{lastName}"* placeholder=*"Employee Last Name"* class=*"form-control mb-4 col-4"*>

<input type=*"text"* th:field=*"\*{email}"* placeholder=*"Employee Email"* class=*"form-control mb-4 col-4"*>

<button type=*"submit"* class=*"btn btn-info col-2"*> Save Employee</button>

</form>

<hr><hr>

<a th:href=*"@{/}"*> Back to Employee List</a>

</div>

</body>

</html>

**Update.html**

<!DOCTYPE html>

<html lang=*"en"* xmlns:th=*"http://www.thymeleaf.org"*>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Employee Management System</title>

<link rel=*"stylesheet"* href=*"https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"* integrity=*"sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"* crossorigin=*"anonymous"*>

</head>

<body>

<div class=*"container"*>

<h1>Employee Management System</h1>

<hr>

<h2>Update Employee</h2>

<form action=*"#"* th:action=*"@{/saveEmployee}"* th:object=*"${employee}"* method=*"POST"*>

<!-- Add hidden form field to handle update -->

<input type=*"hidden"* th:field=*"\*{id}"* />

<input type=*"text"* th:field=*"\*{firstName}"* class=*"form-control mb-4 col-4"*>

<input type=*"text"* th:field=*"\*{lastName}"* class=*"form-control mb-4 col-4"*>

<input type=*"text"* th:field=*"\*{email}"* class=*"form-control mb-4 col-4"*>

<button type=*"submit"* class=*"btn btn-info col-2"*> Update Employee</button>

</form>

<hr>

<a th:href=*"@{/}"*> Back to Employee List</a>

</div>

</body>

</html>

**Application.properties**

spring.application.name=obs1

spring.datasource.url=jdbc:mysql://localhost:3306/pace?useSSL=false&serverTimezone=UTC&useLegacyDatetimeCode=false

spring.datasource.username=root

spring.datasource.password=root

server.port=904

# Hibernate

# The SQL dialect makes Hibernate generate better SQL for the chosen database

spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQLDialect

# Hibernate ddl auto (create, create-drop, validate, update)

spring.jpa.hibernate.ddl-auto = update

logging.level.org.hibernate.SQL=DEBUG

logging.level.org.hibernate.type=TRACE

output

spring.application.name=obs1

spring.datasource.url=jdbc:mysql://localhost:3306/pace?useSSL=false&serverTimezone=UTC&useLegacyDatetimeCode=false

spring.datasource.username=root

spring.datasource.password=root

server.port=904

# Hibernate

# The SQL dialect makes Hibernate generate better SQL for the chosen database

spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQLDialect

# Hibernate ddl auto (create, create-drop, validate, update)

spring.jpa.hibernate.ddl-auto = update

logging.level.org.hibernate.SQL=DEBUG

logging.level.org.hibernate.type=TRACE

**Output**

