**Employee Management System**

**Name: Sahikh Sadakat Kamar Kamre Alam**

**Registration No: 12214906**

**School of Computer Science Engineering**

**Java Full Stack Winter PEP Project Report**

**Trainer Name: Mr. Nilesh Magar**



Contents

1. **Introduction**
2. **Objective And Goals**
3. **Project Functionality**
4. **Technologies Used**
5. **Project Functionality**
6. **Project Structure**
7. **System Features**
   1. **Dashboard**
   2. **View Employees**
   3. **Search Employee**
   4. **Add Employee**
   5. **Update Employee**
   6. **Delete Employee**
8. **Images of Project**
   1. **Dashboard**
   2. **View Employees**
   3. **Search Employee**
   4. **Add Employee**
   5. **Update Employee**
   6. **Delete Employee**
9. **Conclusion**
10. **Introduction**

The Employee Management System (EMS) is a web-based application designed to manage employee data efficiently. It is developed using **Spring Boot** for the backend, **MySQL** for database management, and **Angular** for the frontend, styled with **Bootstrap** for responsiveness and enhanced UI design. The system enables users to perform CRUD (Create, Read, Update, Delete) operations on employee data seamlessly.

1. **Objective and Goals**

The main objective of the Employee Management System is to streamline employee record management by automating various HR-related tasks. The goals of this project include:

* **Efficient Employee Data Management:** Ensure all employee records are stored in a structured and organized manner.
* **Simplified Data Access:** Enable quick retrieval of employee details using search and filter functionalities.
* **User-Friendly Interface:** Provide an intuitive interface that allows HR personnel to manage employee records effortlessly.
* **Automated CRUD Operations:** Allow users to create, read, update, and delete employee records with minimal effort.
* **Data Accuracy and Integrity:** Ensure that all employee data is consistently updated and maintained securely.
* **Scalability and Performance:** Develop a system that can handle increasing employee records efficiently.

1. **Project Functionality**

The Employee Management System (EMS) provides a full-fledged solution for managing employee data. The main functionalities include:

* **User Authentication and Role-Based Access Control:** Secure login for administrators and HR personnel.
* **CRUD Operations:** The ability to add, view, update, and delete employee records efficiently.
* **Dashboard Analytics:** Insights into employee statistics, including total employees, average salary, and age distribution.
* **Real-time Data Updates:** Whenever an employee record is modified, the changes are immediately reflected in the system.
* **Search and Filter Options:** Enables searching employees by their ID and filtering records based on criteria such as age and salary.
* **Interactive User Interface:** A responsive and user-friendly design that enhances usability.

1. **Technologies Used**

* **Spring Boot (Backend):** A lightweight framework based on Java that simplifies backend development. It provides built-in features such as dependency management, embedded servers, and RESTful APIs, making it ideal for microservices architecture.
* **MySQL (Database):** A popular relational database management system (RDBMS) that efficiently stores and manages employee records. It supports SQL queries to retrieve, insert, update, and delete data securely.
* **Angular (Frontend):** A TypeScript-based frontend framework developed by Google. It enables the creation of dynamic and responsive user interfaces with components, two-way data binding, and dependency injection.
* **Bootstrap (Styling):** A CSS framework that provides pre-designed components and responsive layouts to enhance the user experience. It ensures the application is mobile-friendly and visually appealing.
* **Server Ports:**
  + Spring Boot runs on port **9090**, handling API requests and database operations.
  + Angular application runs on **localhost:4200**, serving the user interface for the system.

1. **Project Functionality**

The Employee Management System (EMS) provides a full-fledged solution for managing employee data. The main functionalities include:

* User Authentication and Role-Based Access Control: Secure login for administrators and HR personnel.
* CRUD Operations: The ability to add, view, update, and delete employee records efficiently.
* Dashboard Analytics: Insights into employee statistics, including total employees, average salary, and age distribution.
* Real-time Data Updates: Whenever an employee record is modified, the changes are immediately reflected in the system.
* Search and Filter Options: Enables searching employees by their ID and filtering records based on criteria such as age and salary.
* Interactive User Interface: A responsive and user-friendly design that enhances usability.

1. **Project Structure**

The Employee Management System follows an organized structure for seamless development and scalability. The structure is as follows:

Backend (Spring Boot)

* src/main/java/com/ems/
  + controller/ - Contains REST API controllers for handling HTTP requests.
  + service/ - Includes business logic and service layer for employee management.
  + repository/ - Interfaces that interact with the MySQL database using JPA.
  + model/ - Defines entity classes representing database tables.
* src/main/resources/
  + application.properties - Configuration file for database connection and server properties.

Frontend (Angular)

* src/app/
  + components/ - Contains reusable UI components such as forms and tables.
  + services/ - Handles API requests and data fetching from the backend.
  + pages/ - Defines different page views like Dashboard, View Employees, Add Employee, etc.
* src/assets/ - Stores static files like images and stylesheets.
* angular.json - Angular project configuration file.

1. **System Features**

The system consists of a **navigation bar** with six main components:

1. **Dashboard**
2. **View Employees**
3. **Search Employee**
4. **Add Employee**
5. **Update Employee**
6. **Delete Employee**

**7.1) Dashboard**

The dashboard provides an overview of employee data with the following key metrics:

* **Total Number of Employees:** Displays the total count of employees in the system.
* **Average Salary of Employees:** Calculates and displays the average salary of all employees in the database.
* **Age Distribution Graph:** A graphical representation of employees’ ages, helping in understanding the workforce demographics.

**7.2) View Employees**

This section displays all employee records stored in the database. The displayed attributes include:

* **Employee ID (Eid):** A unique identifier for each employee.
* **First Name and Last Name:** The full name of the employee.
* **Age:** The age of the employee.
* **Salary:** The employee’s monthly or annual salary.
* **Action Buttons:**
  + **Edit:** Clicking this button opens a form pre-filled with the employee’s details, allowing modifications.
  + **Delete:** Clicking this button removes the employee from the system after confirmation.

**7.3) Search Employee**

* Employees can be searched by their **Employee ID**.
* Once found, the details of the employee (same as those in the View Employees section) are displayed.
* The record includes action buttons for **editing** and **deleting** the employee’s details, allowing quick modifications or removal.

**7.4) Add Employee**

This section allows users to add new employees by providing the following details:

* **Employee ID:** A unique number assigned to the employee.
* **First Name & Last Name:** The employee’s name.
* **Age:** The age of the employee.
* **Salary:** The salary of the employee.
* **Process:** Once all required fields are filled, clicking the **Add Employee** button sends the data to the database, and the updated list is immediately reflected in the system.

**7.5) Update Employee**

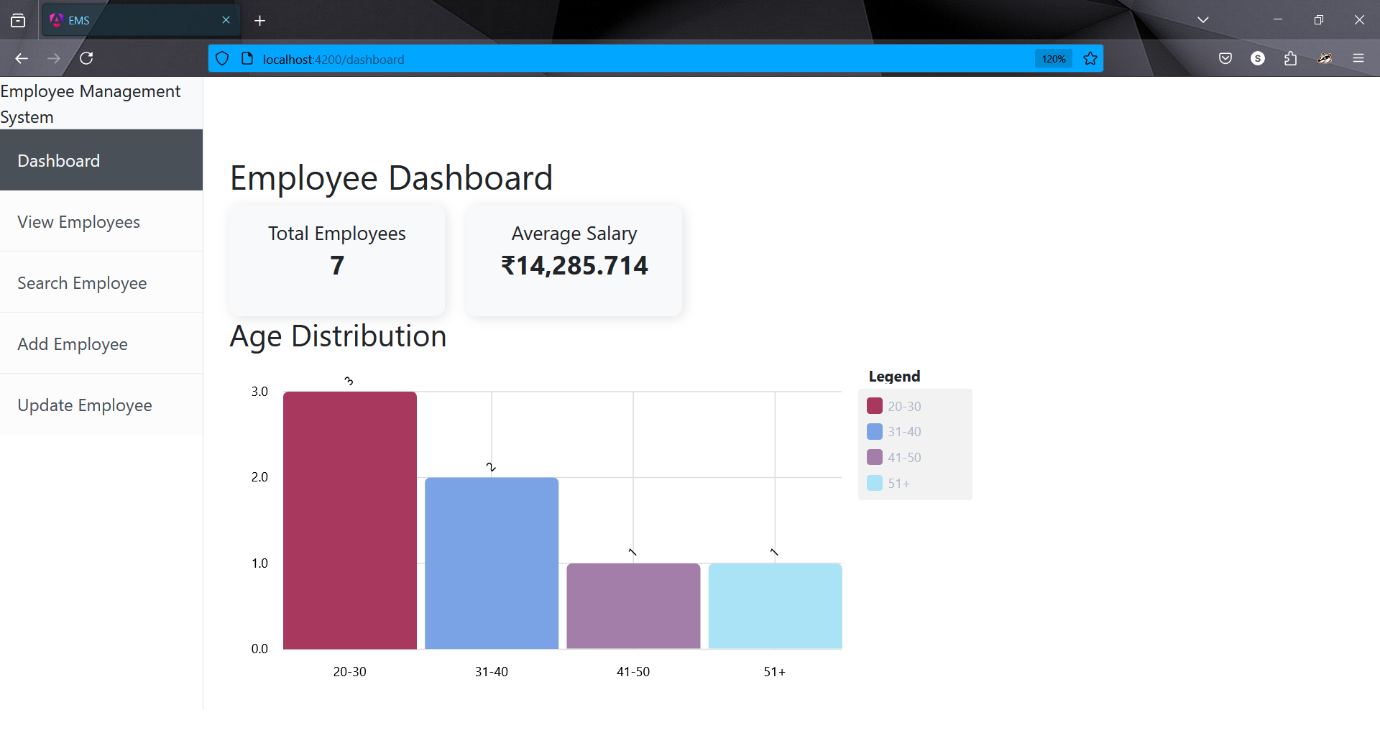
* Users can update an existing employee's details using the same input fields as the Add Employee section.
* After entering the updated details, clicking the **Update** button modifies the respective employee’s data in the database.
* The system validates the provided **Employee ID** to ensure the correct record is being updated.

**7.6) Delete Employee**

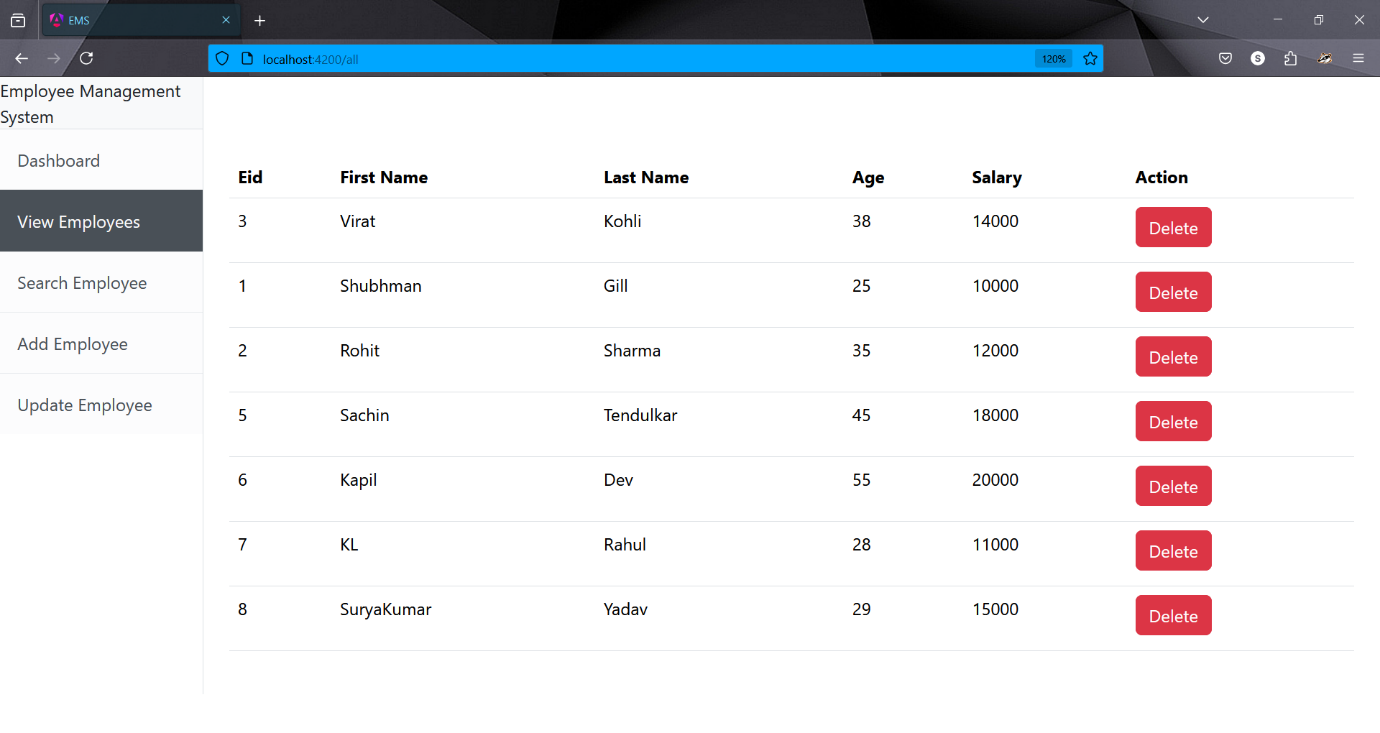
* This section allows users to remove an employee by entering their **Employee ID**.
* Clicking the **Delete** button triggers a confirmation prompt before the employee is permanently removed from the database.
* Once deleted, the record is no longer accessible from the View Employees or Search Employee sections.

1. **Images Of Project**

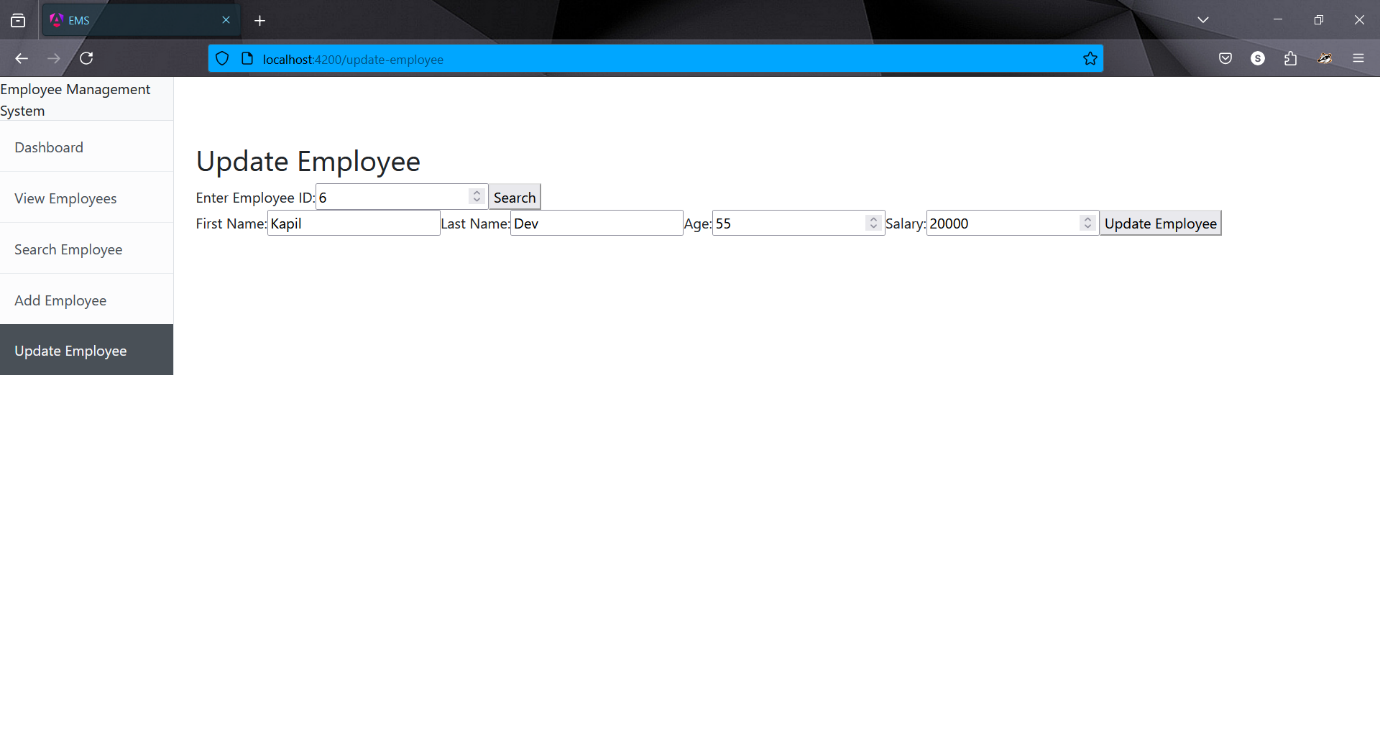
**8.1) Dashboard**



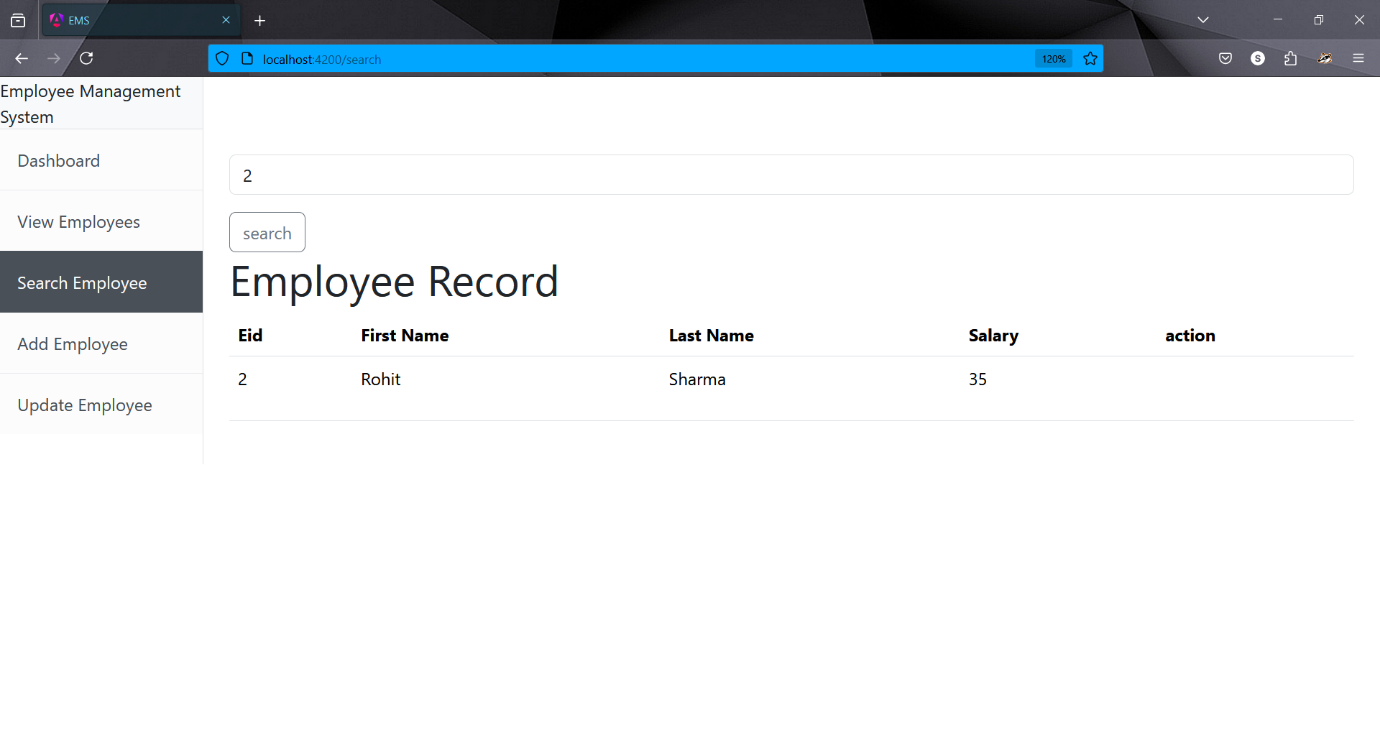
**8.2) View Employees**

****

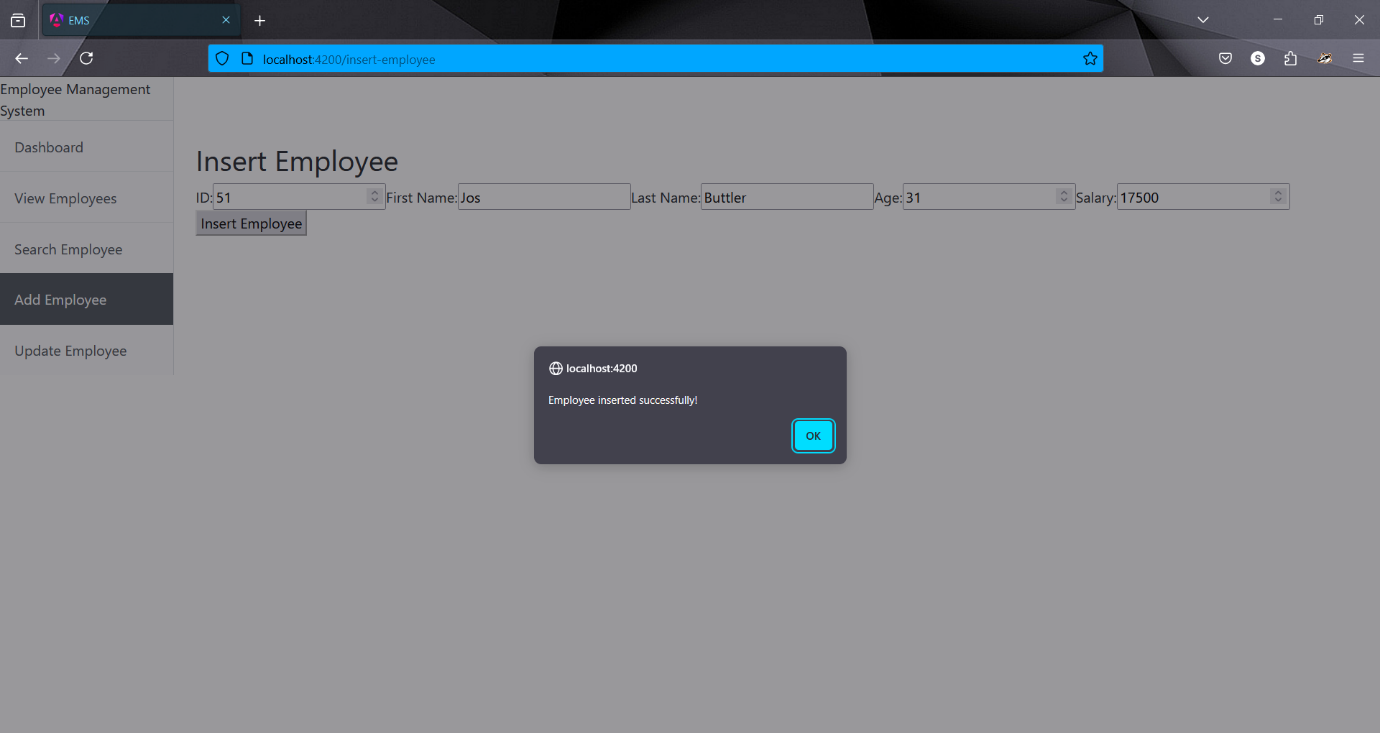
**8.3) Updated Values**

****

**8.4) Search Employee**

****

**8.5) Add Employee**

****

1. **Conclusion**

The Employee Management System provides a simple and efficient way to manage employee records. With its user-friendly interface and CRUD functionality, it enhances the overall efficiency of employee data management. The integration of **Spring Boot, MySQL, and Angular** ensures a robust and scalable system, making it ideal for enterprise-level applications.