Employee Management System

A PROJECT REPORT

Submitted by
Chandan Singh Rajput
12220324
Java Full Stack

Bachelor of Computer Science and Engineering

LOVELY PROFESSIONAL UNIVERSITY
PUNJAB (INDIA)

Acknowledgement

I would like to express my sincere gratitude to my university for providing me with the opportunity to undertake this project on **Employee Management System** as part of my academic curriculum. The support and encouragement from the faculty and administration have been invaluable throughout this journey.

I extend my heartfelt thanks to **Mr. Nilesh Magar**, my external mentor, for his guidance, valuable insights, and continuous support during the development of this project. His expertise and constructive feedback have greatly contributed to refining my work and enhancing my technical skills.

I am also grateful to **Hitbullseye** for providing me with resources and guidance that have helped me gain practical knowledge and hands-on experience in full-stack development. Their mentorship and technical assistance have been instrumental in shaping this project.

Furthermore, I would like to acknowledge the efforts of my professors and peers who have provided encouragement and valuable suggestions during the project's execution. Their feedback and discussions have played a significant role in improving my understanding of the subject matter.

Finally, I extend my deepest appreciation to my family and friends for their constant motivation and support throughout this project. Their encouragement has been a source of strength in completing this work successfully.

Abstract

The **Employee Management System (EMS)** is a web-based application designed to streamline the management of employee records. The system provides core functionalities such as adding, updating, searching, viewing, and deleting employee information through an intuitive user interface. Developed using **Angular for the frontend, Java Spring Boot for the backend, and MySQL as the database**, the EMS ensures efficient data handling and secure employee management.

The primary objective of this project is to enhance HR operations by offering a centralized platform for managing employee data. The system follows a **RESTful API architecture**, ensuring scalability and seamless integration with other applications. With a responsive user interface, a robust backend, and optimized database management, the EMS improves efficiency, reduces manual workload, and enhances data accuracy.

Future enhancements may include role-based authentication, attendance tracking, payroll management, and cloud deployment to further extend the system’s capabilities.

Keywords

Employee Management System, Angular, Spring Boot, MySQL, REST API, CRUD Operations, Web Application, HR Software, Database Management.

Introduction

In the modern corporate environment, managing employee information efficiently is essential for organizational success. Traditional methods of maintaining employee records, such as paper-based files or spreadsheets, are often time-consuming, error-prone, and inefficient. To address these challenges, organizations are increasingly adopting digital solutions that streamline employee data management and improve operational efficiency

The Employee Management System is designed as a web-based application to facilitate the management of employee records in an organization. It provides functionalities such as adding, updating, searching, viewing, and deleting employee details through an intuitive user interface. The system ensures secure and structured data handling, reducing manual efforts and minimizing errors

The primary objective of this project is to develop a fully functional and user-friendly Employee Management System that enhances HR operations. The system automates the process of managing employee information, provides a centralized database for easy access to employee records, and improves the efficiency, accuracy, and security of employee data. It enables quick and easy retrieval, modification, and deletion of records while offering a responsive and interactive UI for a seamless user experience

The project is built using Angular for the frontend, Java Spring Boot for backend development, and MySQL as the database. It is designed to cater to small and medium-sized enterprises as well as larger organizations that require an efficient employee record management system. The current version focuses on basic CRUD operations with potential future enhancements such as role-based authentication, payroll management, and attendance tracking

By implementing this system, organizations can reduce manual workload, increase efficiency, and improve HR management processes, ultimately leading to better workforce management and organizational growth

Problem and Problem Statement

Problem

Managing employee data manually or through traditional methods such as spreadsheets and paper-based records leads to inefficiencies, data redundancy, security risks, and difficulty in retrieving and updating records. HR departments often face challenges in maintaining accurate employee details, which can result in mismanagement of payroll, leaves, and employee records. Additionally, as an organization grows, handling a large workforce without a centralized system becomes increasingly complex and time-consuming.

Problem Statement

The absence of an efficient and centralized Employee Management System makes it challenging for organizations to manage employee records accurately and securely. This results in data inconsistencies, delays in HR operations, and increased administrative workload. To address this, there is a need for a digital solution that streamlines employee management, automates key HR processes, and provides a user-friendly interface for easy data handling.

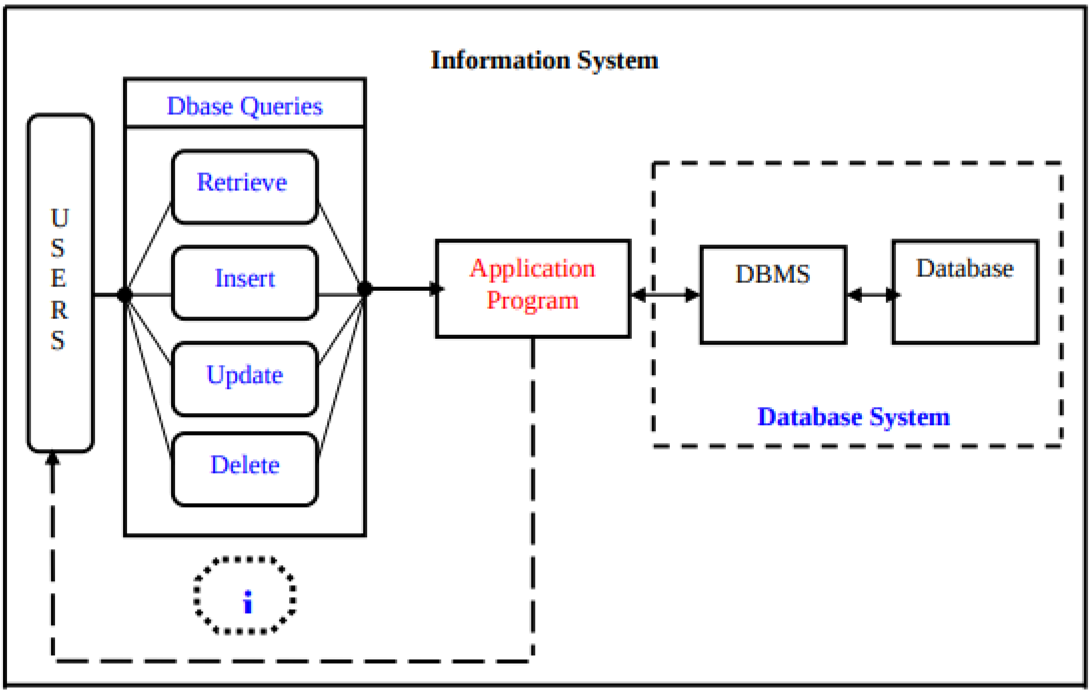
Solution

To overcome these challenges, this project proposes a web-based **Employee Management System (EMS)** that allows organizations to efficiently manage employee records with ease. The system is designed using **Angular** for the frontend, **Spring Boot** for the backend, and **MySQL** as the database.

The key features of the system include:

- **Centralized Database:** Securely stores employee information, ensuring easy access and retrieval.
- **CRUD Operations:** Enables adding, updating, searching, viewing, and deleting employee records.
- **User-Friendly Interface:** Provides an intuitive UI for seamless interaction.
- **Improved Data Accuracy:** Reduces human errors and ensures consistency in records.
- **Enhanced Security:** Protects sensitive employee data through proper authentication mechanisms.
- **Scalability:** Can be expanded to accommodate growing organizational needs.

By implementing this solution, organizations can significantly improve their HR operations, reduce administrative burdens, enhance decision-making processes, and create a more structured and efficient employee management system.



Programming Environments

The **Employee Management System (EMS)** is developed using a combination of modern technologies and frameworks to ensure a robust, scalable, and efficient application. The programming environments used in this project include:

1. Frontend Environment

- **Framework:** Angular
- **Language:** TypeScript, HTML, CSS
- **IDE/Code Editor:** Visual Studio Code
- **Libraries:** Bootstrap, Angular Material for UI enhancements
- **Package Manager:** Node Package Manager (NPM)

2. Backend Environment

- **Framework:** Spring Boot
- **Language:** Java
- **IDE:** IntelliJ IDEA / Eclipse / Spring Tool Suite (STS)
- **Libraries:** Spring MVC, Spring Data JPA, Hibernate
- **Build Tool:** Maven

3. Database Environment

- **Database Management System:** MySQL
- **Query Language:** SQL
- **Database Tool:** MySQL Workbench

4. Server and Deployment

- **Development Server:** Localhost (Spring Boot embedded Tomcat server)
- **API Handling:** RESTful APIs
- **Testing API Calls:** Postman, Swagger

5. Version Control & Collaboration

- **Version Control System:** Git
- **Repository Hosting:** GitHub
- **Collaboration Tool:** GitHub Issues, GitHub Projects

These programming environments collectively ensure a seamless development process, from writing and testing code to deploying and managing the application efficiently.

Database Analyzing, Design, and Implementation

1. Database Analysis

The **Employee Management System (EMS)** requires an efficient and structured database to manage employee records. The key considerations during database analysis include:

- Storing employee details such as ID, first name, last name, age, and salary.
- Enabling CRUD (Create, Read, Update, Delete) operations for employee management.
- Ensuring data integrity, security, and optimized query performance.

2. Database Design

The **MySQL** database follows a relational structure, with a primary focus on the **Employee** table.

2.1. Employee Table Schema

```
CREATE TABLE employees (
    id INT PRIMARY KEY AUTO_INCREMENT,
    fname VARCHAR(255) NOT NULL,
    lname VARCHAR(255) NOT NULL,
    age INT NOT NULL,
    salary FLOAT NOT NULL
);
```

2.2. Database Constraints

- **Primary Key:** `id` is the unique identifier for each employee.
- **Not Null Constraints:** Ensures mandatory fields (`fname`, `lname`, `age`, `salary`) are always filled.
- **Auto-Increment:** Automatically generates unique `id` values for new employees.

3. Database Implementation

3.1. Inserting Sample Employee Data

```
INSERT INTO employees (fname, lname, age, salary) VALUES
('Anil', 'Kumble', 31, 75000),
('Zaheer', 'Khan', 29, 68000),
('Rishabh', 'Pant', 24, 59000),
('Suresh', 'Raina', 27, 62000);
```

3.2. Retrieving Employee Data

```
SELECT * FROM employees;
```

3.3. Updating Employee Information

```
UPDATE employees SET salary = 80000 WHERE id = 1;
```

3.4. Deleting an Employee Record

```
DELETE FROM employees WHERE id = 3;
```

This database implementation ensures smooth management of employee records, with easy data retrieval and modification capabilities.

Database Management System (DBMS) Connections

The **Employee Management System (EMS)** uses **MySQL** as the database management system, connected via **Spring Boot JPA** and **Hibernate**. The connection is established using the following configuration in **application.properties**:

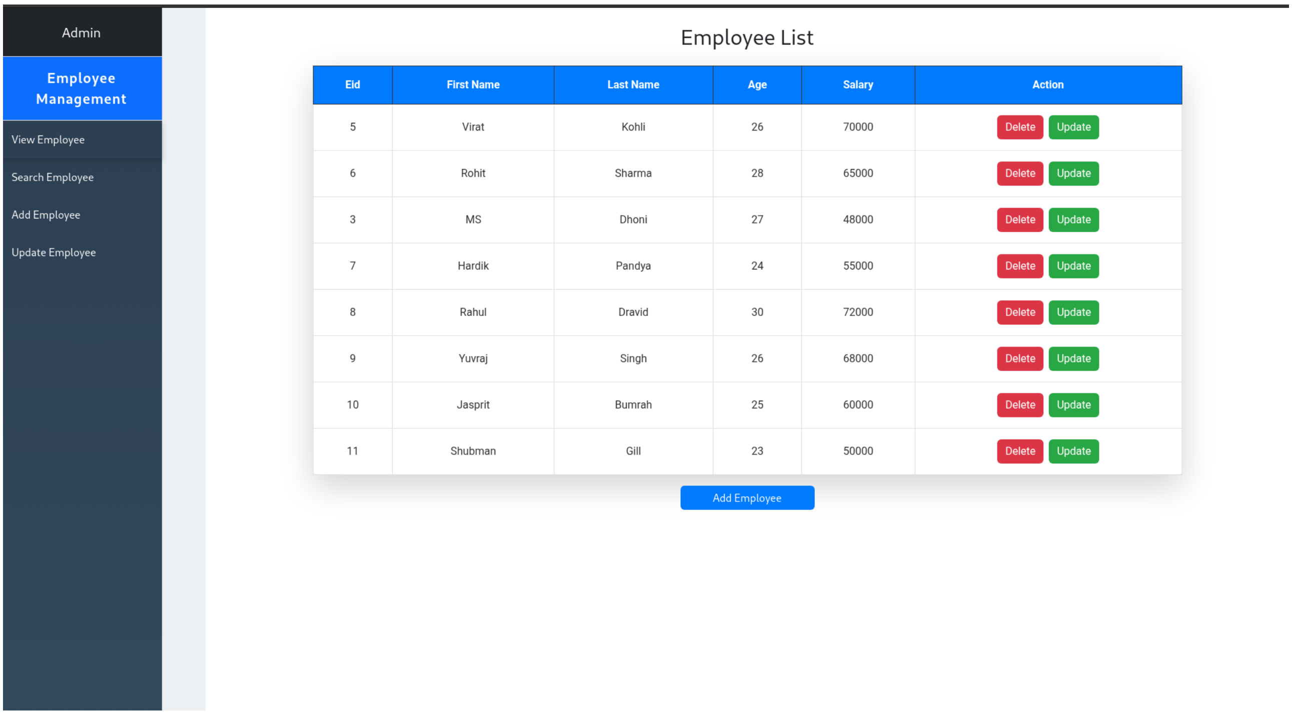
```
server.port=9090

# MySQL Configuration
spring.datasource.url=jdbc:mysql://localhost:3306/EMS
spring.datasource.username=root
spring.datasource.password=root
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

# Hibernate Configuration
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect
```

This configuration allows seamless communication between the **Spring Boot application** and the **MySQL database**, enabling CRUD operations for employee records.

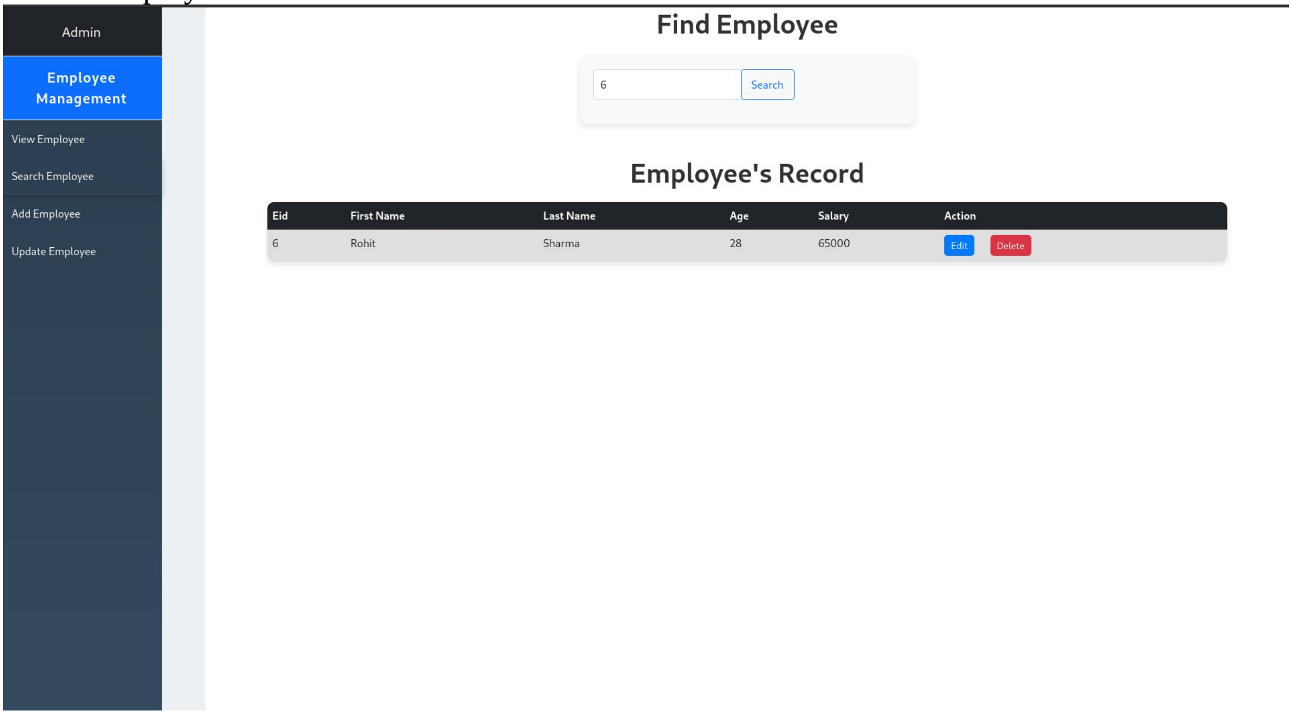
View Employee



Employee List

Eid	First Name	Last Name	Age	Salary	Action
5	Virat	Kohli	26	70000	Delete Update
6	Rohit	Sharma	28	65000	Delete Update
3	MS	Dhoni	27	48000	Delete Update
7	Hardik	Pandya	24	55000	Delete Update
8	Rahul	Dravid	30	72000	Delete Update
9	Yuvraj	Singh	26	68000	Delete Update
10	Jasprit	Bumrah	25	60000	Delete Update
11	Shubman	Gill	23	50000	Delete Update

Search Employee



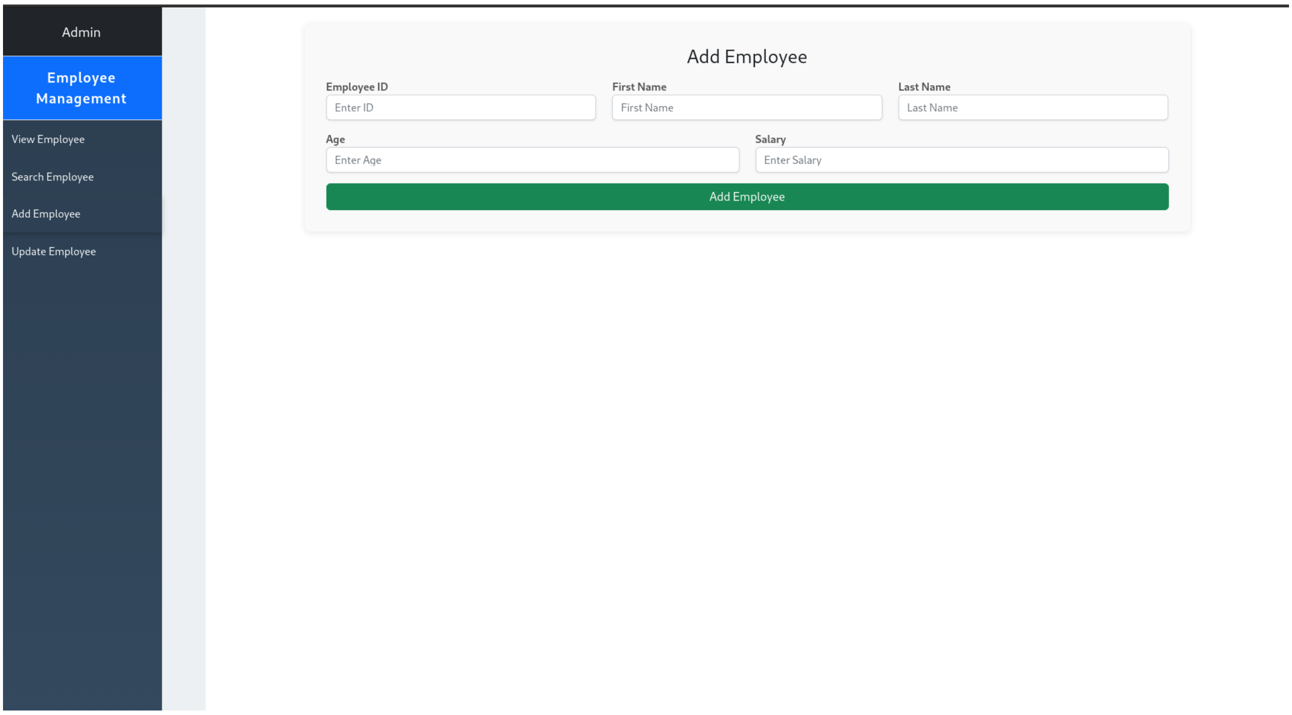
Find Employee

6

Employee's Record

Eid	First Name	Last Name	Age	Salary	Action
6	Rohit	Sharma	28	65000	Edit Delete

Add Employee



Add Employee

Employee ID: Enter ID

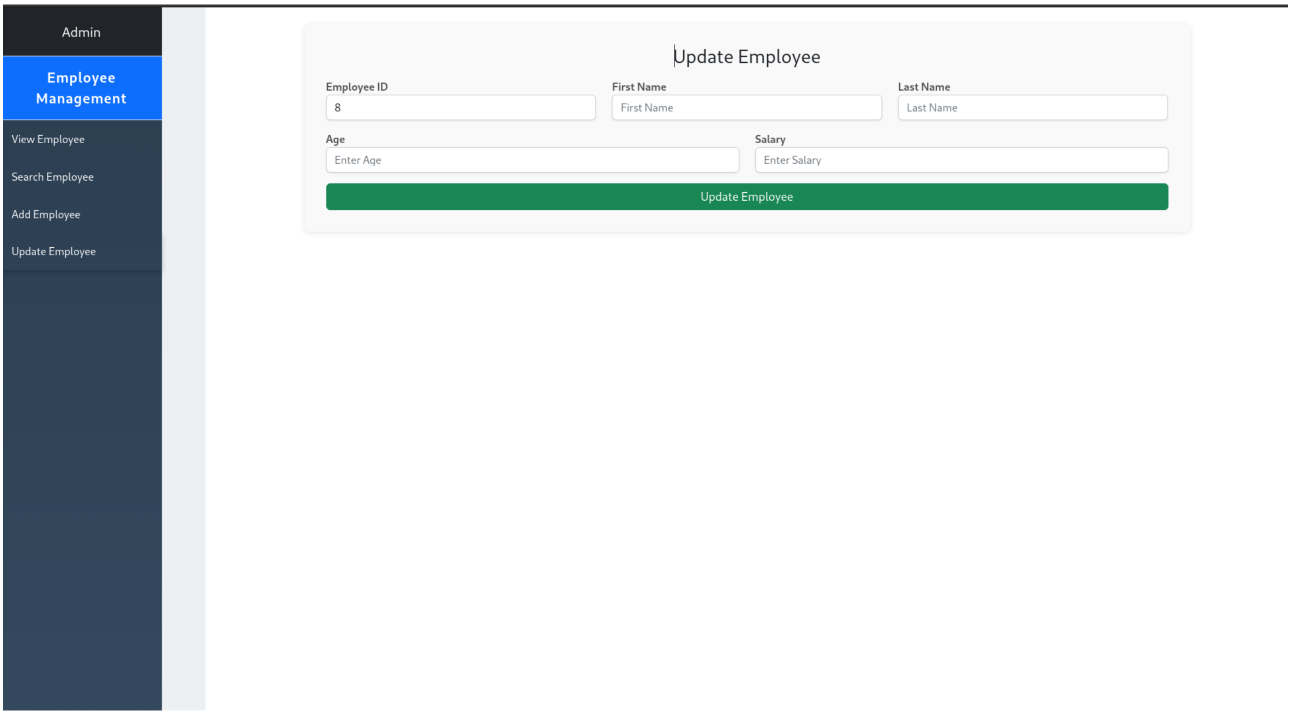
First Name: First Name

Last Name: Last Name

Age: Enter Age

Salary: Enter Salary

Update Employee



Update Employee

Employee ID: 8

First Name: First Name

Last Name: Last Name

Age: Enter Age

Salary: Enter Salary

Future Growth

The Employee Management System has the potential for significant enhancements and expansions to improve its efficiency, scalability, and security. Some key areas for future growth include:

1. **Role-Based Access Control (RBAC) –** Implementing authentication and authorization mechanisms to allow different access levels for employees, HR managers, and administrators.
2. **Cloud Integration –** Migrating the database and application to cloud platforms like AWS, Azure, or Google Cloud to improve scalability, reliability, and accessibility.
3. **Advanced Data Analytics –** Introducing **AI-driven analytics** to generate insights on employee performance, salary trends, and workforce planning for better decision-making.
4. **Payroll Management –** Integrating a **salary and payroll system** to automate salary processing, deductions, and tax calculations.
5. **Employee Self-Service Portal –** Enabling employees to update their personal details, apply for leaves, and access company policies through a secure portal.
6. **Mobile Application –** Developing an **Android and iOS app** to allow employees and HR managers to access the system from their mobile devices.
7. **Multi-Tenant Support –** Expanding the system to support multiple organizations or departments within a company, making it suitable for larger enterprises.
8. **Automated Notifications –** Implementing email or SMS notifications for key events such as **leave approvals, salary updates, or work anniversaries**.
9. **Integration with Third-Party Tools –** Connecting with **HR software, biometric attendance systems, and accounting tools** for seamless workflow management.

By implementing these improvements, the system can evolve into a **comprehensive HR management solution** that meets modern business requirements while maintaining flexibility and efficiency.

Conclusion

The Employee Management System is a robust and efficient application designed to streamline employee data management, enhancing productivity and organizational efficiency. By leveraging **Spring Boot for backend development, Angular for frontend, and MySQL for data storage**, the system ensures seamless interaction between the user interface and the database.

Throughout the project, various functionalities such as **adding, updating, searching, and deleting employee records** were successfully implemented, providing a user-friendly experience. The integration of RESTful APIs and database connectivity ensures real-time data processing, making the system highly responsive.

This project has demonstrated the **practical implementation of full-stack development** and the importance of structured database management in business applications. The system can be further improved by incorporating features like **role-based authentication, enhanced security mechanisms, and advanced reporting tools**.

Overall, the Employee Management System serves as a valuable tool for organizations looking to efficiently handle employee records while reducing manual effort and improving data accuracy.