

Employee Management System

Documentation

Employee Management System

Technology Stack: React | Spring Boot | MySQL | Bootstrap | JavaScript

Project Overview:

A full-stack Employee Management System that enables basic CRUD (Create, Read, Update, Delete) operations for employee data. The application uses **React** for the frontend to build a responsive user interface and **Spring Boot** for the backend to manage RESTful web services.

Key Features:

- **Create Employee:** Add new employee records with necessary details.
- **List Employees:** View all employees in a structured table with real-time data from the database.
- **Update Employee:** Modify existing employee information seamlessly.
- **Delete Employee:** Remove employee records efficiently.

Frontend (Client-side) Technologies:

- ReactJS with modern ES6 JavaScript
- Bootstrap 4.5 for responsive UI
- Axios for API integration
- Node.js & npm
- VS Code IDE

Backend (Server-side) Technologies:

- Spring Boot 2+ for RESTful services
- Spring Data JPA (Hibernate) for database interaction
- Maven 3.2+ for project management
- Embedded Tomcat 8.5+
- MySQL database
- IntelliJ IDEA IDE

Project Highlights:

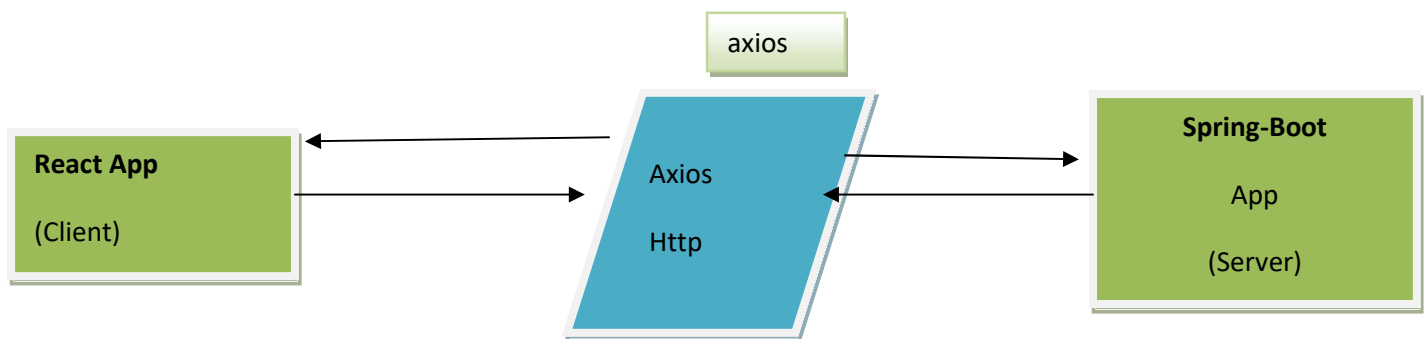
- Implemented **REST APIs** for CRUD operations using Spring Boot.
- Connected React frontend with backend REST APIs using Axios.
- Designed reusable React components (Header, Footer, Employee List, Add & Update Employee Forms).

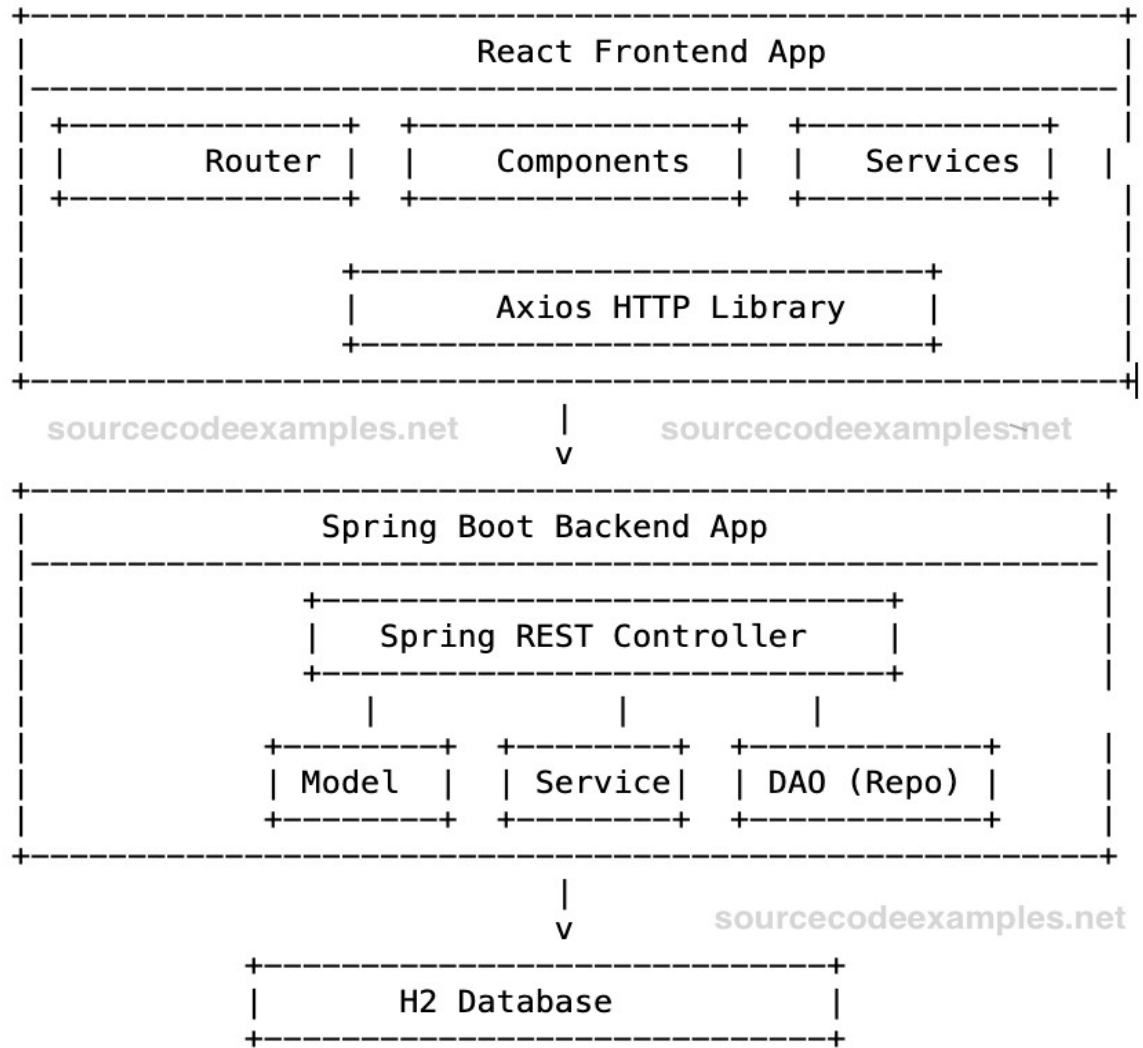
- Managed state and form handling in React for smooth user experience.
- Configured routing in React to navigate between different components.
- Used JPA Repository for database interactions and entity management.

Learning Outcomes:

- Gained hands-on experience with **full-stack development** using React and Spring Boot.
- Understood **client-server architecture** and RESTful API integration.
- Improved **frontend UI/UX design** with Bootstrap.
- Learned **CRUD operations implementation** with database connectivity.
- Strengthened knowledge of **React state management, form handling, and routing**.

Client Server Architecture





Change Styles

Editing

it

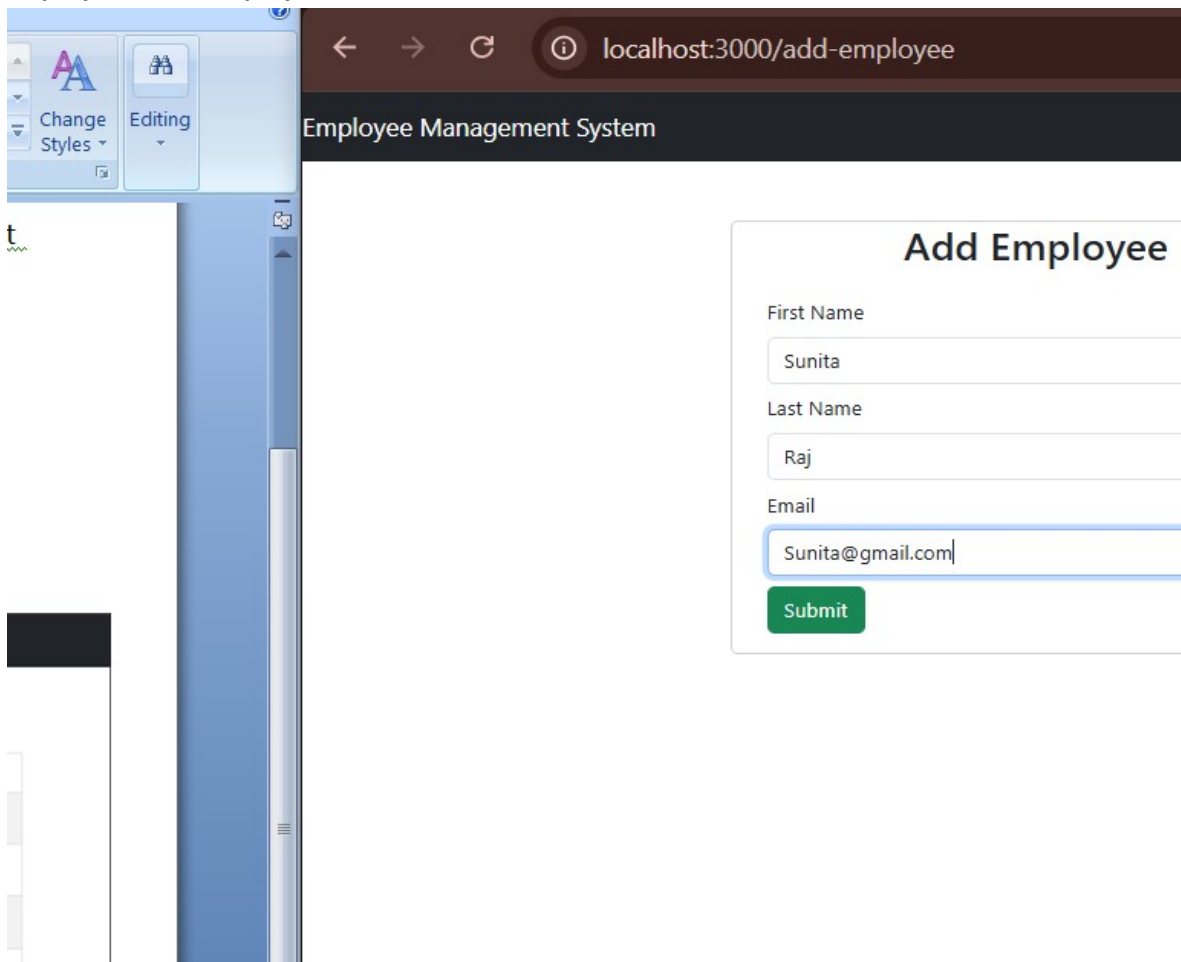
Employee Management System

List of Employees

Add Employee

Employee ID	Employee First Name	Employee Last Name	Employee Email
16	Ishani	Mahobiya	IshaniMahobiya@gmail.com
17	Geetanjali	Mahobiya	Geet@mahobiya.com
18	Yash	Raj	Yash@raj.com
19	Prachi	Pandey	Prachi@pandey.com
20	Nandini	Rajput	Nandini343@gmail.com

Employee—Add Employee

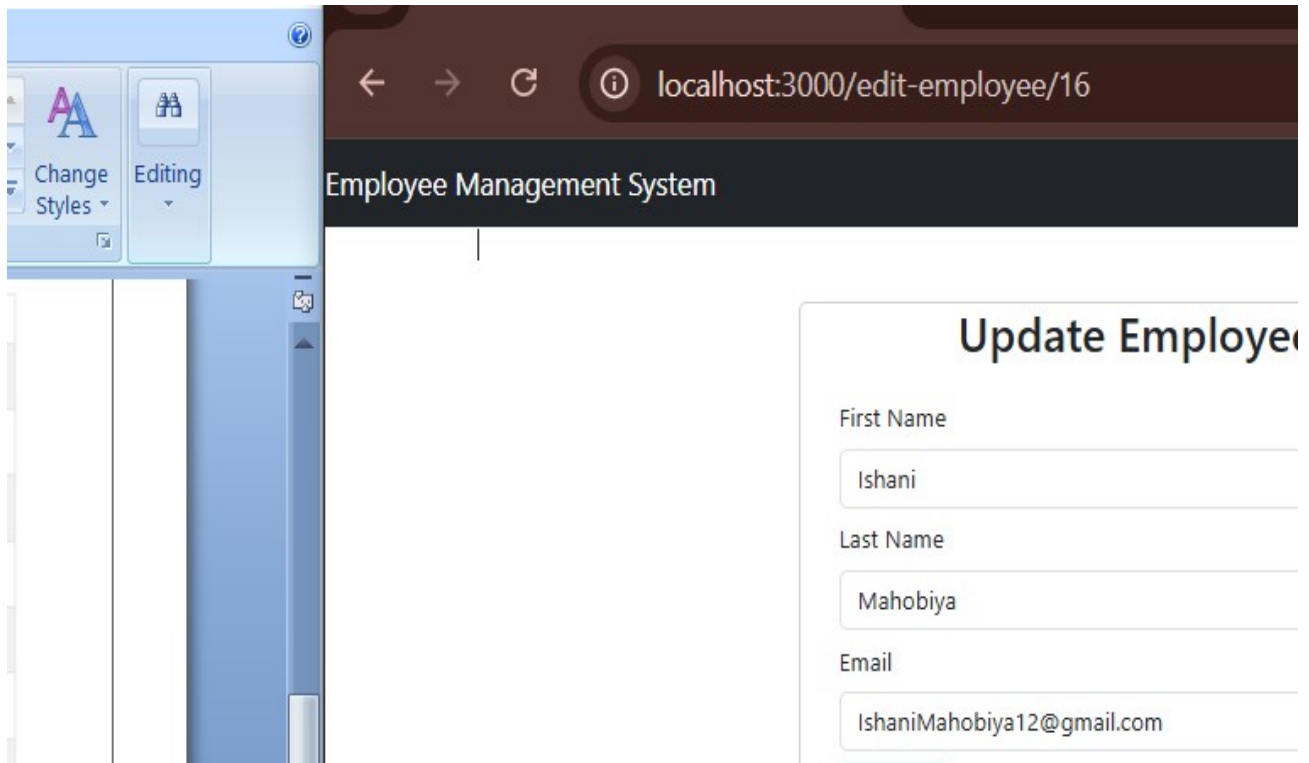


The screenshot shows a web browser window with the address bar displaying 'localhost:3000/add-employee'. The page title is 'Employee Management System'. On the left, there is a sidebar with a 'Change Styles' dropdown and an 'Editing' button. The main content area features a form titled 'Add Employee' with the following fields:

- First Name:** Sunita
- Last Name:** Raj
- Email:** Sunita@gmail.com

A green 'Submit' button is located at the bottom of the form.

Update Employee



The screenshot shows a web browser window with the address bar displaying `localhost:3000/edit-employee/16`. The page title is "Employee Management System". On the left, there is a sidebar with a "Change Styles" button and an "Editing" dropdown menu. The main content area features a form titled "Update Employee" with three input fields: "First Name" (containing "Ishani"), "Last Name" (containing "Mahobiya"), and "Email" (containing "IshaniMahobiya12@gmail.com"). A green underline is visible under the email field.

localhost:3000/edit-employee/16

Employee Management System

Update Employee

First Name

Ishani

Last Name

Mahobiya

Email

IshaniMahobiya12@gmail.com