

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:

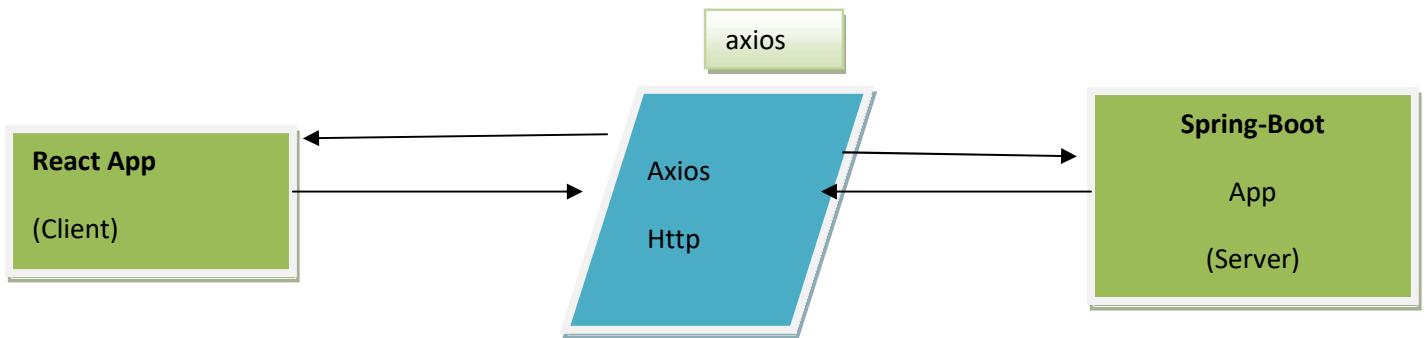
- Implemented **REST APIs** for CRUD operations using Spring Boot.
- Connected React frontend with backend REST APIs using Axios.
- Designed reusable React components (Header, Footer, Employee List, Add & Update Employee Forms).

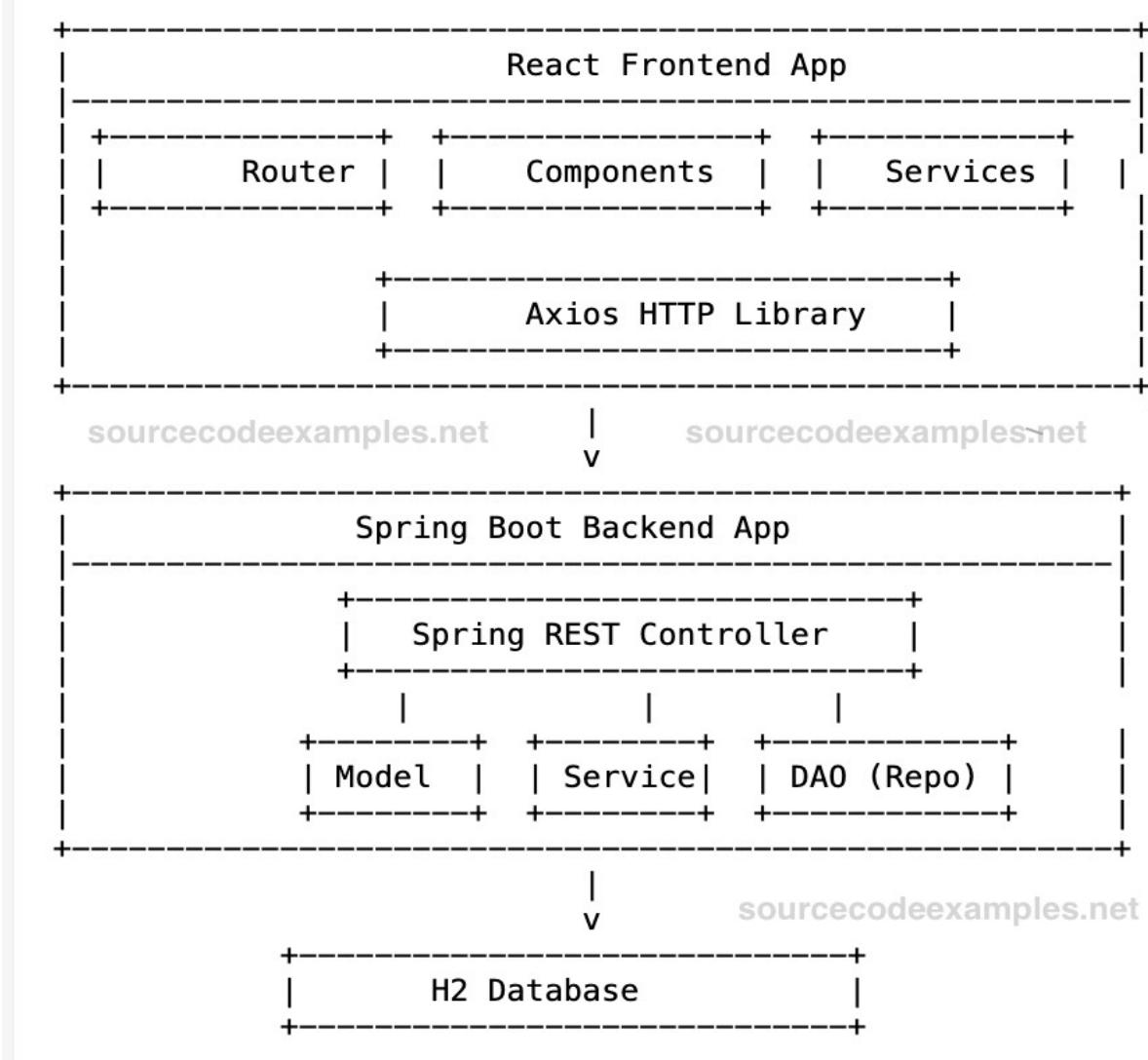
- Managed state and form handling in React for smooth user experience.
- Configured routing in React to navigate between different components.
- Used JPA Repository for database interactions and entity management.

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





The screenshot shows a Microsoft Word document titled "Employee Management System". The title bar includes "Employee Management System" and "List of Employees". The ribbon menu has "Change Styles" and "Editing" tabs. A blue button labeled "Add Employee" is visible above a table. The table has four columns: "Employee ID", "Employee First Name", "Employee Last Name", and "Employee Email". The data rows are:

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

Employee—Add Employee

The screenshot shows a web browser window with the URL `localhost:3000/add-employee` in the address bar. The title bar of the browser says "Employee Management System". On the left side of the browser, there is a vertical toolbar with icons for "Change Styles" and "Editing". The main content area is titled "Add Employee". It contains three input fields: "First Name" with the value "Sunita", "Last Name" with the value "Raj", and "Email" with the value "Sunita@gmail.com". A green "Submit" button is located below the email field. The browser's interface includes standard navigation buttons (back, forward, search) and a status bar at the bottom.

Employee Management System

Add Employee

First Name
Sunita

Last Name
Raj

Email
Sunita@gmail.com

Submit

Update Employee

localhost:3000/edit-employee/16

Employee Management System

Update Employee

First Name
Ishani

Last Name
Mahobiya

Email
IshaniMahobiya12@gmail.com