**Employee Management System**

**Name:** Subhendu Adak **Data:**27/03/2025

**Email:** Subhendu.Adak@version1.com **Emp Id:** 7411

**Technologies Used:**

**Front-End:** React Js, CSS, Bootstrap

**Back-End:** JSON Server with JSON File as Database

**Source Code:**

**1.App Component:**

* **App.js**
* // import logo from './logo.svg';
* import './App.css';
* import EmployeeManagementSystem from './component/EmployeeManagementSystem';
* function App() {
* return (
* <div className="App">
* {/\* <h1>Employee Management System</h1> \*/}
* <EmployeeManagementSystem></EmployeeManagementSystem>
* </div>
* );
* }
* export default App;

**2.Employee Management System Component and CSS:**

* **employeeManagementSystem.js:**
* import React, { Component } from "react";
* import "./EmployeeManagementSystem.css";
* import "bootstrap/dist/css/bootstrap.min.css";
* import { EMSService } from "../service/EMSService";
* import {Employee} from "../model/Employee";
* export default class EmployeeManagementSystem extends Component {
* constructor(props) {
* super(props);
* this.emsService = new EMSService();
* this.employee = new Employee();
* this.state = {
* employeeId: "",
* employeeIdError: '',
* employeeName: "",
* employeeNameError: '',
* employeeSalary: "",
* employeeSalaryError: '',
* result : ''
* }
* }
* handleEmpIdChange = (event) =>{
* this.setState ({employeeId: event.target.value});
* let error = this.validateEmployeeId(this.state.employeeId);
* this.setState({employeeIdError: error});
* }
* validateEmployeeId = (employeeId) => {
* let pattern = /^[0-9]+$/;
* if(employeeId === ''){
* return 'Employee ID is required';
* }
* else if(! pattern.test(employeeId)){
* return 'Employee ID should be a number';
* }
* else{
* return null;
* }
* }
* handleEmpNameChange = (event) =>{
* this.setState ({employeeName: event.target.value});
* let error = this.validateEmployeeName(this.state.employeeName);
* this.setState({employeeNameError: error});
* }
* validateEmployeeName = (employeeName) => {
* let pattern = /^[a-zA-Z]+$/;
* if(employeeName === ''){
* return 'Employee Name is required';
* }
* else if(! pattern.test(employeeName)){
* return 'Employee Name should be a string';
* }
* else{
* return null;
* }
* }
* handleEmpSalaryChange = (event) =>{
* this.setState ({employeeSalary: event.target.value});
* let error = this.validateEmployeeSalary(this.state.employeeSalary);
* this.setState({employeeSalaryError: error});
* }
* validateEmployeeSalary = (employeeSalary) => {
* let pattern = /^[0-9]+$/;
* if(employeeSalary === ''){
* return 'Employee Salary is required';
* }
* else if(! pattern.test(employeeSalary)){
* return 'Employee Salary should be a number';
* }
* else{
* return null;
* }
* }
* insertEmployee = (event) => {
* this.employee.id = this.state.employeeId;
* this.employee.name = this.state.employeeName;
* this.employee.salary = this.state.employeeSalary;
* let res = this.emsService.insertEmployee(this.employee);
* this.setState({result: res});
* }
* updateEmployee = (event) => {
* this.employee.id = this.state.employeeId;
* this.employee.name = this.state.employeeName;
* this.employee.salary = this.state.employeeSalary;
* let res = this.emsService.updateEmployee(this.employee);
* this.setState({result: res});
* }
* deleteEmployee = (event) => {
* let id = this.state.employeeId;
* let res = this.emsService.deleteEmployee(id);
* this.setState({result: res});
* }
* // findEmployee = (event) => {
* //   let id = this.state.employeeId;
* //   let res = this.emsService.findEmployee(id);
* //   this.setState({result: res});
* // }
* findEmployee = (event) => {
* event.preventDefault();
* let id = this.state.employeeId;
* this.emsService.findEmployee(id).then((res) => {
* this.setState({ result: res });
* }).catch((err) => {
* this.setState({ result: 'Employee is not in the list' });
* });
* }
* findAllEmployee = (event) => {
* let res = this.emsService.findAllEmployee();
* this.setState({result: res});
* }
* render() {
* return (
* <div
* className="d-flex flex-column justify-content-center align-items-center vh-100"
* style={{ backgroundColor: "#e9ecef" }}
* >
* <h1 className="mb-4" style={{ color: "#007bff" }}>
* Employee Management System
* </h1>
* <div
* className="container-fluid p-4"
* style={{
* maxWidth: "500px",
* backgroundColor: "#ffffff",
* border: "1px solid #dee2e6",
* borderRadius: "8px",
* boxShadow: "0 4px 6px rgba(0, 0, 0, 0.1)",
* }}
* >
* <form>
* <div className="form-group mb-3">
* <label
* htmlFor="empid"
* className="form-label"
* style={{ color: "#495057" }}
* >
* Employee ID
* </label>
* <input
* type="text"
* className="form-control"
* id="empid"
* placeholder="Enter Employee ID"
* style={{ borderColor: "#007bff" }}
* onChange={this.handleEmpIdChange}
* />
* {this.state.employeeIdError != null ? <span><b><font color = 'red'>{this.state.employeeIdError}</font></b></span>: ''}
* </div>
* <div className="form-group mb-3">
* <label
* htmlFor="empname"
* className="form-label"
* style={{ color: "#495057" }}
* >
* Employee Name
* </label>
* <input
* type="text"
* className="form-control"
* id="empname"
* placeholder="Enter Employee Name"
* style={{ borderColor: "#007bff" }}
* onChange={this.handleEmpNameChange}
* />
* {this.state.employeeNameError != null ? <span><b><font color = 'red'>{this.state.employeeNameError}</font></b></span>: ''}
* </div>
* <div className="form-group mb-3">
* <label
* htmlFor="empsalary"
* className="form-label"
* style={{ color: "#495057" }}
* >
* Employee Salary
* </label>
* <input
* type="text"
* className="form-control"
* id="empsalary"
* placeholder="Enter Employee Salary"
* style={{ borderColor: "#007bff" }}
* onChange={this.handleEmpSalaryChange}
* />
* {this.state.employeeSalaryError != null ? <span><b><font color = 'red'>{this.state.employeeSalaryError}</font></b></span>: ''}
* </div>
* </form>
* </div>
* <br></br>
* <div className="d-flex flex-wrap justify-content-center">
* <button className="btn btn-primary me-2 mb-2" onClick={this.insertEmployee}>Add Employee</button>
* <button className="btn btn-danger me-2 mb-2" onClick={this.deleteEmployee}>Delete Employee</button>
* <button className="btn btn-warning me-2 mb-2" onClick={this.updateEmployee}>Update Employee</button>
* <button className="btn btn-info me-2 mb-2" onClick={this.findEmployee}>Find Employee</button>
* <button className="btn btn-success mb-2" onClick={this.findAllEmployee}>Get All Employees</button>
* </div>
* <hr></hr>
* <div className="alert alert-danger" role="alert">
* {this.state.result}
* </div>
* </div>
* );
* }
* }
* **employeeManagementSystem.css:**
* .MyHeading{
* color: #18d52b;
* font-size: 60px;
* font-weight: bold;
* text-align: center;
* border: 2px solid black;
* border-radius: 20px;
* width: 90%;
* margin: 0 auto;
* margin-top: 15px;
* }

**3.Model Class:**

* **Employee.js**
* export class Employee {
* id = 0;
* name = '';
* salary = 0
* }

**4.Service Class:**

* **EMSService.js**
* import axios from 'axios';
* export class EMSService {
* constructor() {
* this.url = 'http://localhost:3004/employees';
* }
* insertEmployee = (employee) => {
* let result = 'Employee inserted successfully';
* axios.post(this.url, employee).then(() => {
* }).catch(() => {
* result = 'Employee insertion failed';
* });
* return result;
* }
* updateEmployee = (employee) => {
* let result = 'Employee update successfully';
* axios.put(this.url + "/" + employee.id, employee).then(() => {
* }).catch(() => {
* result = 'Employee update failed';
* });
* return result;
* }
* deleteEmployee = (id) => {
* let result = 'Employee delete successfully';
* axios.delete(this.url + "/" + id).then(() => {
* }).catch(() => {
* result = 'Employee delete failed';
* });
* return result;
* }
* // findEmployee = (id) => {
* //     let result = 'Employee not find';
* //     axios.get(this.url + "/" + id).then((resonse) => {
* //         alert(resonse.data.id + " " + resonse.data.name + " " + resonse.data.salary);
* //         result = "Employee id: " + resonse.data.id + ", name: " + resonse.data.name + ", salary: " + resonse.data.salary;
* //     }).catch(() => {
* //         result = 'Employee is not in the list';
* //     });
* //     return result;
* // }
* // using async and await
* findEmployee = async(id) => {
* try {
* const res = await axios.get(this.url + "/" + id);
* return `Employee ID: ${res.data.id}, Name: ${res.data.name}, Salary: ${res.data.salary}`;
* }catch(error){
* return 'Employee is not in the list';
* }
* }
* // using promise
* // findEmployee = (id) => {
* //     return new Promise((resolve, reject) => {
* //       axios.get(this.url + "/" + id)
* //         .then((response) => {
* //           const result = `Employee ID: ${response.data.id}, Name: ${response.data.name}, Salary: ${response.data.salary}`;
* //           resolve(result);
* //         })
* //         .catch((error) => {
* //           reject('Employee is not in the list');
* //         });
* //     });
* //   }
* findAllEmployee = async(id) => {
* try {
* const res = await axios.get(this.url);
* let employees = res.data;
* let result = '';
* employees.forEach(employees => {
* result += `Employee ID: ${employees.id}, Name: ${employees.name}, Salary: ${employees.salary} <br>`;
* });
* return result;
* }catch(error){
* return 'Employee is not in the list';
* }
* }
* }

**5.db.json:**

* {
* "employees": [
* {
* "id": 101,
* "name": "Subhendu",
* "salary": 3000
* },
* {
* "id": "102",
* "name": "Irfan",
* "salary": "8000"
* },
* {
* "id": "106",
* "name": "Raju",
* "salary": "4000"
* },
* {
* "id": "107",
* "name": "Ravi",
* "salary": ""
* },
* {
* "id": "104",
* "name": "Subhendu",
* "salary": "5000"
* }
* ]
* }

**6.Package.json**

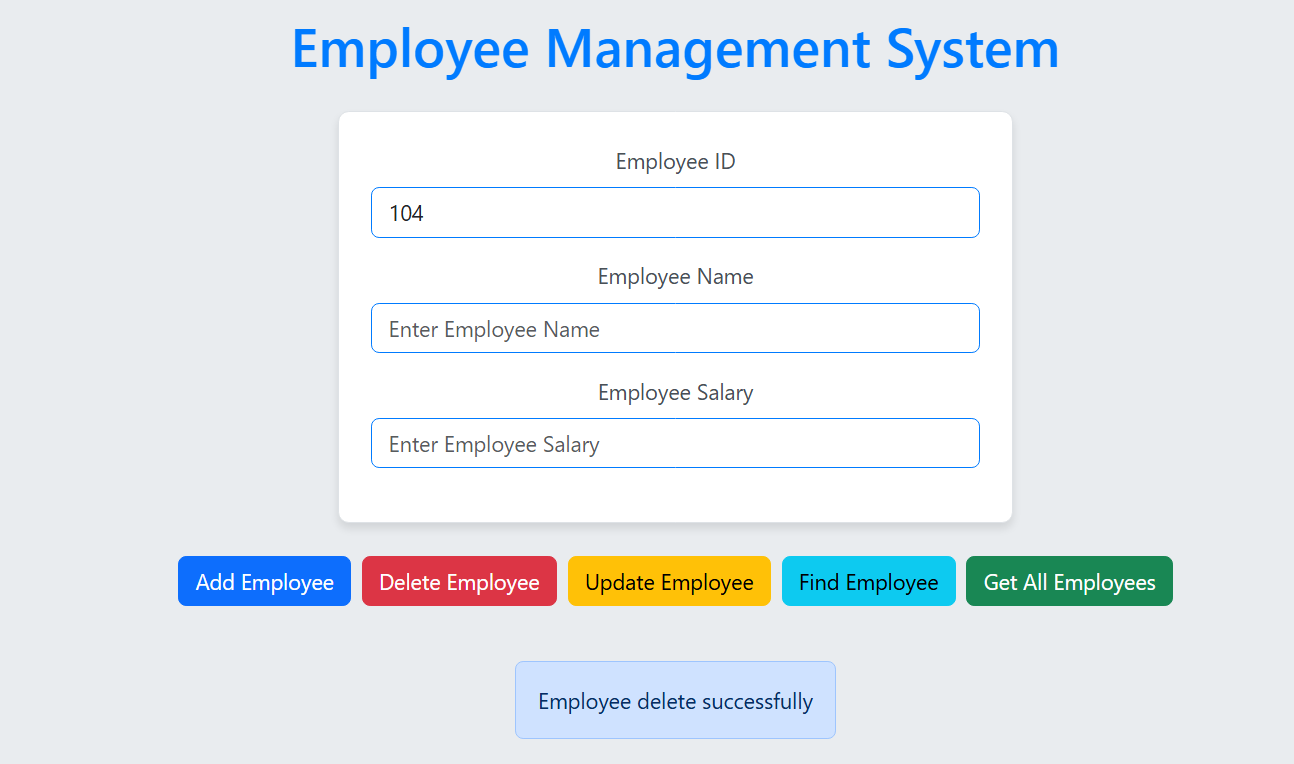
* {
* "name": "ems-app",
* "version": "0.1.0",
* "private": true,
* "dependencies": {
* "@testing-library/dom": "^10.4.0",
* "@testing-library/jest-dom": "^6.6.3",
* "@testing-library/react": "^16.2.0",
* "@testing-library/user-event": "^13.5.0",
* "axios": "^1.8.4",
* "bootstrap": "^5.3.3",
* "react": "^19.0.0",
* "react-dom": "^19.0.0",
* "react-scripts": "5.0.1",
* "web-vitals": "^2.1.4"
* },
* "scripts": {
* "start": "react-scripts start",
* "build": "react-scripts build",
* "test": "react-scripts test",
* "eject": "react-scripts eject",
* "json-server": "json-server --watch db.json --port 3004"
* },
* "eslintConfig": {
* "extends": [
* "react-app",
* "react-app/jest"
* ]
* },
* "browserslist": {
* "production": [
* ">0.2%",
* "not dead",
* "not op\_mini all"
* ],
* "development": [
* "last 1 chrome version",
* "last 1 firefox version",
* "last 1 safari version"
* ]
* }
* }

**Output Screen Shots:**

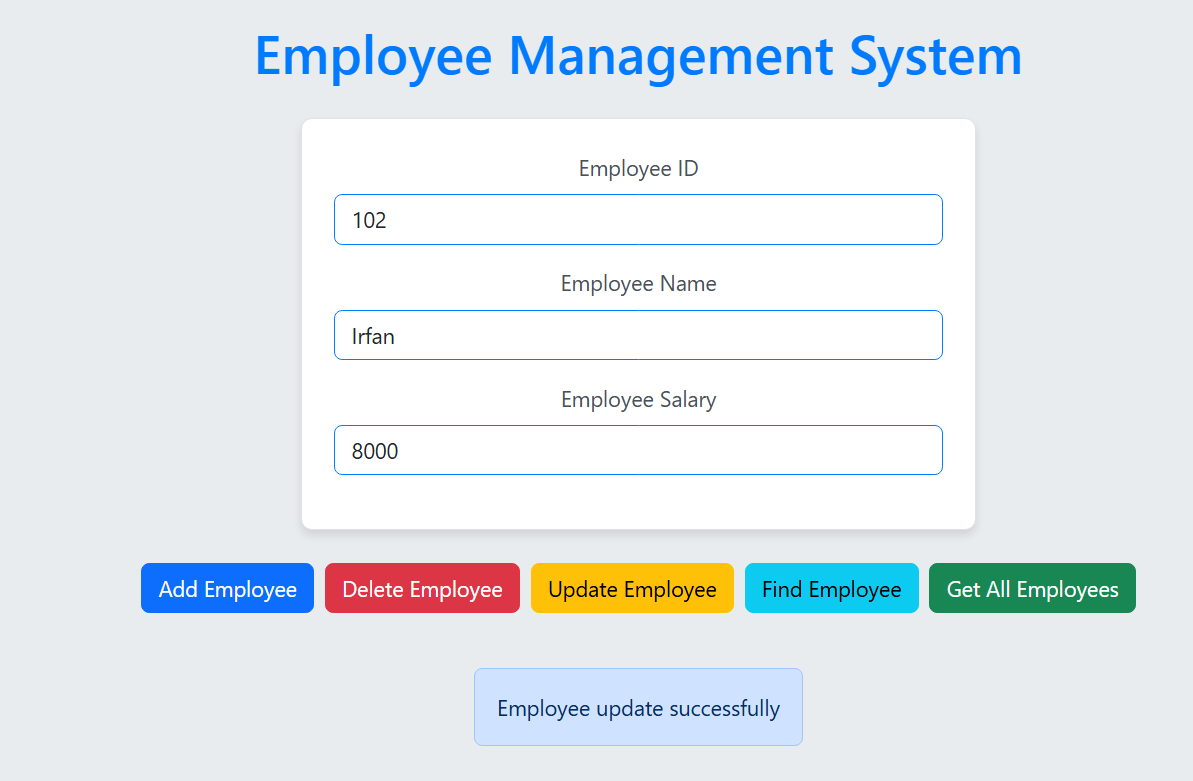
* + 1. **Insert:**

****

* + 1. **Delete:**

****

* + 1. **Update:**

****

* + 1. **Find:**

**A screenshot of a computer

AI-generated content may be incorrect.**

* + 1. **Find All:**

****