Employee Management System

Detailed React.js Project Report

1. Abstract

The Employee Management System is a frontend web application developed using React.js. The purpose of this project is to demonstrate how CRUD operations can be implemented using modern React features such as hooks, routing, and browser-based storage. This project is suitable for students learning React and frontend development.

2. Problem Statement

Managing employee information manually is inefficient and error-prone. There is a need for a simple system that allows adding, viewing, editing, and deleting employee records digitally with persistent storage.

3. Objectives of the Project

- To build a React-based CRUD application
- To understand React Hooks like useState and useEffect
- To implement client-side routing using React Router
- To store and retrieve data using localStorage
- To design responsive UI using Tailwind CSS

4. System Architecture

The application follows a component-based architecture. Each major functionality is divided into reusable React components. Data is stored in the browser's localStorage and shared across components.

- Home Component – Displays employee cards with search
- AddEmployee Component – Adds new employee records
- ViewEmployee Component – Displays employees in tabular format
- UpdateEmployee Component – Updates employee details
- localStorage – Persistent client-side storage

5. Technology Stack

- React.js – Frontend framework
- Tailwind CSS – Styling and layout
- Lucide React – Icon library
- React Router DOM – Page navigation
- localStorage – Data persistence

6. Functional Modules

6.1 Home Page

The Home page displays all employees in a card layout. It includes a search bar that filters employees by name or employee ID.

6.2 Add Employee

This module provides a form to enter employee details such as ID, name, email, phone number, designation, department, salary, status, and profile image.

6.3 View Employee

The View Employee module displays employee data in a table format. Users can update or delete employee records from this view.

6.4 Update Employee

This module allows users to edit existing employee information. The form is auto-filled with selected employee data using route parameters.

6.5 Delete Employee

The delete functionality removes an employee record from the system. The data is permanently deleted from localStorage.

7. Data Storage Structure

Employee records are stored as an array of objects in the browser's localStorage. Each object represents a single employee record.

Sample Employee Data Structure: { eid, name, email, phone, designation, department, salary, status, img }

8. Learning Outcomes

- Hands-on experience with React Hooks
- Understanding CRUD operations in React
- Using client-side routing effectively
- Working with browser storage
- Building responsive UI layouts

9. Conclusion

The Employee Management System project successfully demonstrates the implementation of a real-world CRUD application using React. This project helps students strengthen their frontend development skills.