**WEEK-7-React-HOL 14**

**SUPERSET ID 6393674**

**Objectives**

**• Explain the need and Benefits of React Context API**

React Context API helps in avoiding prop drilling by allowing data to be shared globally across components. It improves code maintainability and simplifies state management in deeply nested components.

**• Working with createContext()**

The createContext() function creates a Context object used to pass data through the component tree without manually passing props. It includes a Provider to supply data and a Consumer or useContext() to access it.

**• List the types of Router Components**

React Router provides components like BrowserRouter, HashRouter, MemoryRouter, Routes, and Route to manage navigation and render components based on the URL.

14. Developers of Apps Centric Solutions have created an employee management application which supports light and dark themes for the buttons. The current solution uses the react state and props to provide the theme name to be used from App component to Employee List component and from there to Employee Card component. Quality assurance team analyzed the solutions and found the technique being used to be a substandard one. React architect suggested to use the react context API to share the theme name with nested child components instead of passing them down using props from the parent component.

**App.js**

import logo from './logo.svg';

import './App.css';

import { EmployeesData } from './Employee';

import EmployeesList from './EmployeesList';

import { useState } from 'react';

import ThemeContext from './ThemeContext'; // ✅ Step 1: Import ThemeContext

function App() {

  const Employees = EmployeesData;

  const [theme, setTheme] = useState('light');

  return (

    // ✅ Step 2: Wrap everything inside ThemeContext.Provider

    <ThemeContext.Provider value={theme}>

      <div>

        <label>SELECT A THEME </label>

        <select onChange={(e) => setTheme(e.target.value)}>

          <option value='light'>Light</option>

          <option value='dark'>Dark</option>

        </select>

      </div>

      {/\* ✅ Step 3: Do NOT pass theme as a prop \*/}

      <EmployeesList employees={Employees} />

    </ThemeContext.Provider>

  );

}

export default App;

**OUTPUT**

