**Objectives**

**Explain about conditional rendering in React**

Conditional rendering means showing different components or elements based on certain conditions (like user login status, permissions, etc.).

Example:

{isLoggedIn ? <User /> : <Guest />}

React renders <User /> if isLoggedIn is true, otherwise it shows <Guest />.

**Define element variables**

Element variables are used to store JSX elements in a variable, making conditional rendering more readable and flexible.

Example:

let page;

if (isLoggedIn) {

page = <User />;

} else {

page = <Guest />;

}

return <div>{page}</div>;

This is useful when rendering logic is more complex.

**Explain how to prevent components from rendering**

To prevent a component from rendering, simply return null or use conditions so the component is not included in the JSX.

Example:

if (!isLoggedIn) return null;

return <UserDashboard />;

This ensures the component is completely skipped and not rendered at all.

**Code :**

**--index.js**

import React from 'react';

import ReactDOM from 'react-dom/client';

import App from './App'; // Import the main App component

// Get the root element from the HTML

const root = ReactDOM.createRoot(document.getElementById('root'));

// Render the App component into the root

root.render(

  <React.StrictMode>

    {/\* Tailwind CSS CDN for global styles \*/}

    <script src="https://cdn.tailwindcss.com"></script>

    {/\* Google Fonts for Inter \*/}

    <link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;600;700&display=swap" rel="stylesheet" />

    <style>

      {`

        body {

          font-family: 'Inter', sans-serif;

          margin-top:10px;

          padding: 200px;

          box-sizing: border-box;

        }

      `}

    </style>

    <App />

  </React.StrictMode>

);

**--App.js**

import React, { useState, useEffect } from 'react';

import GuestGreeting from './Guest'; // Import GuestGreeting component

import UserGreeting from './User';   // Import UserGreeting component

// App Component: Main application logic, handles login state and page rendering

export default function App() {

  const [isLoggedIn, setIsLoggedIn] = useState(false);

  // Simulate persistent login state (e.g., from local storage or session)

  useEffect(() => {

    const storedLoginStatus = localStorage.getItem('isLoggedIn');

    if (storedLoginStatus === 'true') {

      setIsLoggedIn(true);

    }

  }, []);

  const handleLogin = () => {

    setIsLoggedIn(true);

    localStorage.setItem('isLoggedIn', 'true'); // Persist login status

  };

  const handleLogout = () => {

    setIsLoggedIn(false);

    localStorage.removeItem('isLoggedIn'); // Clear login status

  };

  return (

    <div className="min-h-screen bg-gradient-to-br from-blue-100 to-purple-100 font-sans text-gray-900 flex items-center justify-center">

      {isLoggedIn ? (

        <UserGreeting onLogoutClick={handleLogout} />

      ) : (

        <GuestGreeting onLoginClick={handleLogin} />

      )}

    </div>

  );

}

**-- Guest.js**

import React from 'react';

// LoginButton Component

const LoginButton = ({ onClick }) => (

  <button

    onClick={onClick}

    className="bg-blue-600 hover:bg-blue-700 text-white font-bold py-2 px-6 rounded-lg shadow-md transition duration-300 ease-in-out transform hover:scale-105 focus:outline-none focus:ring-2 focus:ring-blue-500 focus:ring-offset-2"

  >

    Login

  </button>

);

// GuestGreeting Component: Displays "Please sign up."

const GuestGreeting = ({ onLoginClick }) => (

  <div className="flex flex-col items-center justify-center p-8 bg-white rounded-lg shadow-xl">

    <h2 className="text-4xl font-bold text-gray-800 mb-6">Please sign up.</h2>

    <LoginButton onClick={onLoginClick} />

  </div>

);

export default GuestGreeting;

**--User.js**

import React from 'react';

// LogoutButton Component

const LogoutButton = ({ onClick }) => (

  <button

    onClick={onClick}

    className="bg-red-500 hover:bg-red-600 text-white font-bold py-2 px-6 rounded-lg shadow-md transition duration-300 ease-in-out transform hover:scale-105 focus:outline-none focus:ring-2 focus:ring-red-500 focus:ring-offset-2"

  >

    Logout

  </button>

);

// UserGreeting Component: Displays "Welcome back"

const UserGreeting = ({ onLogoutClick }) => (

  <div className="flex flex-col items-center justify-center p-8 bg-white rounded-lg shadow-xl">

    <h2 className="text-4xl font-bold text-gray-800 mb-6">Welcome back</h2>

    <LogoutButton onClick={onLogoutClick} />

  </div>

);

export default UserGreeting;

**Output :**



