## **Objectives**

* Explain about conditional rendering in React
* Define element variables
* Explain how to prevent components from rendering

In this hands-on lab, you will learn how to:

* Implement conditional rendering in React applications

## **Prerequisites**

The following is required to complete this hands-on lab:

* Node.js
* NPM
* Visual Studio Code

## **Notes**

Estimated time to complete this lab: **60 minutes.**

Create a React Application named “ticketbookingapp” where the guest user can browse the page where the flight details are displayed whereas the logged in user only can book tickets.

The Login and Logout buttons should accordingly display different pages. Once the user is logged in the User page should be displayed. When the user clicks on Logout, the Guest page should be displayed.





**Hint:**







Code:

import React, { useState } from 'react';

// --- Mock Data ---

// A simple array of flight data for the guest view.

const flightData = [

{ id: 'AI-202', from: 'Delhi (DEL)', to: 'Mumbai (BOM)', time: '08:45 AM' },

{ id: '6E-531', from: 'Bangalore (BLR)', to: 'Chennai (MAA)', time: '11:00 AM' },

{ id: 'UK-820', from: 'Kolkata (CCU)', to: 'Delhi (DEL)', time: '02:15 PM' },

];

// --- Child Components for Different States ---

// Component to show a welcome message to a logged-in user.

function UserGreeting() {

return <h1 className="text-3xl font-bold text-gray-800">Welcome back!</h1>;

}

// Component to show a message to a guest user.

function GuestGreeting() {

return <h1 className="text-3xl font-bold text-gray-800">Please sign up.</h1>;

}

// A simple Login button component. It receives the onClick handler from its parent.

function LoginButton({ onClick }) {

return (

<button

onClick={onClick}

className="bg-blue-600 text-white font-bold py-2 px-6 rounded-lg hover:bg-blue-700 transition duration-300 ease-in-out shadow-md"

>

Login

</button>

);

}

// A simple Logout button component.

function LogoutButton({ onClick }) {

return (

<button

onClick={onClick}

className="bg-red-600 text-white font-bold py-2 px-6 rounded-lg hover:bg-red-700 transition duration-300 ease-in-out shadow-md"

>

Logout

</button>

);

}

// --- Content Components for Different Views ---

// This component displays the flight list for guest users to browse.

function FlightList() {

return (

<div className="mt-6 w-full max-w-lg">

<h2 className="text-2xl font-semibold text-gray-700 mb-4">Available Flights</h2>

<div className="space-y-4">

{flightData.map(flight => (

<div key={flight.id} className="bg-white p-4 rounded-lg shadow border border-gray-200">

<p className="font-bold text-lg text-blue-800">{flight.id}</p>

<p className="text-gray-600">{flight.from} to {flight.to}</p>

<p className="text-gray-500 text-sm">Departure: {flight.time}</p>

</div>

))}

</div>

</div>

);

}

// This component displays the ticket booking interface for logged-in users.

function BookingSection() {

const bookTicket = () => {

alert('Your ticket has been booked successfully!');

}

return (

<div className="mt-6 text-center">

<h2 className="text-2xl font-semibold text-gray-700 mb-4">Ready to book your next trip?</h2>

<button

onClick={bookTicket}

className="bg-green-600 text-white font-bold py-3 px-8 rounded-lg hover:bg-green-700 transition duration-300 ease-in-out text-lg shadow-lg"

>

Book a Ticket Now

</button>

</div>

);

}

// --- Main App Component ---

// This is the main component that controls the application's state and logic.

export default function App() {

// useState hook to keep track of the user's login status.

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

// Event handler to set isLoggedIn to true when the user clicks "Login".

const handleLoginClick = () => {

setIsLoggedIn(true);

};

// Event handler to set isLoggedIn to false when the user clicks "Logout".

const handleLogoutClick = () => {

setIsLoggedIn(false);

};

return (

<div className="bg-gray-100 min-h-screen flex items-center justify-center font-sans">

<div className="w-full max-w-2xl mx-auto p-8 bg-gray-50 rounded-xl shadow-2xl text-center">

{/\*

Conditional (Ternary) Operator:

This is the core of the conditional rendering logic.

- If `isLoggedIn` is true, it renders the UserGreeting.

- If `isLoggedIn` is false, it renders the GuestGreeting.

\*/}

{isLoggedIn ? <UserGreeting /> : <GuestGreeting />}

{/\*

Content Area:

We use the same logic to display the main content.

- If logged in, show the BookingSection.

- If not logged in, show the FlightList.

\*/}

{isLoggedIn ? <BookingSection /> : <FlightList />}

<div className="mt-8">

{/\*

Button Area:

Finally, we conditionally render the correct button.

- If logged in, show the LogoutButton.

- If not logged in, show the LoginButton.

\*/}

{isLoggedIn ? (

<LogoutButton onClick={handleLogoutClick} />

) : (

<LoginButton onClick={handleLoginClick} />

)}

</div>

</div>

</div>

);

}

OUTPUT:



