## **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:**







**Solution :**

**1. Project Setup**

In your terminal:

npx create-react-app ticketbookingapp

cd ticketbookingapp

Inside the src folder, create a folder named components and add the following files:

* GuestPage.js
* UserPage.js
* FlightList.js
* LoginToggle.js

2. FlightList.js

import React from 'react';

const flights = [

{ id: 1, origin: "Chennai", destination: "Delhi", price: 5000 },

{ id: 2, origin: "Bangalore", destination: "Mumbai", price: 4500 },

{ id: 3, origin: "Hyderabad", destination: "Kolkata", price: 6000 }

];

function FlightList({ canBook }) {

return (

<div>

<h3>Flight Details</h3>

<ul>

{flights.map(flight => (

<li key={flight.id}>

✈️ {flight.origin} → {flight.destination} - ₹{flight.price}

{canBook && <button style={{ marginLeft: '10px' }}>Book</button>}

</li>

))}

</ul>

</div>

);

}

export default FlightList;

3. GuestPage.js

import React from 'react';

import FlightList from './FlightList';

function GuestPage() {

return (

<div>

<h2>Welcome, Guest!</h2>

<p>Please login to book your tickets.</p>

<FlightList canBook={false} />

</div>

);

}

export default GuestPage;

4. UserPage.js

import React from 'react';

import FlightList from './FlightList';

function UserPage() {

return (

<div>

<h2>Welcome, User!</h2>

<p>You can now book your tickets below.</p>

<FlightList canBook={true} />

</div>

);

}

export default UserPage;

5. LoginToggle.js

import React from 'react';

function LoginToggle({ isLoggedIn, onLoginToggle }) {

return (

<button onClick={onLoginToggle}>

{isLoggedIn ? 'Logout' : 'Login'}

</button>

);

}

export default LoginToggle;

**6. App.js**

Update src/App.js to manage the login state

import React, { useState } from 'react';

import './App.css';

import GuestPage from './components/GuestPage';

import UserPage from './components/UserPage';

import LoginToggle from './components/LoginToggle';

function App() {

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

const handleLoginToggle = () => {

setIsLoggedIn(prev => !prev);

};

return (

<div className="App">

<h1>🎫 Ticket Booking App</h1>

<LoginToggle isLoggedIn={isLoggedIn} onLoginToggle={handleLoginToggle} />

<hr />

{isLoggedIn ? <UserPage /> : <GuestPage />}

</div>

);

}

export default App;

7. Run the App

npm start