## **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:

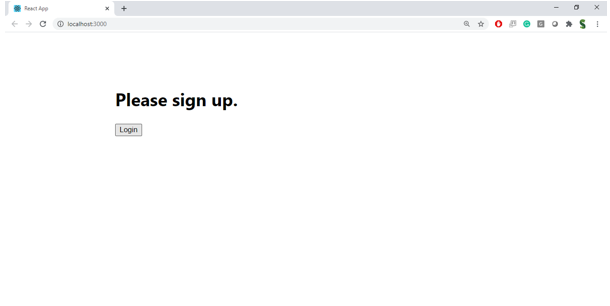
* Node.js
* NPM
* Visual Studio Code

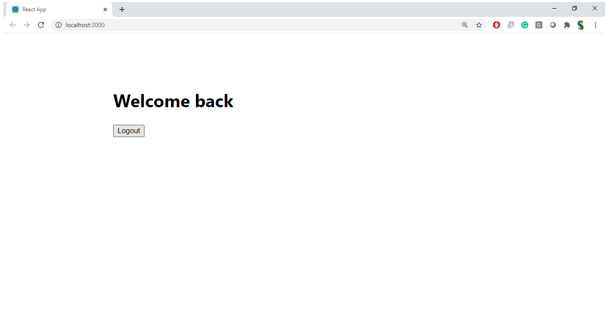
## **Notes**

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

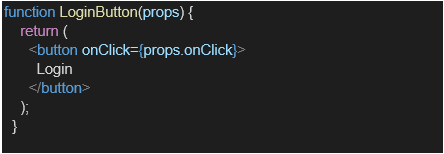
Create a React Application named “ticketbookingapp” where the guest user can browse the page where the flight details are displayed whereas the logged in user only can book tickets.

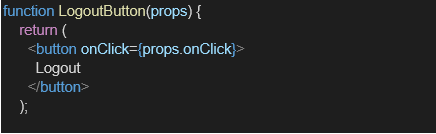
The Login and Logout buttons should accordingly display different pages. Once the user is logged in the User page should be displayed. When the user clicks on Logout, the Guest page should be displayed.

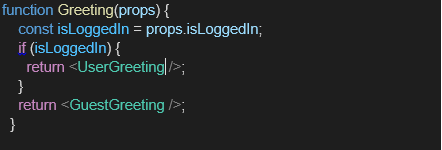




**Hint:**







**SOURCE CODE**

**File: App.js**

import React, { useState } from 'react';

// LogoutButton component

function LogoutButton(props) {

return <button style={styles.button} onClick={props.onClick}>Sign Out</button>;

}

// Greeting component

function Greeting(props) {

const isLoggedIn = props.isLoggedIn;

return (

<h1 style={styles.heading}>

{isLoggedIn ? 'Welcome Back' : 'Please Sign Up'}

</h1>

);

}

function App() {

const [isLoggedIn, setIsLoggedIn] = useState(false); // Tracks login status

// Sample flight details

const flightDetails = [

{ id: 1, destination: "New York", price: "$300" },

{ id: 2, destination: "Los Angeles", price: "$250" },

{ id: 3, destination: "London", price: "$400" }

];

// Handles login action

const handleLogin = () => {

setIsLoggedIn(true);

};

// Handles logout action

const handleLogout = () => {

setIsLoggedIn(false);

};

return (

<div className="App">

{/\* Guest Page (if not logged in) \*/}

{!isLoggedIn ? (

<div style={styles.container}>

<Greeting isLoggedIn={isLoggedIn} />

<button style={styles.button} onClick={handleLogin}>Login</button>

{/\* Flight Details visible only for Guest \*/}

<div style={styles.flightDetails}>

<h3>Available Flights:</h3>

<ul>

{flightDetails.map(flight => (

<li key={flight.id}>

{flight.destination} - {flight.price}

</li>

))}

</ul>

</div>

</div>

) : (

// User Page (if logged in)

<div style={styles.container}>

<Greeting isLoggedIn={isLoggedIn} />

<LogoutButton onClick={handleLogout} />

{/\* Booking section visible only for logged-in users \*/}

<div style={styles.booking}>

<h3>Book Your Ticket:</h3>

<button style={styles.button}>Book Now</button>

</div>

</div>

)}

</div>

);

}

// Inline CSS Styles

const styles = {

container: {

display: 'flex',

flexDirection: 'column',

justifyContent: 'center',

alignItems: 'center',

height: '100vh',

backgroundColor: '#f0f8ff',

},

heading: {

fontWeight: 'bold',

fontSize: '24px',

textAlign: 'center',

},

button: {

padding: '10px 20px',

fontSize: '16px',

marginTop: '20px',

cursor: 'pointer',

backgroundColor: '#007BFF',

color: '#fff',

border: 'none',

borderRadius: '5px',

},

flightDetails: {

marginTop: '20px',

textAlign: 'left',

padding: '10px',

backgroundColor: '#fff',

borderRadius: '5px',

width: '80%',

},

booking: {

marginTop: '20px',

textAlign: 'center',

},

};

export default App;

**OUTPUT:**

