## **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:**







**App.js**

import React, { useState } from 'react';

function App() {

const [count, setCount] = useState(0);

const [rupees, setRupees] = useState('');

const [euro, setEuro] = useState('');

// Multiple methods on increment

const increment = () => {

sayHello(); // Method 1

setCount(prev => prev + 1); // Method 2

};

const sayHello = () => {

alert("Hello! This is a static message.");

};

const decrement = () => {

setCount(prev => prev - 1);

};

// Welcome with argument

const sayWelcome = (msg) => {

alert(msg);

};

// Synthetic event

const onPressHandler = (event) => {

alert("I was clicked");

console.log("Synthetic Event:", event);

};

// Currency Converter

const handleSubmit = (e) => {

e.preventDefault();

const converted = parseFloat(rupees) \* 0.011; // Simple fixed rate

setEuro(converted.toFixed(2));

};

return (

<div style={{ padding: '20px' }}>

<h1>Event Handling Example</h1>

<h2>Counter: {count}</h2>

<button onClick={increment}>Increment</button>

<button onClick={decrement}>Decrement</button>

<br /><br />

<button onClick={() => sayWelcome("Welcome!")}>Say Welcome</button>

<br /><br />

<button onClick={onPressHandler}>Click Me (Synthetic Event)</button>

<br /><br />

<h2>Currency Convertor</h2>

<form onSubmit={handleSubmit}>

<label>Enter amount in INR:</label><br />

<input

type="number"

value={rupees}

onChange={(e) => setRupees(e.target.value)}

/>

<br /><br />

<button type="submit">Convert to Euro</button>

</form>

{euro && <p>Equivalent Euro: €{euro}</p>}

</div>

);

}

export default App;

**index.css**

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

background-color: #f5f5f5;

}

h1 {

text-align: center;

color: #333;

}

h2 {

color: #444;

}

div {

max-width: 600px;

margin: 40px auto;

background-color: white;

padding: 30px;

border-radius: 10px;

box-shadow: 0 0 15px rgba(0, 0, 0, 0.1);

}

button {

margin: 5px;

padding: 10px 20px;

font-size: 16px;

border: none;

border-radius: 6px;

background-color: #007bff;

color: white;

cursor: pointer;

}

button:hover {

background-color: #0056b3;

}

input[type="number"] {

padding: 8px;

font-size: 16px;

margin-top: 5px;

width: 100%;

box-sizing: border-box;

}

**OUTPUT:**



