Digital-Nurture-4.0-JavaFSE-main

Hands-on in this document

Name: Thilak Ragav M

Superset ID : 6409405

Week Task : Week - 7

## **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:**







Index.js:

import React from 'react';import ReactDOM from 'react-dom/client';import './index.css';import App from './App';import reportWebVitals from './reportWebVitals';const root = ReactDOM.createRoot(document.getElementById('root'));root.render( <React.StrictMode> <App /> </React.StrictMode>);// If you want to start measuring performance in your app, pass a function// to log results (for example: reportWebVitals(console.log))// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitalsreportWebVitals();

Index.css:

body { margin: 0; font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen', 'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue', sans-serif; -webkit-font-smoothing: antialiased; -moz-osx-font-smoothing: grayscale;}code { font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New', monospace;}App.js

import React, { useState } from 'react';import Greeting from './components/Greeting';import LoginButton from './components/LoginButton';import LogoutButton from './components/LogoutButton';function App() { const [isLoggedIn, setIsLoggedIn] = useState(false); function handleLoginClick() { setIsLoggedIn(true); } function handleLogoutClick() { setIsLoggedIn(false); } return ( <div style={{ textAlign: "center", marginTop: "100px" }}> <Greeting isLoggedIn={isLoggedIn} /> {isLoggedIn ? ( <LogoutButton onClick={handleLogoutClick} /> ) : ( <LoginButton onClick={handleLoginClick} /> )} </div> );}export default App;

Report Web Vitals:

const reportWebVitals = onPerfEntry => { if (onPerfEntry && onPerfEntry instanceof Function) { import('web-vitals').then(({ getCLS, getFID, getFCP, getLCP, getTTFB }) => { getCLS(onPerfEntry); getFID(onPerfEntry); getFCP(onPerfEntry); getLCP(onPerfEntry); getTTFB(onPerfEntry); }); }};export default reportWebVitals;

Package.json:

{ "name": "ticketbookingapp", "version": "0.1.0", "private": true, "dependencies": { "@testing-library/dom": "^10.4.0", "@testing-library/jest-dom": "^6.6.3", "@testing-library/react": "^16.3.0", "@testing-library/user-event": "^13.5.0", "react": "^19.1.0", "react-dom": "^19.1.0", "react-scripts": "5.0.1", "web-vitals": "^2.1.4" }, "scripts": { "start": "react-scripts start", "build": "react-scripts build", "test": "react-scripts test", "eject": "react-scripts eject" }, "eslintConfig": { "extends": [ "react-app", "react-app/jest" ] }, "browserslist": { "production": [ ">0.2%", "not dead", "not op\_mini all" ], "development": [ "last 1 chrome version", "last 1 firefox version", "last 1 safari version" ] }}

Output:





