**Explain about conditional rendering in React**

**Definition:**  
Conditional rendering means displaying different UI elements based on certain conditions, like user state, props, or values.

**Examples:**

**Using if/else:**

if (isLoggedIn) {

return <h1>Welcome back!</h1>;

} else {

return <h1>Please log in.</h1>;

}

**Using ternary operator:**

<h1>{isLoggedIn ? "Welcome back!" : "Please log in."}</h1>

**Using && operator:**

{hasItems && <p>You have items in your cart.</p>}

**Define element variables**

**Definition:**  
Element variables store JSX elements in variables, making it easier to return different elements conditionally.

**Example:**

let message;

if (isLoggedIn) {

message = <h1>Welcome back!</h1>;

} else {

message = <h1>Please sign up.</h1>;

}

return <div>{message}</div>;

* message is an **element variable** storing JSX.

**Explain how to prevent components from rendering**

**A. Return null from a component:**

function WarningBanner(props) {

if (!props.showWarning) {

return null; // prevents rendering

}

return <div className="warning">Warning!</div>;

}

**B. Conditional render with &&:**

{shouldShow && <MyComponent />}

**C. Early return in render():**

render() {

if (!this.props.visible) {

return null;

}

return <div>Visible content</div>;

}

**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.**

**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>

);

reportWebVitals();

**App.js :-**

import React, { Component } from 'react';

class App extends Component {

  constructor(props) {

    super(props);

    this.state = {

      isLoggedIn: false, // Initial state - user is not logged in

    };

    this.handleLoginClick = this.handleLoginClick.bind(this);

    this.handleLogoutClick = this.handleLogoutClick.bind(this);

  }

  handleLoginClick() {

    this.setState({ isLoggedIn: true });

  }

  handleLogoutClick() {

    this.setState({ isLoggedIn: false });

  }

  render() {

    const { isLoggedIn } = this.state;

    // Element variable for content

    let pageContent;

    if (isLoggedIn) {

      pageContent = <UserPage />;

    } else {

      pageContent = <GuestPage />;

    }

    return (

      <div style={{ textAlign: 'center', marginTop: '100px' }}>

        {pageContent}

        <br />

        {

          isLoggedIn ? (

            <button onClick={this.handleLogoutClick}>Logout</button>

          ) : (

            <button onClick={this.handleLoginClick}>Login</button>

          )

        }

      </div>

    );

  }

}

function GuestPage() {

  return (

    <div>

      <h1>Please sign up.</h1>

      <p>Browse available flight schedules below.</p>

      {/\* Example Flight Info \*/}

      <ul>

        <li>Chennai → Delhi - 9:00 AM</li>

        <li>Mumbai → Bangalore - 1:00 PM</li>

      </ul>

    </div>

  );

}

function UserPage() {

  return (

    <div>

      <h1>Welcome back</h1>

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

      {/\* Booking Example \*/}

      <button>Book Flight</button>

    </div>

  );

}

export default App;

**Output :-**





