**WEEK-7\_ReactJs\_HOL**

**SUPERSET ID : 6393674**

**1. Explain about conditional rendering in React**

Conditional rendering in React means displaying components or elements based on certain conditions (like login status or user role). It uses JavaScript logic such as if, ternary (?:), or && inside JSX to decide what to render.

**Example:**

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

**2. Define element variables**

Element variables are used in React to store JSX content in a variable. This helps conditionally assign elements and improve code readability before rendering.

**Example:**

let content;

if (isLoggedIn) {

content = <UserPage />;

} else {

content = <GuestPage />;

}

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

**3. Explain how to prevent components from rendering**

You can prevent a component from rendering by returning null from it. When null is returned, nothing is displayed on the UI and no DOM changes are made.

**Example:**

function Warning(props) {

if (!props.show) {

return null;

}

return <div>Warning Message</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.**

**App.js**

import React, { useState } from 'react';

function App() {

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

const handleLoginClick = () => {

setIsLoggedIn(true);

};

const handleLogoutClick = () => {

setIsLoggedIn(false);

};

let message;

if (isLoggedIn) {

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

} else {

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

}

return (

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

{message}

{!isLoggedIn ? (

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

) : (

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

)}

</div>

);

}

export default App;

**OUTPUT:**



