**1. Explain Various Ways of Conditional Rendering**

React supports several ways to conditionally render components:

**a. if/else Statement:**

if (isLoggedIn) {

return <Welcome />;

} else {

return <Login />;

}

**b. Ternary Operator:**

return (

isLoggedIn ? <Dashboard /> : <GuestHome />

);

**c. Logical AND (&&) Operator:**

return (

<div>

{hasPermission && <AdminTools />}

</div>

);

**d. Using Switch Statements:**

switch(userType) {

case 'admin': return <Admin />;

case 'user': return <User />;

default: return <Guest />;

}

**2. Explain How to Render Multiple Components**

You can render multiple components in two main ways:

**a. Inside a parent component:**

return (

<div>

<Header />

<Main />

<Footer />

</div>

);

**b. Using React Fragments (to avoid extra <div>):**

return (

<>

<Nav />

<Content />

<Sidebar />

</>

);

**3. Define List Component**

A **list component** displays a collection of items using .map() and renders them as a list of elements.

**Example:**

function NameList(props) {

const names = props.names;

return (

<ul>

{names.map((name) => <li key={name}>{name}</li>)}

</ul>

);

}

**4. Explain About Keys in React Applications**

* **Keys** are unique identifiers assigned to list elements to help React identify which items changed, added, or removed.
* Improves performance and stability in re-rendering lists.

**Good Practice:** Use a unique ID from data (not array index if items can change).

{users.map(user => <li key={user.id}>{user.name}</li>)}

**5. Explain How to Extract Components with Keys**

You can extract individual list items into separate components and pass the key when rendering them.

**Example:**

function NameItem(props) {

return <li>{props.name}</li>;

}

function NameList(props) {

return (

<ul>

{props.names.map((n) => (

<NameItem key={n.id} name={n.name} />

))}

</ul>

);

}

**6. Explain React Map, map() Function**

* The .map() function is a JavaScript array method used in React to transform and render arrays of elements or components.
* Returns a new array without modifying the original.

**Example:**

const numbers = [1, 2, 3];

const listItems = numbers.map((num) => <li key={num}>{num}</li>);

* Commonly used to render dynamic lists in UI from data arrays.