**1. Explain various ways of conditional rendering**

In React, we can conditionally render components or elements using different techniques:

a) Using if-else:

if (isLoggedIn) {

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

} else {

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

}

b) Using ternary operator:

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

c) Using logical AND (&&):

{isLoggedIn && <Dashboard />}

d) Using element variables:

let content;

if (isLoggedIn) {

content = <LogoutButton />;

} else {

content = <LoginButton />;

}

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

**2. Explain how to render multiple components**

We can render multiple components by including them inside a parent component’s JSX.

Example:

function App() {

return (

<div>

<Header />

<Main />

<Footer />

</div>

);

}

Alternatively, use fragments (<>...</>) to avoid adding extra DOM nodes:

<>

<Navbar />

<Content />

</>

**3. Define list component**

A list component in React is a component that renders a collection of items, usually using the JavaScript map() function.

Example:

function NumberList(props) {

const numbers = props.numbers;

return (

<ul>

{numbers.map((num) => (

<li key={num}>{num}</li>

))}

</ul>

);

}

**4. Explain about keys in React applications**

Keys are special attributes used to help React identify which items in a list have changed, been added, or removed.

Rules:

* Keys should be unique among siblings.
* Prefer using a unique ID from your data, not the array index.

Example:

const listItems = items.map((item) =>

<li key={item.id}>{item.name}</li>

);

**5. Explain how to extract components with keys**

When rendering lists, you can extract list items into a separate component and pass the key to the component when it is used, not inside the component itself.

function ListItem(props) {

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

}

function NumberList(props) {

const numbers = props.numbers;

return (

<ul>

{numbers.map((num) => (

<ListItem key={num.toString()} value={num} />

))}

</ul>

);

}

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

React does not have its own map() function; it uses JavaScript's Array.prototype.map() to transform arrays into elements.