1. Explain Various Ways of Conditional Rendering in React

React provides several ways to render components or elements conditionally based on state or props:

Method	Example
if-else	if (loggedIn) { return <Dashboard /> } else { return <Login /> }
Element variables	let content = isAdmin ? <AdminPanel /> : <UserPanel />;
Ternary operator	{loggedIn ? <Welcome /> : <Login />}
Logical AND (&&)	{isLoggedIn && <LogoutButton />}
Immediately Invoked Function Expression (IIFE)	{(() => isAdmin ? <Admin /> : <User />)()}

---

2. Explain How to Render Multiple Components

You can render multiple components together in JSX like this:

```
function App() {
  return (
    <div>
      <Header />
      <MainContent />
      <Footer />
    </div>
  );
}
```

If you don’t want to wrap them in a , use **React fragments**:

```
<>
  <Header />
  <Main />
  <Footer />
</>
```

3. Define List Component

A **List Component** is a React component that renders a list of items dynamically using the map() function.

- Example:

```
function BookList() {
  const books = ["React", "Angular", "Vue"];
  return (
    <ul>
      {books.map(book => <li key={book}>{book}</li>)}
    </ul>
  );
}
```

4. Explain About Keys in React Applications

Keys help React identify which items in a list have changed, been added, or removed.

- Example:

```
const items = ['One', 'Two', 'Three'];
const listItems = items.map((item, index) => <li key={index}>{item}</li>);
```

Best Practice:

- Use a **unique and stable key** (like an ID from the database).
- Avoid using index as key unless items never change order.

5. Explain How to Extract Components with Keys

If a list item is complex, move it into a child component and pass a key when rendering.

- Example:

```
function User({ name }) {
  return <li>{name}</li>;
}

function UserList() {
  const users = ["Alice", "Bob"];
```

```
  return (
    <ul>
      {users.map((user, index) => (
        <User key={user} name={user} />
      ))}
    </ul>
  );
}
```

6. Explain React Map and map() Function

- React uses the **JavaScript Array.prototype.map()** function to transform arrays into JSX elements.
- Each item is converted into a component or HTML element.

🔹 Example:

```
const fruits = ['Apple', 'Banana', 'Mango'];


function FruitList() {
  return (
    <ul>
      {fruits.map((fruit, index) => (
        <li key={index}>{fruit}</li>
      ))}
    </ul>
  );
}
```