Conditional Rendering

Question 1: What is conditional rendering in React? How can you conditionally render elements in a React component?

Ans.

Conditional rendering in React means showing or hiding UI elements based on a condition (like a value in state, props, or a variable). It's similar to using if / else in normal JavaScript—but applied to JSX.

Why conditional rendering?

You use it when:

- User is logged in / logged out
- Data is loading
- Showing error / success messages
- Toggling UI (modal, dropdown, button text)

Common ways to do conditional rendering in React

1. **if / else statement (outside JSX)**

```
function App() {
  const isLoggedIn = true;

  if (isLoggedIn) {
    return <h1>Welcome User</h1>;
  } else {
    return <h1>Please Login</h1>;
  }
}
```

2. **Ternary Operator (condition ? true : false)**

```
function App() {
  const isLoggedIn = false;

  return (
    <div>
      {isLoggedIn ? <h1>Welcome</h1> : <h1>Please Login</h1>}
    </div>
  );
}
```

3. **Logical AND (&&)**

```
function App() {
  const showMessage = true;

  return (
    <div>
      {showMessage && <p>Hello World!</p>}
    </div>
  );
}
```

4. **Using Variables**
   Good for cleaner JSX.

```
function App() {
  const isAdmin = true;
  let message;

  if (isAdmin) {
    message = <h1>Admin Panel</h1>;
  } else {
    message = <h1>User Panel</h1>;
  }

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

5. **Conditional rendering with state (real-world example)**

```
import { useState } from "react";

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

  return (
    <div>
```

```
   {isLoggedIn ? <h2>Welcome Back</h2> : <h2>Please
Login</h2>}

   <button onClick={() => setIsLoggedIn(!isLoggedIn)}>
    Toggle Login
   </button>
  </div>
 );
}
```

Question 2: Explain how if-else, ternary operators, and && (logical AND) are used in JSX for conditional rendering.

Ans.

In React, JSX allows JavaScript expressions, so you can use conditions directly to decide what to render.
Let's clearly break down if-else, ternary operator, and logical AND (&&), with when and how to use each.

1. **if-else in JSX (used *outside* JSX)**
   You cannot write if-else directly inside JSX.
   Instead, use it before return or assign JSX to a variable.

```
function App() {
  const isLoggedIn = true;

  if (isLoggedIn) {
    return <h1>Welcome User</h1>;
  } else {
    return <h1>Please Login</h1>;
  }
}
```

Using a variable :

```
function App() {
  const isAdmin = false;
  let content;

  if (isAdmin) {
    content = <h2>Admin Dashboard</h2>;
  } else {
    content = <h2>User Dashboard</h2>;
  }

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

2. **Ternary Operator (condition ? true : false)**

Ternary Operator (condition ? true : false)

Syntax :

```
{condition ? JSX_if_true : JSX_if_false}
```

```
function App() {
  const isLoggedIn = false;
  return (
    <div>
      {isLoggedIn ? <h1>Welcome</h1> : <h1>Please
Login</h1>}
    </div>
  );
}
```

<u>With state :</u>

```
{isLoading ? <p>Loading...</p> : <p>Data Loaded</p>}
```

3. **Logical AND (&&)**
   Used when you want to render something only if the condition is true.

   Syntax :
   {condition && JSX}

   Example :

```
function App() {
  const showMessage = true;

  return (
    <div>
      {showMessage && <p>Hello React!</p>}
    </div>
  );
}
```

Real-world use case :

```
{error && <p style={{ color: "red" }}>{error}</p>}
```

Comparison :

Method	Use Case	Can be inside JSX?
if-else	Complex logic	No
Ternary ? :	Yes / No UI	Yes
&&	Show only if true	Yes

Which one should you use?

- Complex conditions → if-else
- Two options → Ternary operator
- One-way rendering → &&
