Chapter 6: Conditional Rendering

1 Introduction

Is chapter mein humne **conditional rendering** ke baare mein seekha, jo React mein UI ko conditions ke hisaab se show ya hide karne ke liye use hota hai. Humne if-else, ternary operator, aur logical AND operator ka use dekha.

2 6.1 What is Conditional Rendering?

Conditional Rendering ka matlab hai UI ke parts ko state, props, ya user input ke hisaab se show ya hide karna. Yeh JSX mein JavaScript expressions ke through hota hai.

2.1 Real-World Example

Ek food delivery app mein, agar cart khali hai, toh "Cart is Empty" dikhao, warna cart items ki list dikhao.

3 6.2 Conditional Rendering Techniques

React mein teen main tareeke hain:

- **If-Else**: JSX ke bahar use hota hai.
- **Ternary Operator** (`?:`): JSX ke andar short conditions ke liye.
- **Logical AND** (`&&`): Simple show/hide ke liye.

3.1 6.2.1 Using If-Else (Outside JSX)

If-else statements JSX ke andar directly nahi likhe ja sakte, lekin function ke andar use karke JSX return kar sakte hain.

3.1.1 Code Example: If-Else

```
function CartStatus({ items }) {
  if (items.length === 0) {
    return (
      <div className="text-center p-4">
        <h1 className="text-2xl font-bold text-red-600">Cart is Empty</h1>
      </div>
    );
  } else {
    return (
      <div className="text-center p-4">
        <h1 className="text-2xl font-bold text-green-600">Your Cart</h1>
        <p>{items.length} items in cart</p>
      </div>
    );
  }
}
```

3.1.2 Explanation

- Agar `items` array khali hai, toh "Cart is Empty" dikhega.
- Warna, item count dikhega.

3.2 6.2.2 Using Ternary Operator (Inside JSX)

Ternary operator (`?:`) JSX ke andar use hota hai kyunki yeh single expression hai.

3.2.1 Code Example: Ternary Operator

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Ternary Conditional Rendering</title>
  <script src="https://cdn.jsdelivr.net/npm/react@18.2.0/umd/react.development.js
  <script src="https://cdn.jsdelivr.net/npm/react-dom@18.2.0/umd/react-dom.develo
  <script src="https://cdn.jsdelivr.net/npm/@babel/standalone@7.20.6/babel.min.js
  <script src="https://cdn.tailwindcss.com"></script>
</head>
<body>
  <div id="root"></div>
  <script type="text/babel">
    function CartStatus() {
      const [items, setItems] = React.useState([]);

      const addItem = () => {
        setItems([...items, { id: items.length + 1, name: ‘Item ${items.length + 1}‘ }]);
      };

      return (
        <div className="text-center p-4">
          <h1 className="text-3xl font-bold text-blue-600">Cart Status</h1>
          {items.length === 0 ? (
            <h2 className="text-2xl text-red-600">Cart is Empty</h2>
          ) : (
            <div>
              <h2 className="text-2xl text-green-600">Your Cart</h2>
              <p className="text-lg">{items.length} items in cart</p>
            </div>
          )}
          <button
            className="bg-blue-500 text-white px-4 py-2 m-2 rounded"
            onClick={addItem}
          >
            Add Item
          </button>
        </div>
      );
    }

    const root = ReactDOM.createRoot(document.getElementById('root'));
    root.render(<CartStatus />);
  </script>
</body>
</html>
```

3.2.2 Explanation

- `items.length === 0 ? ... : ...`: Ternary operator check karta hai cart khali hai ya nahi.
- `addItem`: Dummy item add karta hai.
- Output: Initially "Cart is Empty", button click pe item count.

3.3 6.2.3 Using Logical AND Operator (&&)

Logical AND (&&) simple show/hide ke liye use hota hai.

3.3.1 Code Example: Logical AND

```
function WelcomeMessage({ isLoggedIn }) {
  return (
    <div className="text-center p-4">
      <h1 className="text-3xl font-bold text-blue-600">Welcome</h1>
      {isLoggedIn && (
        <p className="text-lg text-green-600">Hello, User! You're logged in!</p>
      )}
      {!isLoggedIn && (
        <p className="text-lg text-red-600">Please log in to continue.</p>
      )}
    </div>
  );
}
```

3.3.2 Explanation

- `isLoggedIn && ...`: Agar `isLoggedIn` true hai, toh green message dikhega.
- `!isLoggedIn && ...`: Agar false hai, toh red message.

4 6.3 Real-World Example: Login/Logout Toggle

Ek app jisme login/logout ke hisaab se UI change hota hai.

4.1 Code Example: Login Toggle

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Login Toggle</title>
  <script src="https://cdn.jsdelivr.net/npm/react@18.2.0/umd/react.development.js
  <script src="https://cdn.jsdelivr.net/npm/react-dom@18.2.0/umd/react-dom.develo
  <script src="https://cdn.jsdelivr.net/npm/@babel/standalone@7.20.6/babel.min.js
  <script src="https://cdn.tailwindcss.com"></script>
</head>
<body>
  <div id="root"></div>
  <script type="text/babel">
    function LoginApp() {
      const [isLoggedIn, setIsLoggedIn] = React.useState(false);

      const toggleLogin = () => {
        setIsLoggedIn(!isLoggedIn);
      };

      return (
        <div className="text-center p-4">
          <h1 className="text-3xl font-bold text-blue-600">Login App</h1>
          {isLoggedIn ? (
            <div>
              <p className="text-lg text-green-600">Welcome, User!</p>
              <button
                className="bg-red-500 text-white px-4 py-2 m-2 rounded"
                onClick={toggleLogin}
              >
                Logout
              </button>
            </div>
          ) : (
            <div>
              <p className="text-lg text-red-600">Please log in.</p>
              <button
                className="bg-green-500 text-white px-4 py-2 m-2 rounded"
                onClick={toggleLogin}
              >
                Login
              </button>
            </div>
          )}
        </div>
      );
    }

    const root = ReactDOM.createRoot(document.getElementById('root'));
    root.render(<LoginApp />);
  </script>
</body>
</html>
```

4.2 Explanation

- `isLoggedIn`: State track karta hai user logged in hai ya nahi.
- `toggleLogin`: Login/logout toggle karta hai.
- Ternary operator: Login ya logout UI dikhata hai.
- Output: Initially "Please log in.", click pe "Welcome, User!".

5 Common Mistakes

- **JSX Mein Direct If-Else**: JSX mein `if` statements nahi likhe ja sakte.
- **Complex Logic in JSX**: Logic ko component ke andar rakho.
- **Forgetting Parent Element**: Conditional output ko `<div>` ya `<>` mein wrap karo.

6 Interview Tips

- **Conditional rendering kya hai?**
  - UI ko conditions ke hisaab se show/hide karna, using if-else, ternary, ya `&&`.
- **Ternary vs Logical AND?**
  - Ternary do outcomes ke liye, `&&` sirf true condition ke liye.
- **Kab if-else use karte hain?**

  - Complex logic ya multiple JSX blocks ke liye, component ke andar.

7 Assignment: Practice Time

7.1 Task 1: Empty Cart Check

- `Cart` component banao jo `useState` se items array track kare.
- Khali hai toh "Cart is Empty", warna item count dikhao.
- Button se dummy item add karo.
- Tailwind se style karo.

7.2 Task 2: Login Status

- `UserStatus` component banao jo `isLoggedIn` state track kare.
- Logged in hai toh "Welcome Back!", warna "Please Log In".
- Button se login/logout toggle karo.
- Tailwind se style karo.

7.3 Task 3: Show/Hide Details

- `ProductCard` component banao jo `showDetails` state track kare.
- Button se details show/hide toggle karo.
- `showDetails` true hai toh extra info dikhao.
- Tailwind se style karo.