

## **Report Day: 6.4**

**Submitted By: Ayesha Abid**

**Submitted To: Mr. Ali Hyder Hidayat**

**Group: Frontend Development @ProSensia**

**Dated: 31/07/25**

### **Task: Conditional Rendering and list with .map()**

#### **Introduction:**

React is a JavaScript library used to build interactive and dynamic user interfaces, especially single-page applications (SPAs). Two important concepts in React development are conditional rendering and list rendering using `.map()`. These help developers create flexible UIs that can adapt based on user data and actions.

Conditional Rendering:

**Conditional rendering** in React means showing or hiding parts of the user interface based on certain conditions (like `true/false`, user login status, etc.). It works just like JavaScript `if` statements or conditional expressions, but it's used inside JSX.

React lets you decide **what to render** depending on the **state** or **props**. This helps make your interface smart and dynamic.

```
function Greeting(props) {  
  if (props.isLoggedIn) {  
    return <h2>Welcome back!</h2>;  
  } else {  
    return <h2>Please log in.</h2>;  
  }  
}
```

#### **List Rendering in React:**

React uses JavaScript's `.map()` method to render multiple elements (like a list of items). When we have a list of data (names, products, posts), we can loop through the list and show each item using `.map()`.

## **.map() Function:**

`.map()` is a built-in JavaScript function that takes every item in an array and runs a function on it returning a new array.

```
function NameList() {  
  const names = ['Ayesha', 'Ali', 'Sara'];  
  return (  
    <ul>  
      {names.map((name, index) => (  
        <li key={index}>{name}</li>  
      ))}  
    </ul>  
  );  
}
```

## **Benefits of List Rendering with .map()**

- Shows dynamic data from APIs or arrays
- Avoids repetitive hardcoding
- Works well with real-time data (like notifications or chat)

## **Conclusion:**

Conditional rendering and list rendering using `.map()` are core concepts in React. They allow developers to build smart, user-friendly interfaces that respond to the user's data and actions. By understanding these topics, you can create dynamic apps like to-do lists, login systems, or dashboards that display real-time data.