

Chapter 7: Lists and Keys

1 Introduction

Is chapter mein humne **lists** aur **keys** ke baare mein seekha. Lists ko React mein array ke **map** function se render karte hain, aur **keys** se React ko items track karne mein help milti hai.

2 7.1 Rendering Lists in React

Lists ko render karne ke liye **map** function ka use karke array ke items ko JSX elements mein convert karte hain.

2.1 Code Example: Simple List

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Simple List</title>
6   <script
7     src="https://cdn.jsdelivr.net/npm/react@18.2.0/umd/react.development.js
8   <script
9     src="https://cdn.jsdelivr.net/npm/react-dom@18.2.0/umd/react-dom.developo
10  <script
11     src="https://cdn.jsdelivr.net/npm/@babel/standalone@7.20.6/babel.min.js
12  <script src="https://cdn.tailwindcss.com"></script>
13 </head>
14 <body>
15   <div id="root"></div>
16   <script type="text/babel">
17     function MenuList() {
18       const menuItems = ["Chai", "Samosa", "Idli"];
19
20       return (
21         <div className="text-center p-4">
22           <h1 className="text-3xl font-bold text-blue-600">Our
23             Menu</h1>
24           <ul className="list-disc list-inside">
25             {menuItems.map((item) => (
26               <li key={item} className="text-lg">{item}</li>
27             ))}
28           </ul>
29         </div>
30       );
31     }
32
33     const root =
34       ReactDOM.createRoot(document.getElementById('root'));
35     root.render(<MenuList />);
```

```

31 </script>
32 </body>
33 </html>

```

2.2 Explanation

- `menuItems`: Array of strings.
- `map`: Har item ke liye `` element return karta hai.
- `key={item}`: Unique key taaki React track kar sake.
- Output: Bulleted list (Chai, Samosa, Idli).

2.3 Real-World Example

Food delivery app mein menu items ko cards ke roop mein render karna.

3 7.2 Why Keys are Important

Keys React ko batate hain ke list ke kaunse items unique hain, taaki rendering efficient ho. Keys unique aur stable hone chahiye.

3.1 Rules for Keys

- Unique within the list.
- Stable, yani same item ke liye same key.
- Array index ko key ke roop mein avoid karo.

3.2 Code Example: List with Objects

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Menu with Keys</title>
6   <script
7     src="https://cdn.jsdelivr.net/npm/react@18.2.0/umd/react.development.js"
8   >
9   <script
10    src="https://cdn.jsdelivr.net/npm/react-dom@18.2.0/umd/react-dom.development.js"
11  >
12  <script
13    src="https://cdn.jsdelivr.net/npm/@babel/standalone@7.20.6/babel.min.js"
14  >
15  <script src="https://cdn.tailwindcss.com"></script>
16 </head>
17 <body>
18   <div id="root"></div>
19   <script type="text/babel">
20     function MenuList() {
21       const menuItems = [
22         { id: 1, name: "Chai", price: 10 },

```

```

17     { id: 2, name: "Samosa", price: 20 },
18     { id: 3, name: "Idli", price: 30 }
19 ];
20
21 return (
22     <div className="text-center p-4">
23         <h1 className="text-3xl font-bold text-blue-600">Our
24         Menu</h1>
25         <div className="grid grid-cols-1 md:grid-cols-2 gap-4">
26             {menuItems.map((item) => (
27                 <div key={item.id} className="border p-4
28                 rounded-lg">
29                     <h2 className="text-xl
30                     font-bold">{item.name}</h2>
31                     <p>Price: Rs {item.price}</p>
32                 </div>
33             ))}
34         </div>
35     </div>
36 );
37
38 const root =
39     ReactDOM.createRoot(document.getElementById('root'));
40     root.render(<MenuList />);
41 </script>
42 </body>
43 </html>

```

3.3 Explanation

- menuItems: Array of objects (id, name, price).
- key={item.id}: Unique ID as key.
- Output: Cards with Chai (Rs 10), Samosa (Rs 20), Idli (Rs 30).

3.4 Why Not Use Index as Key?

Index unstable hota hai jab list reorder hoti hai, isse bugs aa sakte hain.

4 7.3 Combining Lists with State and useEffect

Dynamic lists ke liye useState aur useEffect ka use hota hai.

4.1 Code Example: Dynamic List

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">

```

```

5 <title>Dynamic List</title>
6 <script
  src="https://cdn.jsdelivr.net/npm/react@18.2.0/umd/react.development.js
7 <script
  src="https://cdn.jsdelivr.net/npm/react-dom@18.2.0/umd/react-dom.developo
8 <script
  src="https://cdn.jsdelivr.net/npm/@babel/standalone@7.20.6/babel.min.js
9 <script src="https://cdn.tailwindcss.com"></script>
10 </head>
11 <body>
12 <div id="root"></div>
13 <script type="text/babel">
14   function DynamicMenu() {
15     const [items, setItems] = React.useState([
16       { id: 1, name: "Chai", price: 10 }
17     ]);
18
19     const addItem = () => {
20       const newItem = {
21         id: items.length + 1,
22         name: `Item ${items.length + 1}`,
23         price: 10 * (items.length + 1)
24       };
25       setItems([...items, newItem]);
26     };
27
28     return (
29       <div className="text-center p-4">
30         <h1 className="text-3xl font-bold
31           text-blue-600">Dynamic Menu</h1>
32         <button
33           className="bg-green-500 text-white px-4 py-2 m-2
34             rounded"
35           onClick={addItem}
36         >
37           Add Item
38         </button>
39         <div className="grid grid-cols-1 md:grid-cols-2 gap-4">
40           {items.map((item) => (
41             <div key={item.id} className="border p-4
42               rounded-lg">
43               <h2 className="text-xl
44                 font-bold">{item.name}</h2>
45               <p>Price: Rs {item.price}</p>
46             </div>
47           ))}
48         </div>
49       </div>
50     );
51   };
52 }

```

```

49     const root =
        ReactDOM.createRoot(document.getElementById('root'));
50     root.render(<DynamicMenu />);
51 </script>
52 </body>
53 </html>

```

4.2 Explanation

- `useState`: `items` array manage karta hai.
- `addItem`: Naya item add karta hai.
- `map`: Items ko cards ke roop mein render karta hai.
- `key={item.id}`: Unique ID for tracking.
- Output: Initially one card, button click pe naye cards.

5 Common Mistakes

- **Keys Na Dena**: Bina `key` ke React warning deta hai.
- **Index as Key**: Index use karna avoid karo.
- **Non-Unique Keys**: Keys unique hone chahiye.

6 Interview Tips

- **Lists kaise render karte hain?**
 - `map` function se JSX elements return karte hain.
- **Keys kyun zaroori?**
 - React ko items track karne mein help karte hain.
- **Index kyun avoid karna?**
 - Index unstable hota hai, bugs cause kar sakta hai.

7 Assignment: Practice Time

7.1 Task 1: Simple List

- `FoodList` component banao jo array of strings render kare as bulleted list.
- Unique `key` use karo.
- Tailwind se style karo.

7.2 Task 2: Dynamic Cart

- `Cart` component banao jo `useState` se objects array track kare.
- Button se naya item add karo.
- Cards ke roop mein render karo, with `key`.
- Tailwind se style karo.

7.3 Task 3: Delete Item from List

- `TodoList` component banao jo `useState` se todos track kare.
- Delete button se todo remove karo.
- List ke roop mein render karo, with `key`.
- Tailwind se style karo.