

Chapter 9: Lifting State Up

1 Introduction

Is chapter mein humne **lifting state up** ke baare mein seekha. Yeh React ka ek important concept hai jab multiple components ko same state share karna hota hai. State ko parent component mein rakha jata hai aur props ke through child components tak pass kiya jata hai.

2 9.1 Why Lift State Up?

Jab do ya zyada components ko same data chahiye ya unke actions ek doosre ko affect karte hain, toh state ko unke common parent component mein rakha jata hai. Isse:

- Data ek jagah manage hota hai (single source of truth).
- Components ke beech sync maintain hota hai.
- Code predictable aur maintainable hota hai.

2.1 Real-World Example

Food order app mein, ek component cart items show karta hai aur doosra total price calculate karta hai. Dono ko same cart data chahiye, toh state parent component mein rakha jata hai.

3 9.2 Lifting State Up Example

Ek parent component state rakhta hai aur do child components usse use karte hain.

3.1 Code Example: Shared Cart State

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Lifting State Up</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     // Child Component 1: CartList
18     function CartList({ items, addItem, removeItem }) {
19       return (
```

```

17     <div className="p-4">
18         <h2 className="text-2xl font-bold text-blue-600">Cart
19           Items</h2>
20         <button
21           className="bg-green-500 text-white px-4 py-2 m-2
22             rounded"
23           onClick={addItem}
24         >
25           Add Item
26         </button>
27         <ul className="list-disc list-inside">
28           {items.map((item) => (
29             <li key={item.id} className="text-lg">
30               {item.name} - Rs {item.price}
31               <button
32                 className="ml-2 text-red-500"
33                 onClick={() => removeItem(item.id)}
34               >
35                 Remove
36               </button>
37             </li>
38           )))}
39         </ul>
40       </div>
41     );
42   }
43
44   // Child Component 2: CartTotal
45   function CartTotal({ items }) {
46     const total = items.reduce((sum, item) => sum +
47       item.price, 0);
48     return (
49       <div className="p-4">
50         <h2 className="text-2xl font-bold text-blue-600">Cart
51           Total</h2>
52         <p className="text-lg">Total Price: Rs {total}</p>
53       </div>
54     );
55   }
56
57   // Parent Component
58   function CartApp() {
59     const [items, setItems] = React.useState([
60       { id: 1, name: "Chai", price: 10 }
61     ]);
62
63     const addItem = () => {
64       const newItem = {
65         id: items.length + 1,
66         name: `Item ${items.length + 1}`,
67         price: 10 * (items.length + 1)
68       };
69       setItems([...items, newItem]);
70     };
71
72     const removeItem = (id) => {
73       const newItems = items.filter(item => item.id !== id);
74       setItems(newItems);
75     };
76
77     return (
78       <div>
79         <CartTotal items={items} />
80         <button onClick={addItem}>Add Item</button>
81         <button onClick={removeItem}>Remove Item</button>
82       </div>
83     );
84   }
85
86   export default CartApp;

```

```

64     };
65     setItems([...items, newItem]);
66 };
67
68 const removeItem = (id) => {
69     setItems(items.filter((item) => item.id !== id));
70 };
71
72 return (
73     <div className="text-center p-4">
74         <h1 className="text-3xl font-bold text-blue-600">Cart
75             App</h1>
76         <CartList items={items} addItem={addItem}
77             removeItem={removeItem} />
78         <CartTotal items={items} />
79     </div>
80 );
81 }
82
83 const root =
84     ReactDOM.createRoot(document.getElementById('root'));
85     root.render(<CartApp />);
86 </script>
87 </body>
88 </html>

```

3.2 Explanation

- CartApp (Parent): items state aur addItem, removeItem functions rakhta hai.
- CartList (Child): items props se list render karta hai aur buttons se state update karta hai.
- CartTotal (Child): items se total price calculate karta hai.
- Lifting State Up: State parent mein rakha aur props ke through share kiya.
- Output: Initially ek item, add/remove pe list aur total update hota hai.

4 9.3 Combining with Previous Concepts

Lifting state up ko useState, lists, aur forms ke saath combine kar sakte hain.

4.1 Code Example: Form + Lifting State Up

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Form with Lifting State</title>
6     <script
7         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.develop
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      // Child Component 1: AddItemForm
15      function AddItemForm({ addItem }) {
16        const [name, setName] = React.useState('');
17        const [price, setPrice] = React.useState('');
18
19        const handleSubmit = (event) => {
20          event.preventDefault();
21          if (name && price) {
22            addItem({ id: Date.now(), name, price: parseInt(price)
23                      });
24            setName('');
25            setPrice('');
26          }
27        };
28
29        return (
30          <form onSubmit={handleSubmit} className="p-4">
31            <h2 className="text-2xl font-bold text-blue-600">Add
32              Item</h2>
33            <input
34              type="text"
35              value={name}
36              onChange={(e) => setName(e.target.value)}
37              placeholder="Item name"
38              className="border p-2 m-2 rounded"
39            />
40            <input
41              type="number"
42              value={price}
43              onChange={(e) => setPrice(e.target.value)}
44              placeholder="Price"
45              className="border p-2 m-2 rounded"
46            />
47            <button
48              type="submit"
49              className="bg-green-500 text-white px-4 py-2 m-2
50                rounded"
51            >
52              Add
53            </button>
54          </form>
55        );

```

```

53     }
54
55     // Child Component 2: CartList
56     function CartList({ items }) {
57         return (
58             <div className="p-4">
59                 <h2 className="text-2xl font-bold text-blue-600">Cart
60                     Items</h2>
61                 <ul className="list-disc list-inside">
62                     {items.map((item) => (
63                         <li key={item.id} className="text-lg">
64                             {item.name} - Rs {item.price}
65                         </li>
66                     ))}
67                 </ul>
68             </div>
69         );
70     }
71
72     // Parent Component
73     function CartApp() {
74         const [items, setItems] = React.useState([]);
75
76         const addItem = (newItem) => {
77             setItems([...items, newItem]);
78         };
79
80         return (
81             <div className="text-center p-4">
82                 <h1 className="text-3xl font-bold text-blue-600">Cart
83                     App with Form</h1>
84                 <AddItemForm addItem={addItem} />
85                 <CartList items={items} />
86             </div>
87         );
88     }
89
90     const root =
91         ReactDOM.createRoot(document.getElementById('root'));
92     root.render(<CartApp />);
93 </script>
94 </body>
95 </html>

```

4.2 Explanation

- CartApp (Parent): items state aur addItem function rakhta hai.
- AddItemForm (Child): Form se input leta hai aur addItem call karta hai.
- CartList (Child): items props se list render karta hai.

- **Lifting State Up:** State parent mein, form se items add hote hain.
- **Output:** Form se item add hone pe list update hoti hai.

5 Common Mistakes

- **State ko Child mein Rakhna:** State child mein rakha toh doosra component access nahi kar sakta.
- **Props Pass Karna Bhoolna:** Event handlers ya state props ke through pass karo.
- **Unnecessary State Duplication:** Ek hi state parent mein rakho.

6 Interview Tips

- **Lifting state up kya hai?**
 - State ko parent component mein move karna taaki child components share kar sakein.
- **Kab lifting state up karte hain?**
 - Jab multiple components ko same state chahiye ya ek ke changes doosre ko affect karein.
- **Props drilling se kaise bach sakte hain?**
 - Lifting state up se kam hota hai, complex apps mein Context API ya Redux use hota hai.

7 Assignment: Practice Time

7.1 Task 1: Simple Counter App

- **CounterApp** parent component banao jo **count** state rakhe.
- **CounterDisplay** (count dikhao) aur **CounterControls** (increment/decrement buttons) child components banao.
- State parent mein rakho, props se pass karo.
- Tailwind se style karo.

7.2 Task 2: Todo List with Form

- **TodoApp** parent component banao jo **todos** array state rakhe.
- **TodoForm** (new todo add) aur **TodoList** (todos render) child components banao.
- Form se todo add karo, list mein dikhao.
- Tailwind se style karo.

7.3 Task 3: Filterable List

- `FilterApp` parent component banao jo `items` aur `filter` state rakhe.
- `FilterInput` (text input) aur `FilteredList` (filtered items) child components banao.
- Filter input ke hisaab se list update ho.
- Tailwind se style karo.