

2. Best Practice

- The **state** should be **immutable**. We shouldn't update the array/object directly.
- The best practice is to create a new array/object from the array/object in the previous state using the **spread** operator.

```
1 this.state.contactsList = initialContactsList
```

JAVASCRIPT

```
1 this.state.contactsList.push(...)
```

JAVASCRIPT

```
1 this.setState(prevState => ({
2   contactsList: [...prevState.contactsList, newContact ],
3 })))
```

JAVASCRIPT

2.1 Updating a Property of an Item inside List

We should not update the property of a list item directly. To update the property of a list item, we should create a new object and return it to the list.

Syntax:

```
1 {...object, newItem}
```

Exmaple:

File: src/App.js

```
1 import {Component} from 'react'
2 import {v4 as uuidv4} from 'uuid'
3
```

JSX

```
4 import ContactItem from './components/ContactItem'
5
6 import './App.css'
7
8 const initialContactsList = [
9   {
10     id: uuidv4(),
11     name: 'Ram',
12     mobileNo: 9999988888,
13     isFavorite: false,
14   },
15   {
16     id: uuidv4(),
17     name: 'Pavan',
18     mobileNo: 8888866666,
19     isFavorite: true,
20   },
21   {
22     id: uuidv4(),
23     name: 'Nikhil',
24     mobileNo: 9999955555,
25     isFavorite: false,
26   },
27 ]
28
29 class App extends Component {
30   state = {
31     contactsList: initialContactsList,
32     name: '',
33     mobileNo: '',
34   }
35
36   toggleIsFavorite = id => {
37     this.setState(prevState => ({
38       contactsList: prevState.contactsList.map(eachContact => {
39         if (id === eachContact.id) {
```

```

39     // toggle isFavorite
40     // eachContact.isFavorite = !eachContact.isFavorite
41     return {...eachContact, isFavorite: !eachContact.isFavorite}
42   }
43   return eachContact
44 },
45 )))
46 }
47
48 onAddContact = event => {
49   event.preventDefault()
50   const {name, mobileNo} = this.state
51   const newContact = {
52     id: uuidv4(),
53     name,
54     mobileNo,
55     isFavorite: false,
56   }
57
58   this.setState(prevState => ({
59     contactsList: [...prevState.contactsList, newContact],
60     name: '',
61     mobileNo: '',
62   }))
63 }
64
65 onChangeMobileNo = event => {
66   this.setState({mobileNo: event.target.value})
67 }
68
69 onChangeName = event => {
70   this.setState({name: event.target.value})
71 }
72
73 render() {
74   const {name, mobileNo, contactsList} = this.state
75   return (

```

```
75   return (
76     <div className="app-container">
77       <div className="responsive-container">
78         <h1 className="heading">Contacts</h1>
79         <form className="contact-form-container" onSubmit={this.onAddContact}>
80           <input
81             value={name}
82             onChange={this.onChangeName}
83             className="input"
84             placeholder="Name"
85           />
86           <input
87             className="input"
88             value={mobileNo}
89             onChange={this.onChangeMobileNo}
90             placeholder="Mobile Number"
91           />
92           <button type="submit" className="button">
93             Add Contact
94           </button>
95         </form>
96         <ul className="contacts-table">
97           <li className="table-header">
98             <p className="table-header-cell name-column">Name</p>
99             <hr className="separator" />
100            <p className="table-header-cell">Mobile Number</p>
101          </li>
102          {contactsList.map(eachContact => (
103            <ContactItem
104              key={eachContact.id}
105              contactDetails={eachContact}
106              toggleIsFavorite={this.toggleIsFavorite}
107            />
108          ))}
109        </ul>
110      </div>
```

```
111     </div>
112   )
113 }
114 }
115
116 export default App
```

Collapse ^