

## Bài thực hành số 3: Làm việc với nhiều component và các kỹ thuật cải tiến mã

Bây giờ chúng ta sẽ thêm một component mới gọi là Counters.jsx (hay Counters.js) đến thư mục Component. Nội dung như sau

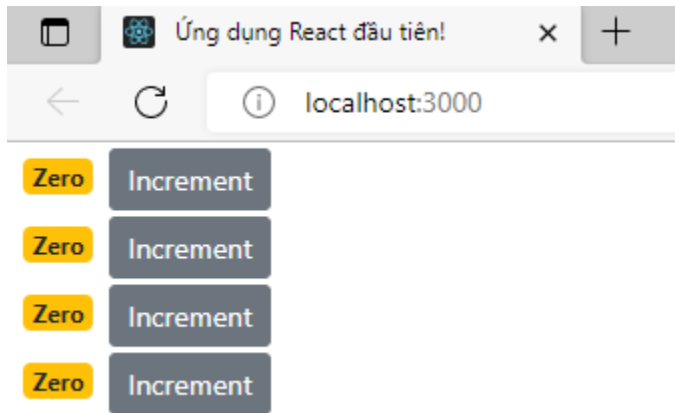
```
import React, { Component } from 'react';
import Counter from './counter';
class Counters extends Component {
  state = {};

  render() {
    return (
      <div>
        <Counter />
        <Counter />
        <Counter />
        <Counter />
      </div>
    );
  }
}
export default Counters;
```

Gọi component Counters từ index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import 'bootstrap/dist/css/bootstrap.css';
import Counters from './components/counters';
ReactDOM.render(<Counters />, document.getElementById("root"));
```

Thực thi ứng dụng



Thay thế count bằng value từ Counter

```
state = {
  value: this.props.value
};
handleIncrement = () => {
  this.setState({ value: this.state.value + 1 });
};

getBadgeClasses() {
  let classes = "badge m-2 badge-";
  classes += (this.state.value === 0) ? "warning" : "primary";
  return classes;
}

formatCount() {
  const { value } = this.state;
  return value === 0 ? "Zero" : value;
}
```

Nội dung Counters thay đổi như sau:

```
state = {
  counters: [
    { id: 1, value: 0 },
    { id: 2, value: 0 },
    { id: 3, value: 0 },
    { id: 4, value: 0 }
  ]
};

render() {
  return (
```

```

        <div>
          {this.state.counters.map(counter => (
            <Counter key={counter.id} value={counter.value}
id={counter.id} />
          ))}
        </div>
      );
    }
  }
}

```

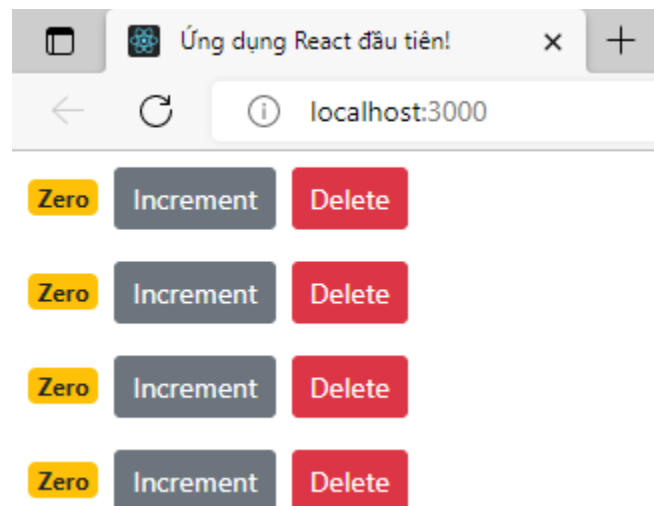
Thực thi kết quả không thay đổi nhưng sẽ có ích sắp tới. Thêm nút Delete đến Counter

```

<div>
  <span
className={this.getBadgeClasses()}>{this.formatCount()}</span>
  <button onClick={this.handleIncrement} className='btn btn-
secondary btn-sm'> Increment</button>
  <button onClick={this.handleDelete} className='btn btn-danger
btn-sm m-2'>Delete</button>
</div>

```

Kết quả



Trong component Counters thêm phương thức handleDelete và thuộc tính sự kiện onDelete trong render() như sau:

```

handleDelete = (counterId) => {
  const counters = this.state.counters.filter(c => c.id !== counterId);
  this.setState({ counters });
}

```

```

};
render() {
  return (
    <div>
      {this.state.counters.map(counter => (
        <Counter
          key={counter.id}
          onDelete={this.handleDelete}
          value={counter.value}
          id={counter.id}
        />
      ))}
    </div>
  );
}

```

Trong component Counter thay đổi nội dung tại thuộc tính onClick của button Delete:

```

<button onClick={() => this.props.onDelete(this.props.id)} className='btn btn-danger btn-sm m-2'>Delete</button>

```

Thực thi lại ứng dụng và nhấn nút Delete.

Chúng ta có thể cập nhật trạng thái bằng cách thay đổi các thuộc tính trong phần tử <Counter> của component Counters

```

render() {
  return (
    <div>
      {this.state.counters.map(counter => (
        <Counter
          key={counter.id}
          onDelete={this.handleDelete}
          counter={counter}
        />
      ))}
    </div>
  );
}

```

Thay đổi nội dung thuộc tính state trong component Counter

```

state = {

```

```
value: this.props.counter.value
};
```

và thuộc tính onClick của button Delete

```
<button onClick={() => this.props.onDelete(this.props.counter.id)} className='btn btn-danger btn-sm m-2'>Delete</button>
```

Thực thi lại ứng dụng và nhấn nút Delete.

Trong Counters thêm button Reset đến render()

```
render() {
  return (
    <div>
      <button
        onClick={this.handleReset}
        className='btn btn-primary btn-sm m-2'>Reset</button>
      {this.state.counters.map(counter => (
        <Counter
          key={counter.id}
          onDelete={this.handleDelete}
          counter={counter}
        />
      ))}
    </div>
  );
}
```

Thêm phương thức handleReset phía trên handleDelete

```
handleReset = () => {
  const counters = this.state.counters.map(c => {
    c.value = 0;
    return c;
  });
  this.setState({ counters });
};
```

Kế tiếp là xóa bỏ thuộc tính state (Local State) trong Counter đồng thời thay đổi các nội dung liên quan đến thuộc tính này như sau:

- Xóa bỏ phương thức handleIncrement

- Thay đổi nội dung thuộc tính onClick của button Increment

```
<button onClick={() => this.props.onIncrement(this.props.counter)} className='btn btn-secondary btn-sm'> Increment</button>
```

- Thay đổi các nội dung liên quan khác từ các phương thức getBadgeClasses và formatCount

```
getBadgeClasses() {
  let classes = "badge m-2 badge-";
  classes += (this.props.counter.value === 0) ? "warning" : "primary";
  return classes;
}

formatCount() {
  const { value } = this.props.counter;
  return value === 0 ? "Zero" : value;
}
```

Trong Counters thêm hàm handleIncrement

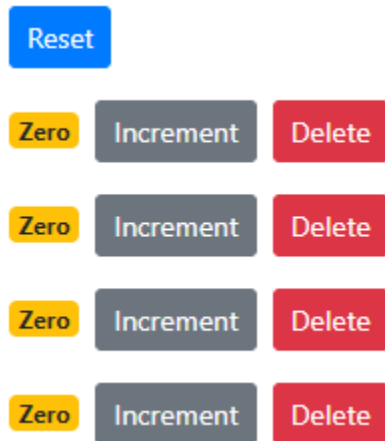
```
handleIncrement = counter => {
  const counters = [...this.state.counters];
  const index = counters.indexOf(counter);
  counters[index] = { ...counter };
  counters[index].value++;
  this.setState({ counters });
};
```

Thêm thuộc tính onIncrement đến <Counter../> trong render():

```
render() {
  return (
    <div>
      <button
        onClick={this.handleReset}
        className='btn btn-primary btn-sm m-2'>Reset</button>
      {this.state.counters.map(counter => (
        <Counter
          key={counter.id}
          onDelete={this.handleDelete}
          onIncrement={this.handleIncrement}
          counter={counter}
        />
      )}
    </div>
  );
}
```

```
    ))}
  </div>
);
}
```

Thực thi ứng dụng



Nhấn nút Increment bất kỳ sau đó nhấn Reset để xem kết quả.

Cho đến thời điểm này chúng ta có 3 component là App, Counter và Counters. Chúng ta sẽ thêm một component tiếp là Navbar bằng cách thêm tập tin Navbar.js đến thư mục Components

```
import React, { Component } from 'react';
class Navbar extends Component {
  render() {
    return (
      <nav className='navbar navbar-light bg-light'>
        <a className='navbar-brand' href='#'>
          Navbar
        </a>
      </nav>
    );
  }
}
export default Navbar;
```

trong App.js thay đổi nội dung

```
import React, { Component } from 'react';
import Navbar from './components/navbar';
```

```

import Counters from './components/counters';
import './App.css'
class App extends Component {
  render() {
    return (
      <React.Fragment>
        <Navbar />
        <main className='container'>
          <Counters />
        </main>
      </React.Fragment>
    );
  }
}
export default App;

```

Trong tập tin Index.js thay đổi lại nội dung

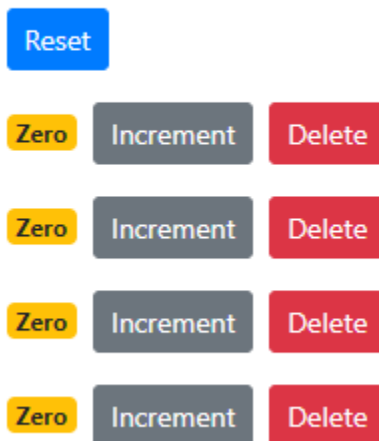
```

import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';
import 'bootstrap/dist/css/bootstrap.css';
import Counters from './components/counters';
ReactDOM.render(<App />, document.getElementById("root"));

```

Thực thi ứng dụng

Navbar





Lúc này chúng ta có các component tổ chức như sau:



Bây giờ chúng ta sẽ di chuyển thuộc tính state chứa counters và các phương thức liên quan như `handleIncrement`, `handleDelete`, `handleReset` trong component `Counters` lên component `App` để tiện quản lý, nội dung `App.js`

```
import React, { Component } from 'react';
import NavBar from './components/navbar';
import Counters from './components/counters';
import './App.css'
class App extends Component {
  state = {
    counters: [
      { id: 1, value: 0 },
      { id: 2, value: 0 },
      { id: 3, value: 0 },
      { id: 4, value: 0 }
    ]
  };

  handleIncrement = counter => {
    const counters = [...this.state.counters];
    const index = counters.indexOf(counter);
    counters[index] = { ...counter };
    counters[index].value++;
    this.setState({ counters });
  };
}
```

```

};

handleReset = () => {
  const counters = this.state.counters.map(c => {
    c.value = 0;
    return c;
  });
  this.setState({ counters });
};

handleDelete = (counterId) => {
  const counters = this.state.counters.filter(c => c.id !== counterId);
  this.setState({ counters });
  //console.log("Toi bi xoa", counterId);
};

render() {
  return (
    <React.Fragment>
      <Navbar />
      <main className='container'>
        <Counters
          counters={this.state.counters}
          onDelete={this.handleDelete}
          onReset={this.handleReset}
          onIncrement={this.handleIncrement} />
      </main>
    </React.Fragment>
  );
}
}
export default App;

```

Nội dung counters.js

```

import React, { Component } from 'react';
import Counter from './counter';
class Counters extends Component {
  render() {
    return (
      <div>
        <button
          onClick={this.props.onReset}
          className='btn btn-primary btn-sm m-2'>Reset</button>
        {this.props.counters.map(counter => (
          <Counter

```

```

        key={counter.id}
        onDelete={this.props.onDelete}
        onIncrement={this.props.onIncrement}
        counter={counter}
      />
    )))
  </div>
);
}
}
export default Counters;

```

Kế tiếp chúng ta thêm tính năng đếm trên thanh Navbar bằng cách bổ sung thuộc tính totalCounters đến <Navbar.../> trong App.js

```

<React.Fragment>
  <Navbar totalCounters={this.state.counters.filter(c => c.value >
0).length} />
  <main className='container'>
    <Counters
      counters={this.state.counters}
      onDelete={this.handleDelete}
      onReset={this.handleReset}
      onIncrement={this.handleIncrement} />
  </main>
</React.Fragment>

```

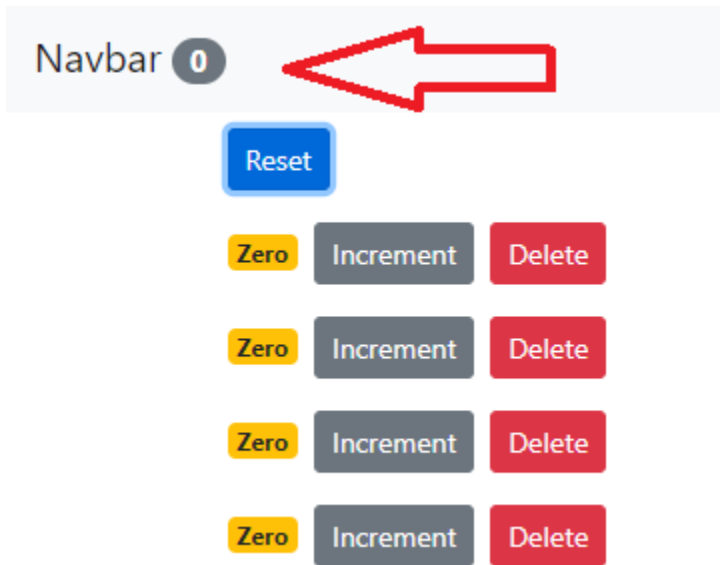
Trong navbar.jsx thay đổi nội dung trong <nav>

```

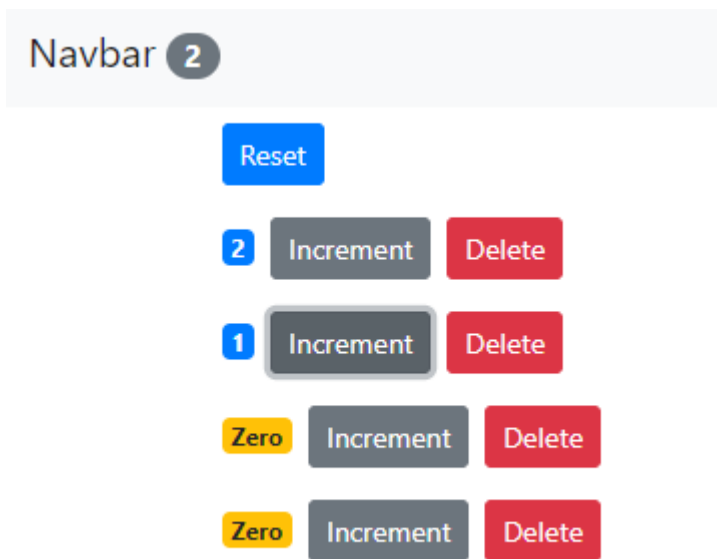
<nav className='navbar navbar-light bg-light'>
  <a className='navbar-brand' href='#'>
    Navbar {" "}
    <span className='badge badge-pill badge-secondary'>
      {this.props.totalCounters}
    </span>
  </a>
</nav>

```

Thực thi ứng dụng



Nhấn các nút Increment



Nhấn nút Delete đầu tiên

# Navbar 1

Reset

1 Increment Delete

Zero Increment Delete

Zero Increment Delete