Working with Fragments

- Any component we create in React, can only return one element.
  In our previous examples, if we observe all the components are returning one **div** container.

What will happen if we return multiple elements from one Component?

```
import React from 'react';
import ReactDOM from 'react-dom';

class ProductInformatonComponent extends React.Component {
    constructor(props) {
        super(props);
    }

    render() {
        return (
            <div>
                <p>Electronics</p>
            </div>
            <div>
                <p>Cloths</p>
            </div>
        );
    }
}

const element = <ProductInformatonComponent></ProductInformatonComponent>
ReactDOM.render(element, document.getElementById("root"));
```

React will throw us an error saying: Parsing error:
Adjacent JSX elements must be wrapped in an enclosing tag. Did you want a JSX fragment <>...</>?

- But it is very common scenario we face in our day to day programming life that we want to return multiple elements from one component.

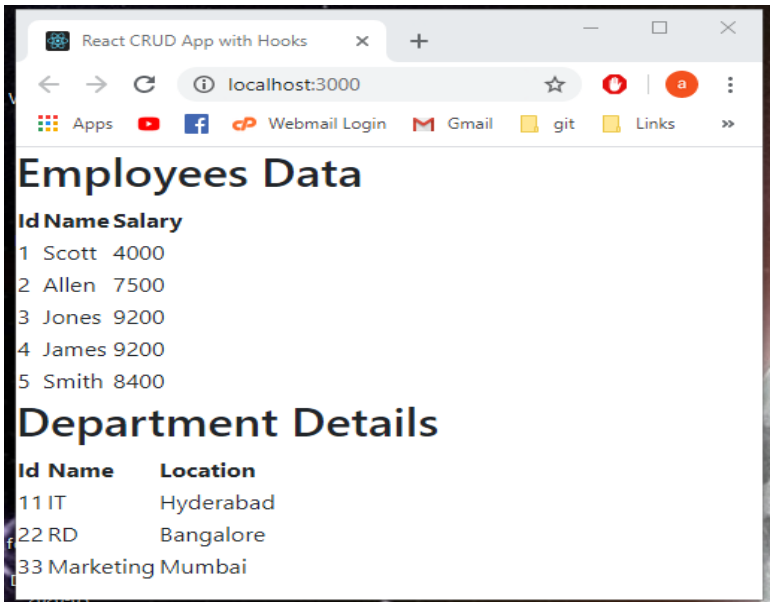
- React provides **Fragments** that let you group a list of children without adding extra nodes to the DOM.
  <React.Fragment></React.Fragment>
  (or)
  Short Cut: <> </>

Eg:

```
import React, { Fragment } from 'react';
import ReactDOM from 'react-dom';

class ProductInformatonComponent extends React.Component {
    constructor(props) {
        super(props);
    }

    render() {
        return (
            <React.Fragment>
                <div>
                    <p>Electronics</p>
                </div>
                <div>
                    <p>Cloths</p>
                </div>
            </React.Fragment>
        );
    }
}

const element = <ProductInformatonComponent></ProductInformatonComponent>
ReactDOM.render(element, document.getElementById("root"));
```

Higher Order Component (HOC)

- As part programming requirement there might be instances where we have to use the same repetitive and redundant state-full logic (such as setting up a subscription and remembering the current value) inside multiple components.
  For Example.



We have two components, one is Employee Component and the other one is Department Component.

Where Employee Component connects to a Rest API and fetch the employee data when the Component is mounted and display the data.

Similarly Department Component connects to a Rest API and fetch the department data when the Component is mounted and display the data.

```
import React from 'react'
import ReactDOM from 'react-dom'

class Employee extends React.Component {
    constructor(props) {
    }

    componentDidMount() {
        //Calling Rest api
    }

    render() {
        return (
            //rendering code
        );
    }
}
class Department extends React.Component {
    constructor(props) {
    }
    componentDidMount() {
      //Calling Rest api
   }

  render() {
      return (
          //rendering code
      );
  }
}

class App extends React.Component {
    render() {
        return (
            <React.Fragment>
                <Employee></Employee>
                <Department></Department>
            </React.Fragment>
        );
    }
}
const element = <App></App>
ReactDOM.render(element, document.getElementById("root"));
```

- In the above program, the REST API code is redundant in both the Components.
- We can avoid this by using **Higher Order Components**.

Higher Order Components:

- A Higher-Order component (HOC) is an advanced technique in React for reusing component logic.
- Higher Order Component is a function that takes a component as input and returns a new component.
- Higher Order Components promote Code Reusability.
- Using HOC's are common in any Enterprise application we develop using React.

Example:

D:\ReactJS\hello-app>json-server --watch db.json --port 4000

D:\ReactJS\hello-app>npm start

hello-app\db.json

```
{
  "employees": [
    {
      "id": 1,
      "name": "Scott",
      "salary": 4000
    },
    {
      "id": 2,
      "name": "Allen",
      "salary": 7500
    }
  ],
  "departments": [
    {
      "id": 11,
      "name": "IT",
      "location": "Hyderabad"
    },
    {
      "id": 22,
      "name": "RD",
      "location": "Bangalore"
    }
  ]
}
```

index.js

```
import React from "react";
import ReactDOM from "react-dom";

function myHOC(InputComponent, inputData) {
  return class extends React.Component {
    constructor(props) {
      super(props);
      this.state = {
        data: [],
        columns: inputData.columns,
        header: inputData.header,
      };
    }

    componentDidMount() {
      fetch(inputData.url)
        .then((res) => res.json())
        .then((result) => {
          this.setState({
            data: result,
          });
        });
    }

    render() {
      return <Data data={this.state}></Data>;
    }
  };
}

class Data extends React.Component {
  constructor(props) {
    super(props);
  }

  render() {
    return (
      <div>
        <h2>{this.props.data.header}</h2>
        <table>
          <thead>
            <tr>
              {this.props.data.columns.map((c) => (
                <th>{c}</th>
              ))}
            </tr>
          </thead>
          <tbody>
            {this.props.data.data.map((emp) => (
              <tr key={emp.id}>
```

```
                {this.props.data.columns.map((c) => (
                  <td>{emp[c]}</td>
                ))}
              </tr>
            ))}
          </tbody>
        </table>
      </div>
    );
  }
}

class Display extends React.Component {
  constructor(props) {
    super(props);
  }

  render() {
    return <div></div>;
  }
}
const Employee = myHOC(Display, {
  url: "http://localhost:4000/employees",
  columns: ["id", "name", "salary"],
  header: "Employee Details",
});

const Department = myHOC(Display, {
  url: "http://localhost:4000/departments",
  columns: ["id", "name", "location"],
  header: "Dept Details",
});

class MainApp extends React.Component {
  constructor(props) {
    super(props);
  }

  render() {
    return (
      <React.Fragment>
        <Employee></Employee>
        <Department></Department>
      </React.Fragment>
    );
  }
}
const element = <MainApp/>s
ReactDOM.render(element, document.getElementById("root"));
```

Integration of Bootstrap with React

We can integrate bootstrap with react in 2 ways;

1. Using Classic Bootstrap
2. Using react-bootstrap

Woking with react-bootstrap:

Installation of react-bootstrap
npm install react-bootstrap bootstrap

hello-app>npm install react-bootstrap bootstrap

Reference:
https://react-bootstrap.github.io/getting-started/introduction

Eg:
index.js

```
import React from "react";
import ReactDOM from "react-dom";
import Button from "react-bootstrap/Button";
import "bootstrap/dist/css/bootstrap.min.css";

class Sample extends React.Component {
  render() {
    return (
      <div>
        <Button variant="success">Sample</Button>
      </div>
    );
  }
}

const element = <Sample />;

ReactDOM.render(element, document.getElementById("root"));
```

Working with classic bootstrap:

Example:

Add the bootstrap cdn links in index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
    <title>React App</title>
    <script
      src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
      integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
      crossorigin="anonymous"
    ></script>
    <script
      src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js"
      integrity="sha384-ho+j7jyWK8fNQe+A12Hb8AhRq26LrZ/JpcUGGOn+Y7RsweNrtN/tE3MoK7ZeZDyx"
      crossorigin="anonymous"
    ></script>
    <link
      rel="stylesheet"
      href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css"
      integrity="sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2"
      crossorigin="anonymous"
    />
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>

  </body>
</html>
```

index.js

```
import React from "react";
import ReactDOM from "react-dom";

class Sample extends React.Component {
  render() {
    return (
      <div>
        <button className="btn btn-primary">sample</button>
      </div>
    );
  }
}

const element = <Sample />;

ReactDOM.render(element, document.getElementById("root"));
```