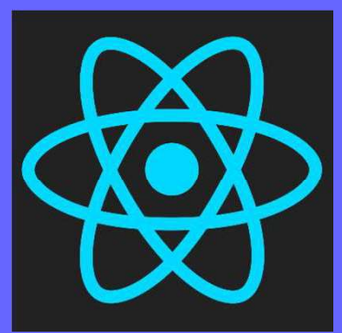
Component Types

Objectives

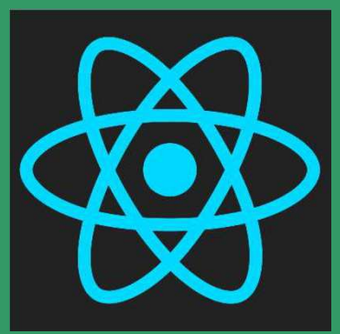
After completing this lesson, you should be able to do the following:

- Creating Components
- Basic Component
- Nesting Components
- Props



Introduction to Stateless & Statefull Components

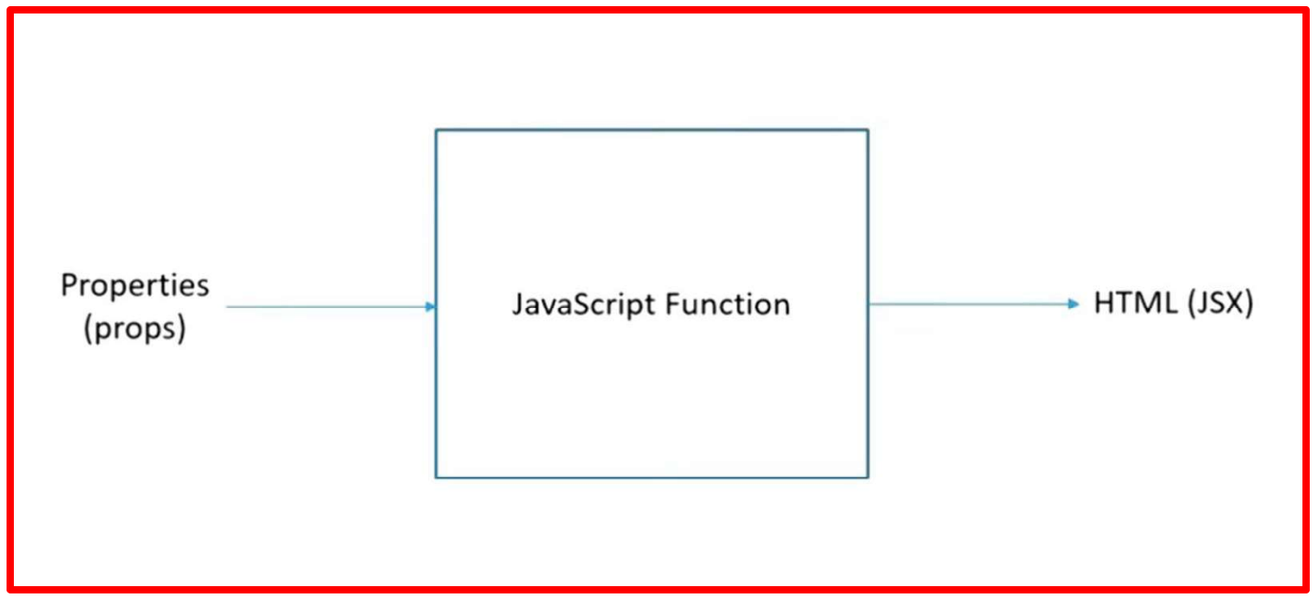
- A **stateless** component as it does not contain state (in the React sense of the word).
- In such a case, some people find it preferable to use Stateless Functional Components, which are based on ES6
- Arrow functions.

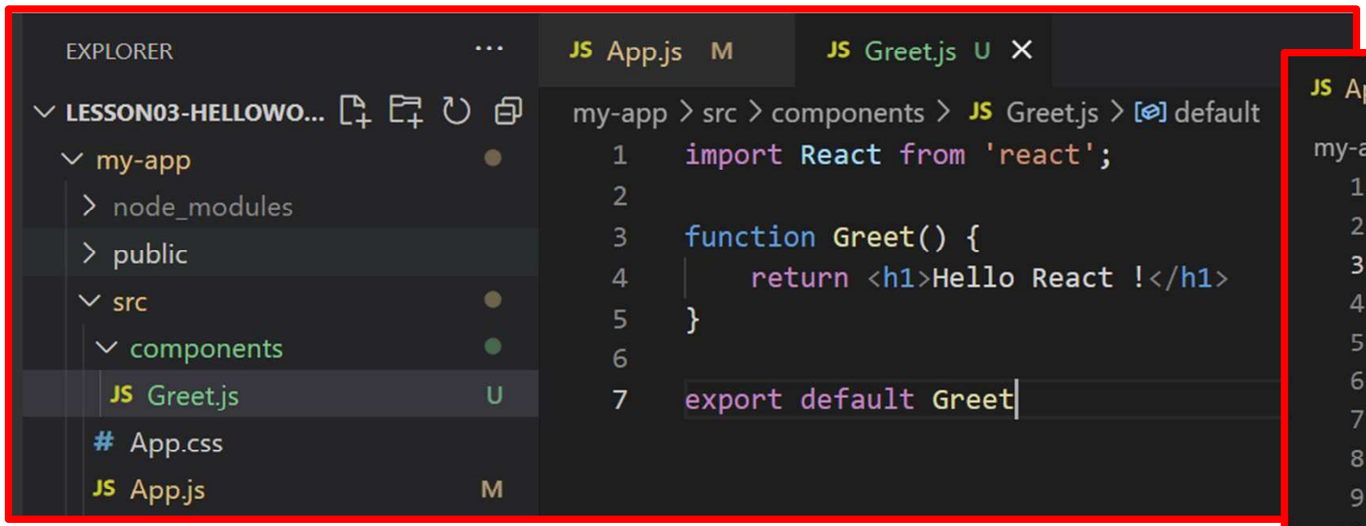


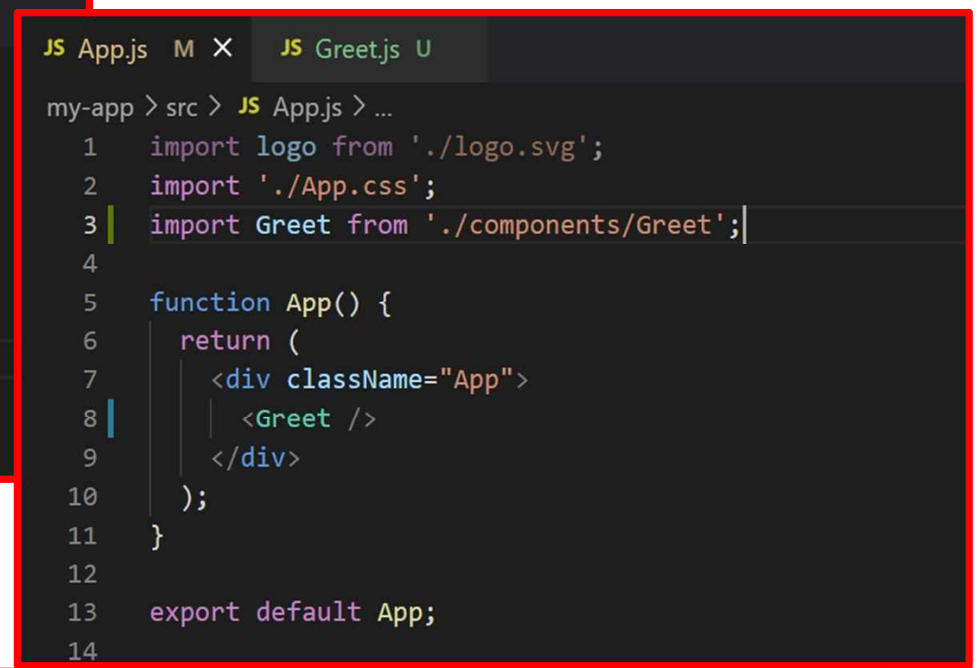
The Functional Components

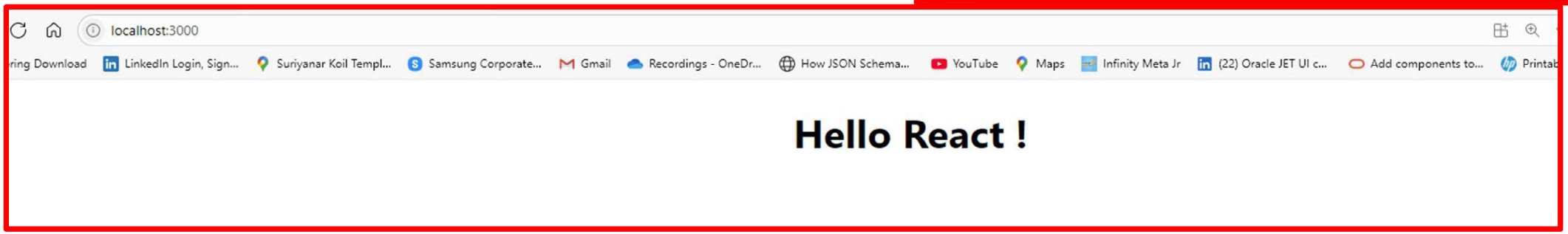
Functional Components

- Functional Components are Just JS Functions
- They can Optionally receive object of Properties Which is referred to as **props**
- **Returns HTML as the UI**
- The Functional component is a JavaScript function that accepts an input of properties and returns HTML that describes the UI

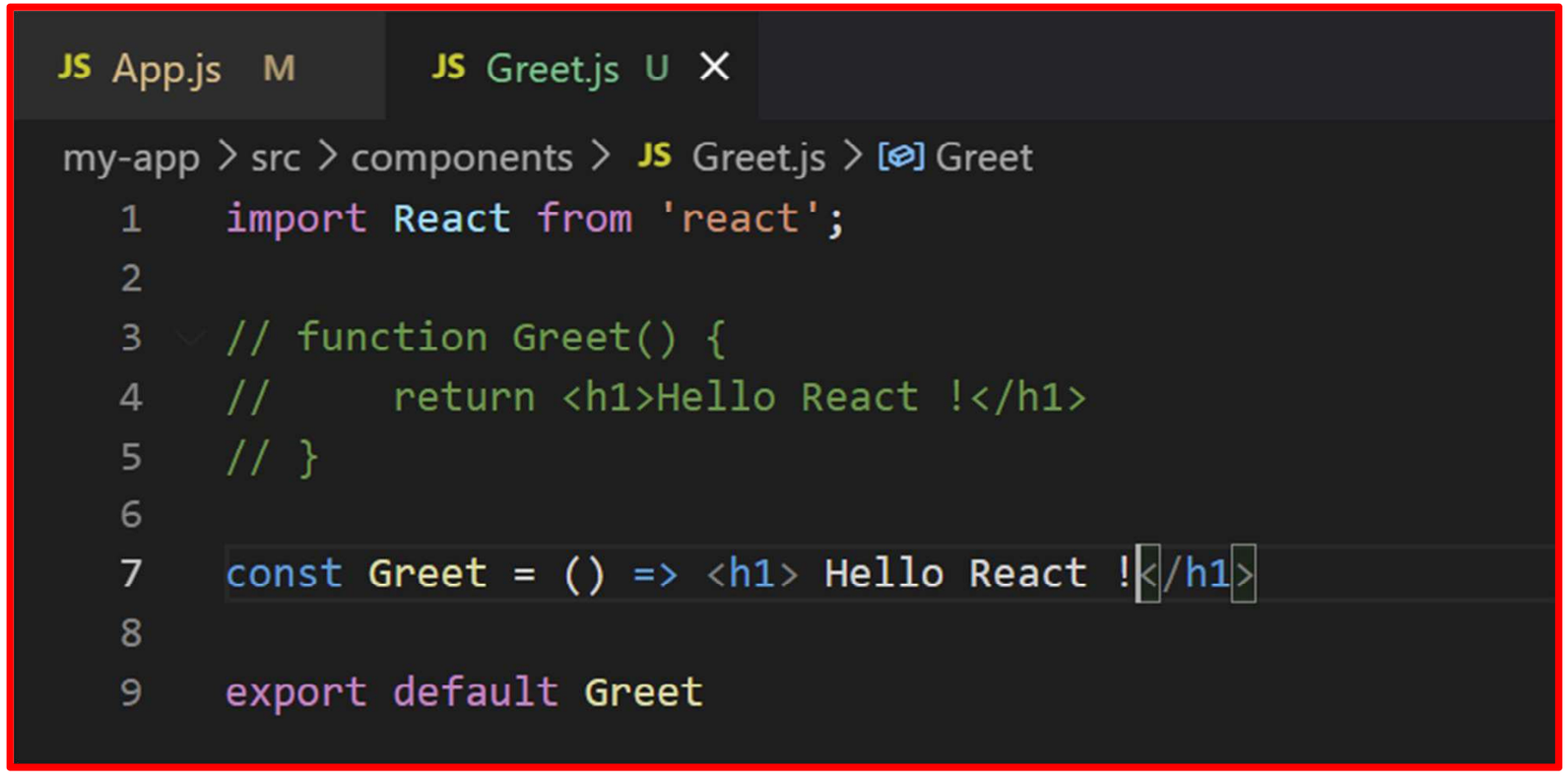



EXPLORER
LESSON03-HELLOWO...
my-app
node_modules
public
src
components
Greet.js U
App.css
App.js M
App.js M
Greet.js U
my-app > src > components > Greet.js > default
1 import React from 'react';
2
3 function Greet() {
4 return <h1>Hello React !</h1>
5 }
6
7 export default Greet


App.js M
Greet.js U
my-app > src > App.js > ...
1 import logo from './logo.svg';
2 import './App.css';
3 import Greet from './components/Greet';
4
5 function App() {
6 return (
7
8 <Greet />
9
10);
11 }
12
13 export default App;
14

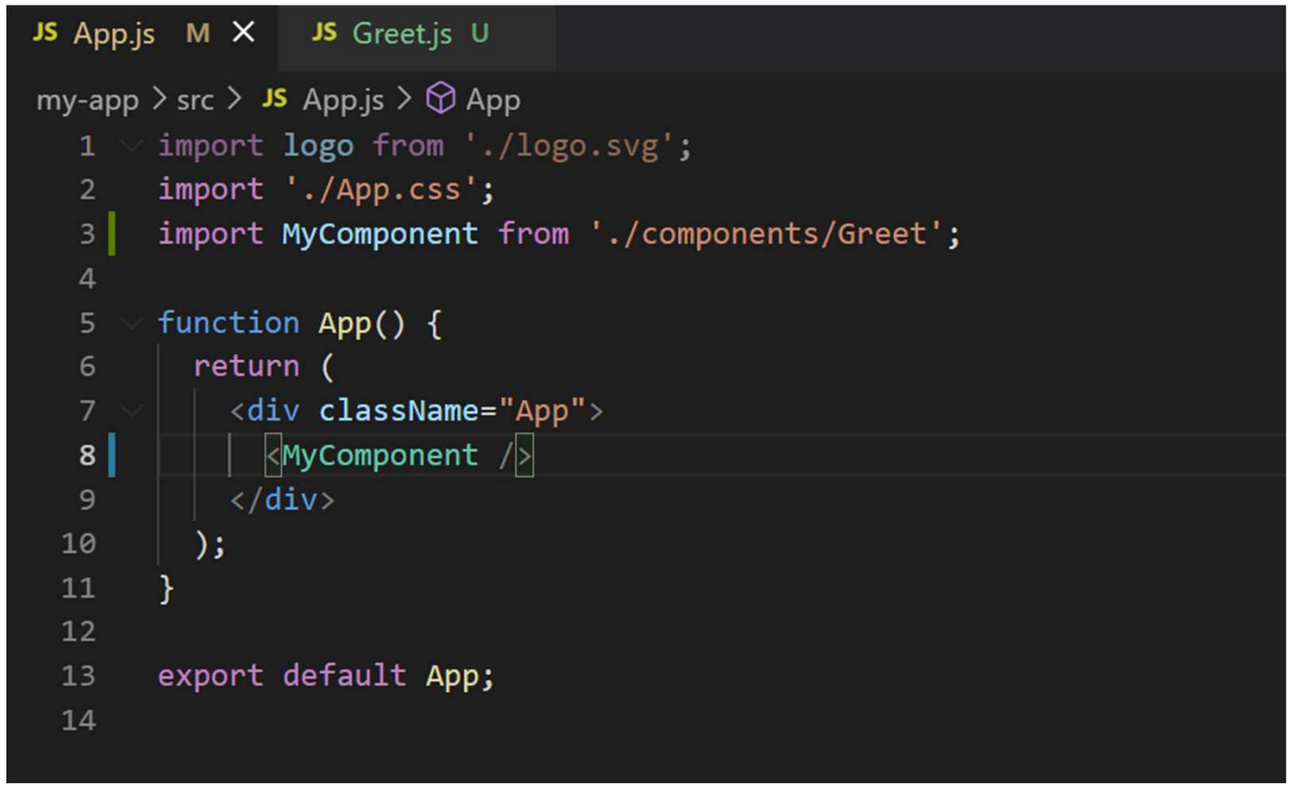

localhost:3000
Hello React !

But Convention is to Use Arrow Functions



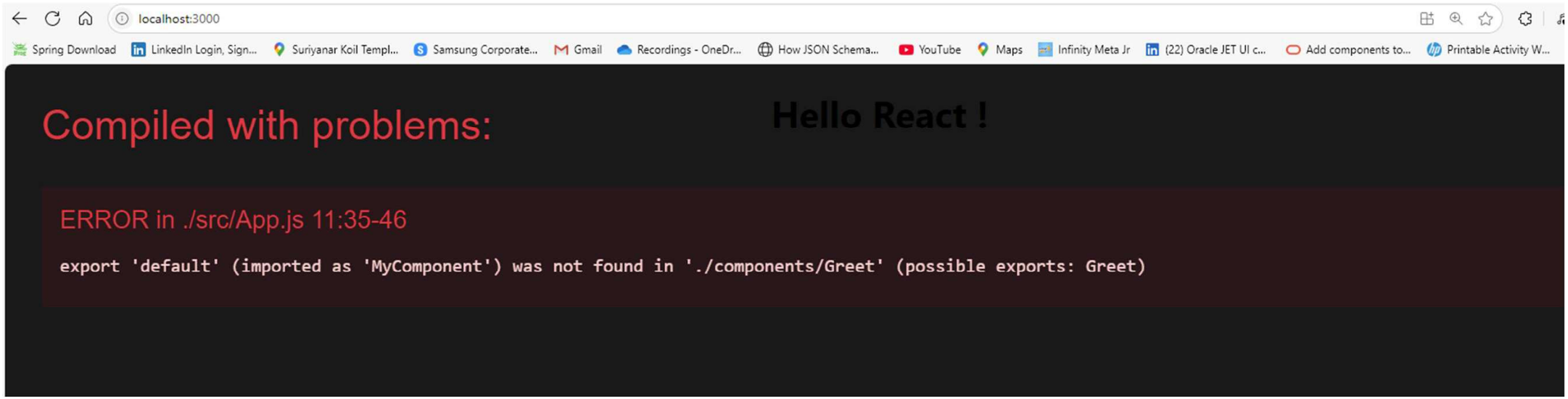
Export default Greet

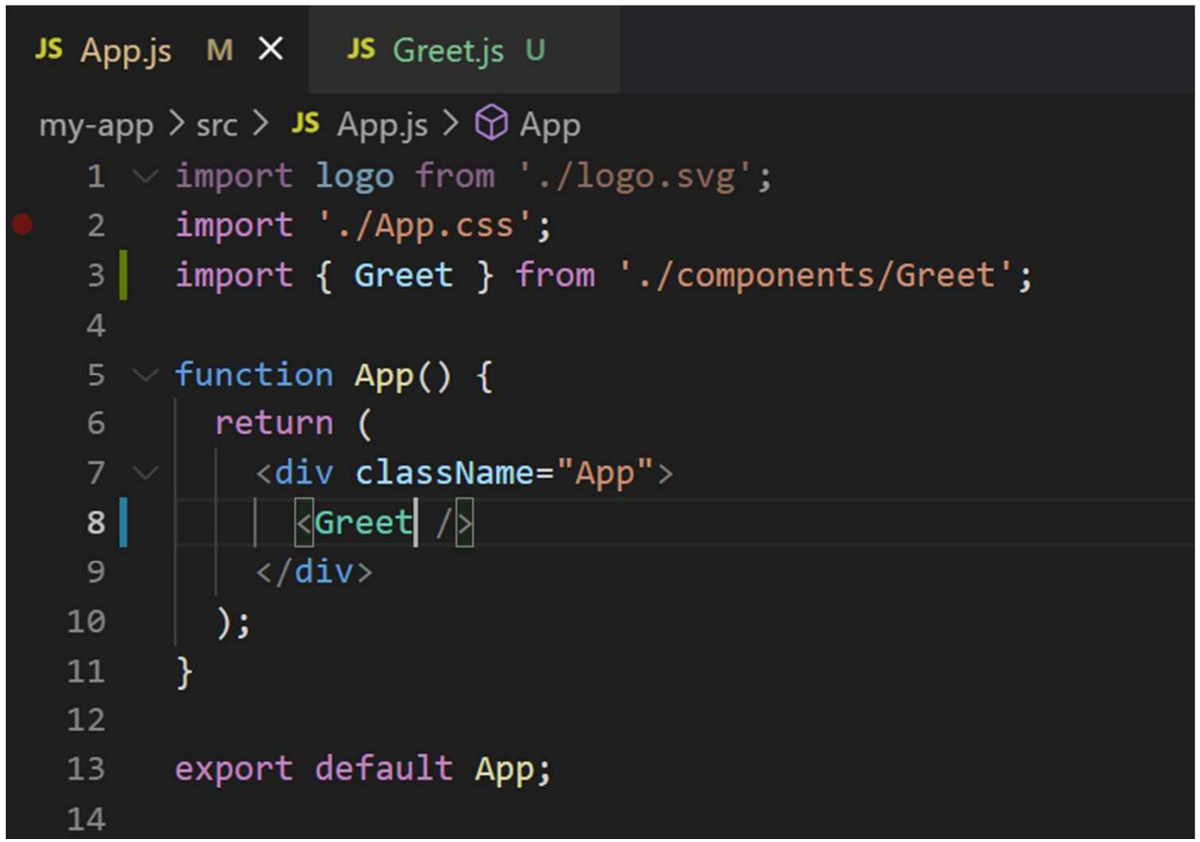
- This allows us to import the component with any name I can change greet to my component and change the tag to my component and you can see that it still works as expected

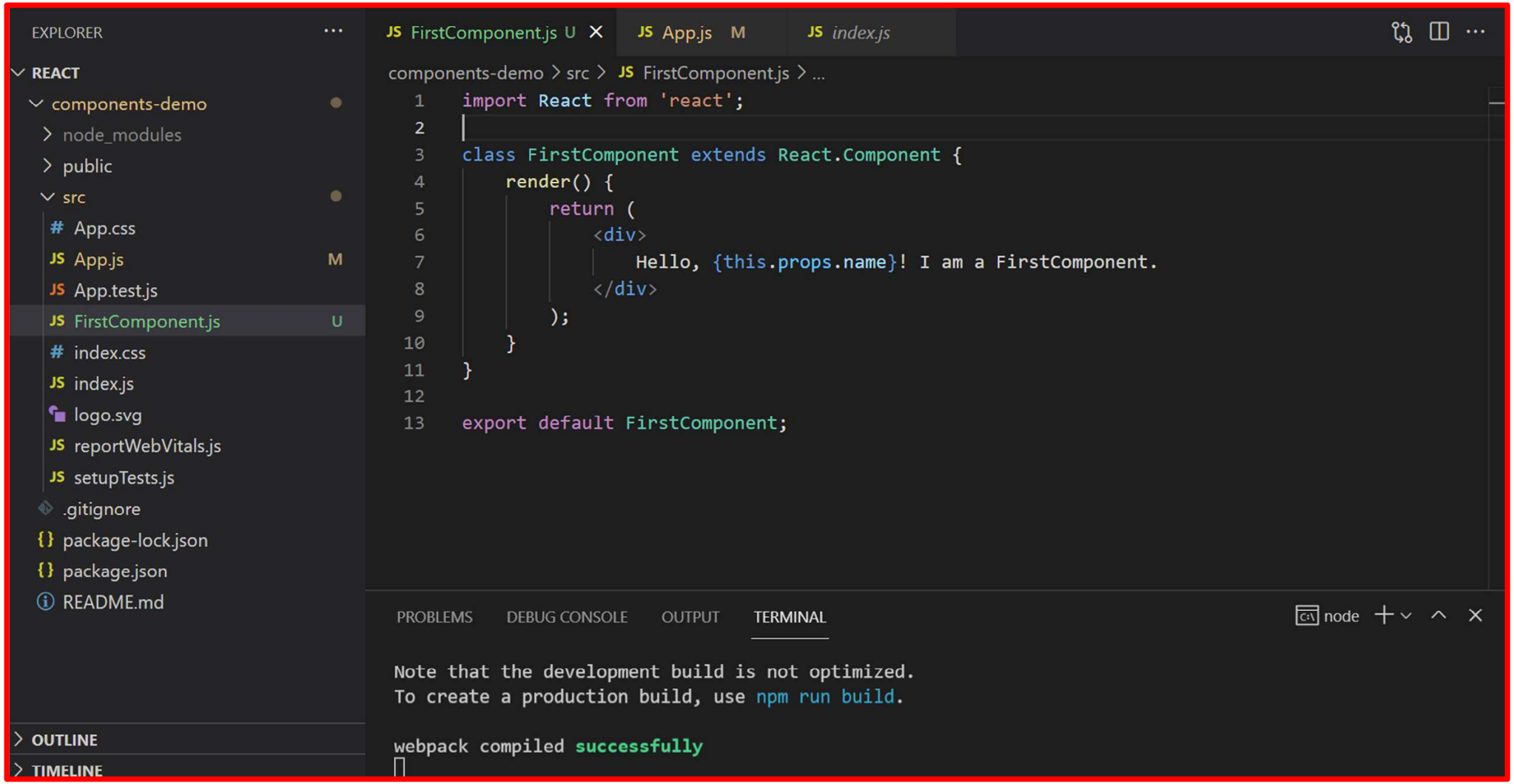


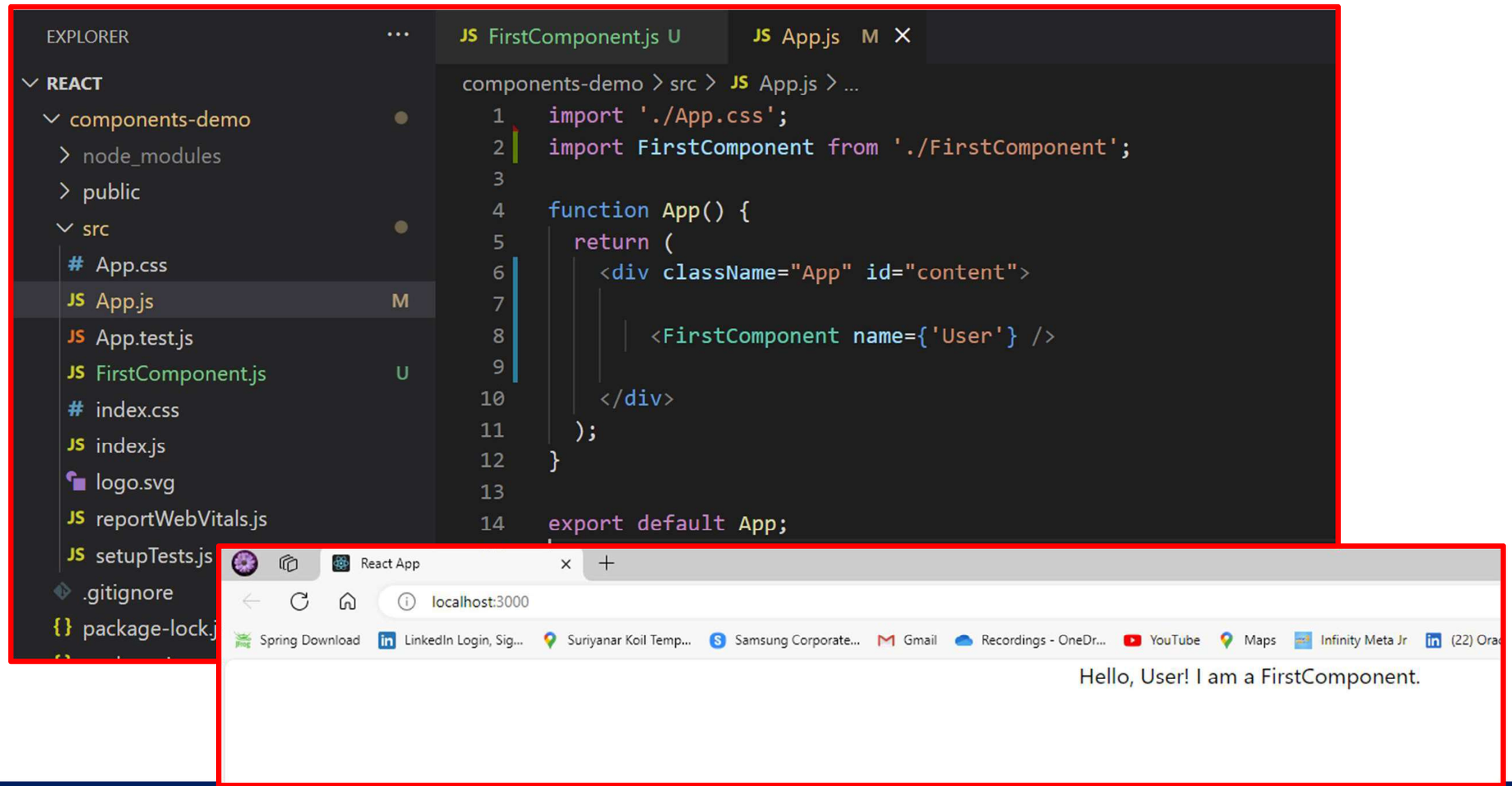
Named exports

- If we use Named export, then we have to import the component with the exact same name.


localhost:3000
Spring Download
LinkedIn Login, Sign...
Suriyanar Koil Templ...
Samsung Corporate...
Gmail
Recordings - OneDr...
How JSON Schema...
YouTube
Maps
Infinity Meta Jr
(22) Oracle JET UI c...
Add components to...
Printable Activity W...
Compiled with problems:
Hello React !
ERROR in ./src/App.js 11:35-46
export 'default' (imported as 'MyComponent') was not found in './components/Greet' (possible exports: Greet)


App.js M
Greet.js U
my-app > src > App.js > App
1 import logo from './logo.svg';
2 import './App.css';
3 import { Greet } from './components/Greet';
4
5 function App() {
6 return (
7
8 <Greet />
9
10);
11 }
12
13 export default App;
14

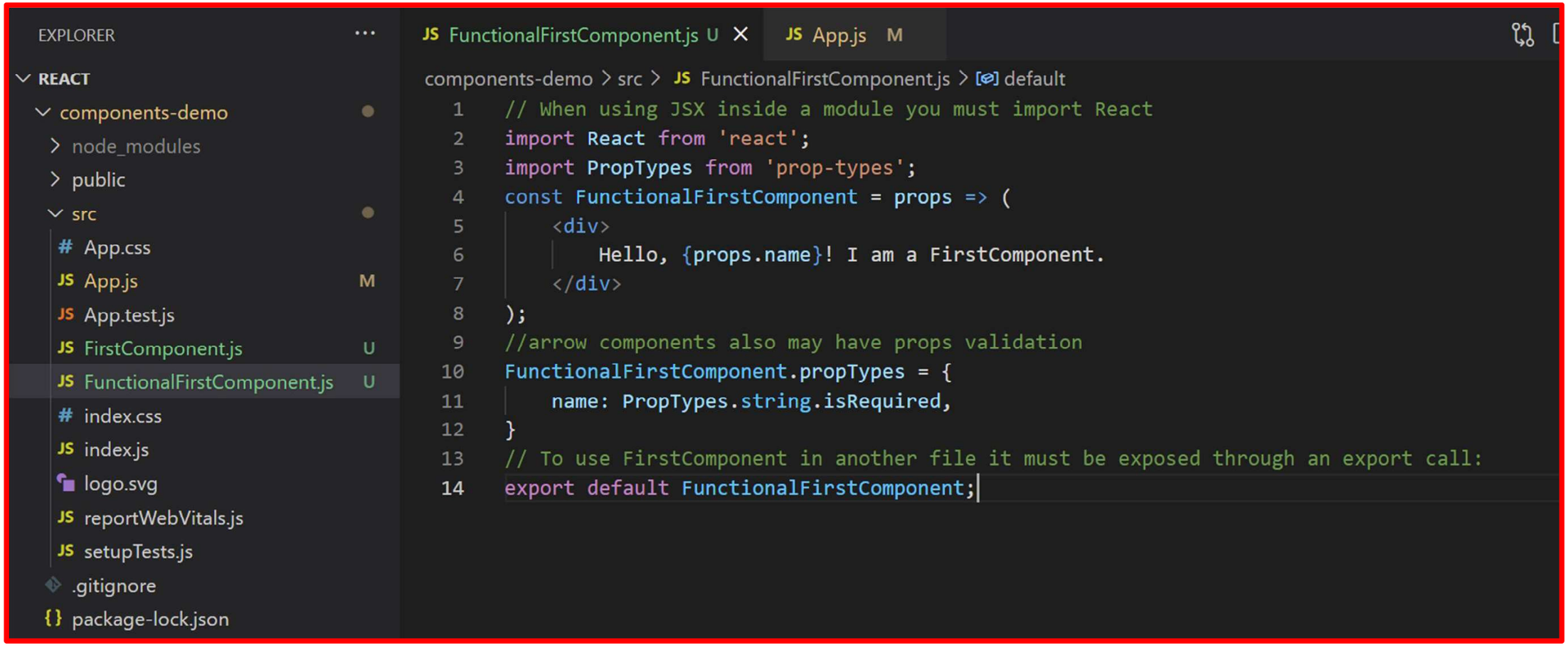

EXPLORER
REACT
components-demo
node_modules
public
src
App.css
App.js M
App.test.js
FirstComponent.js U
index.css
index.js
logo.svg
reportWebVitals.js
setupTests.js
.gitignore
package-lock.json
package.json
README.md
OUTLINE
TIMELINE
FirstComponent.js U
App.js M
index.js
components-demo > src > FirstComponent.js > ...
1 import React from 'react';
2
3 class FirstComponent extends React.Component {
4 render() {
5 return (
6
7 Hello, {this.props.name}! I am a FirstComponent.
8
9);
10 }
11 }
12
13 export default FirstComponent;
PROBLEMS
DEBUG CONSOLE
OUTPUT
TERMINAL
node
Note that the development build is not optimized.
To create a production build, use npm run build.
webpack compiled successfully

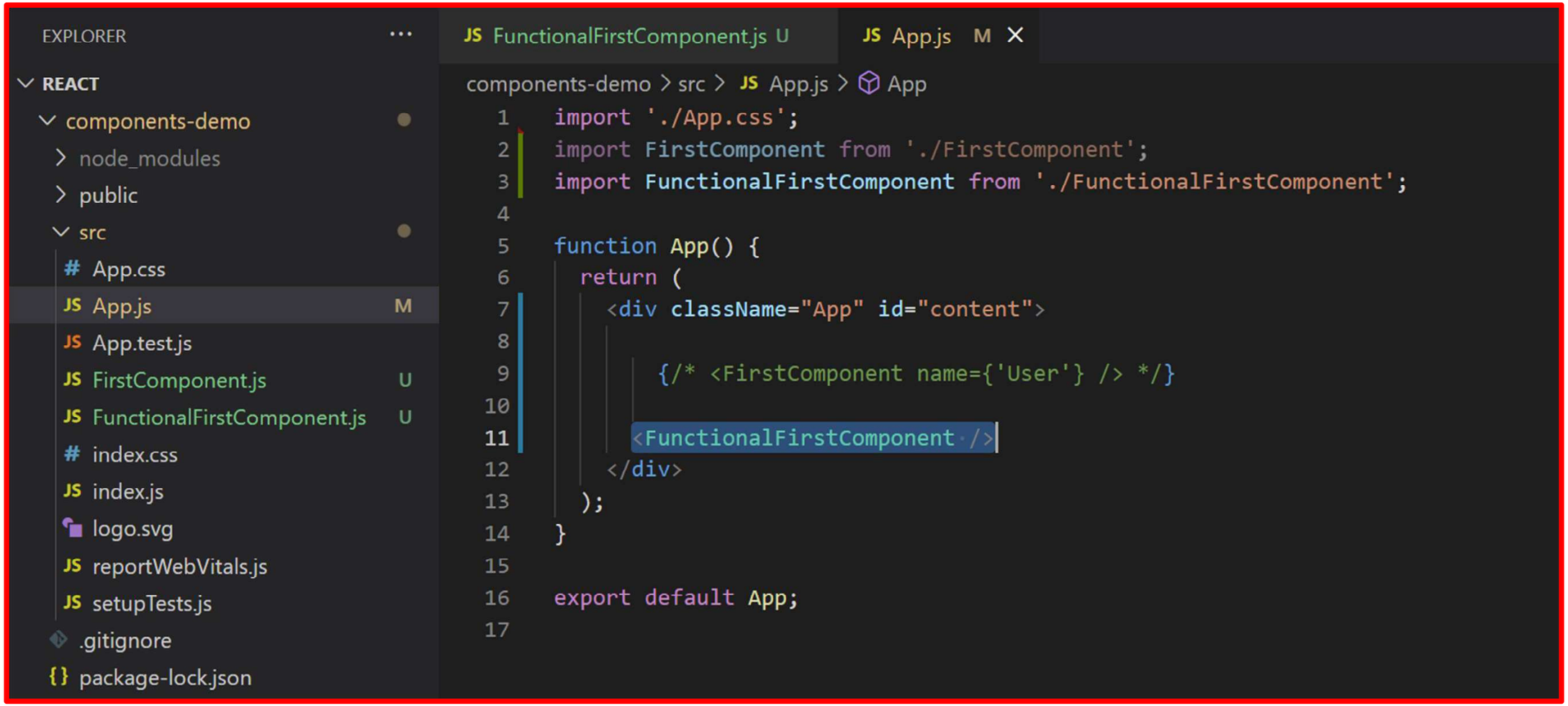

EXPLORER
REACT
components-demo
node_modules
public
src
App.css
App.js M
App.test.js
FirstComponent.js U
index.css
index.js
logo.svg
reportWebVitals.js
setupTests.js
.gitignore
package-lock.j
FirstComponent.js U
App.js M
components-demo > src > App.js > ...
1 import './App.css';
2 import FirstComponent from './FirstComponent';
3
4 function App() {
5 return (
6
7
8 <FirstComponent name={'User'} />
9
10
11);
12 }
13
14 export default App;
React App
localhost:3000
Spring Download
LinkedIn Login, Sig...
Suriyanar Koil Temp...
Samsung Corporate...
Gmail
Recordings - OneDr...
YouTube
Maps
Infinity Meta Jr
Hello, User! I am a FirstComponent.

Stateless Functional Components

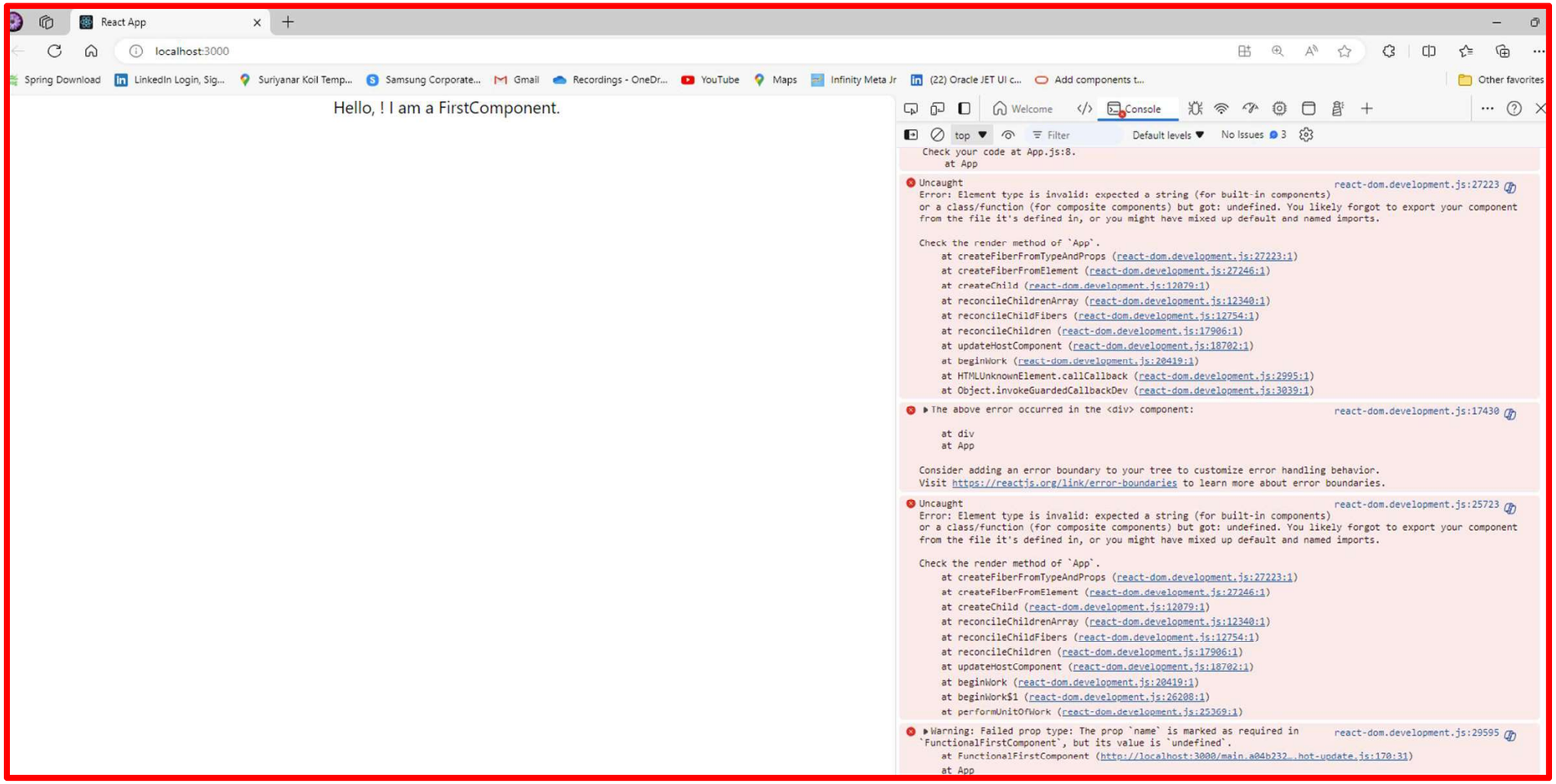
- In many applications there are smart components that hold state but render dumb components that simply receive
- props and return HTML as JSX. Stateless functional components are much more reusable and have a positive
- performance impact on your application.
- They have 2 main characteristics:
  - 1. When rendered they receive an object with all the props that were passed down
  - 2. They must return the JSX to be rendered

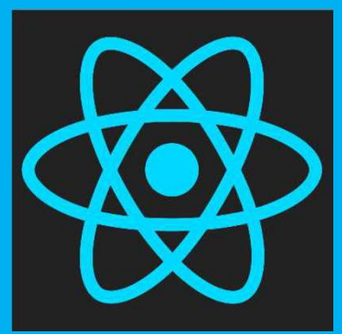
Stateless Functional Components




EXPLORER
REACT
components-demo
node_modules
public
src
App.css
App.js M
App.test.js
FirstComponent.js U
FunctionalFirstComponent.js U
index.css
index.js
logo.svg
reportWebVitals.js
setupTests.js
.gitignore
package-lock.json
FunctionalFirstComponent.js U
App.js M
components-demo > src > App.js > App
1 import './App.css';
2 import FirstComponent from './FirstComponent';
3 import FunctionalFirstComponent from './FunctionalFirstComponent';
4
5 function App() {
6 return (
7
8
9 {/* <FirstComponent name={'User'} /> */}
10
11 <FunctionalFirstComponent />
12
13);
14 }
15
16 export default App;
17

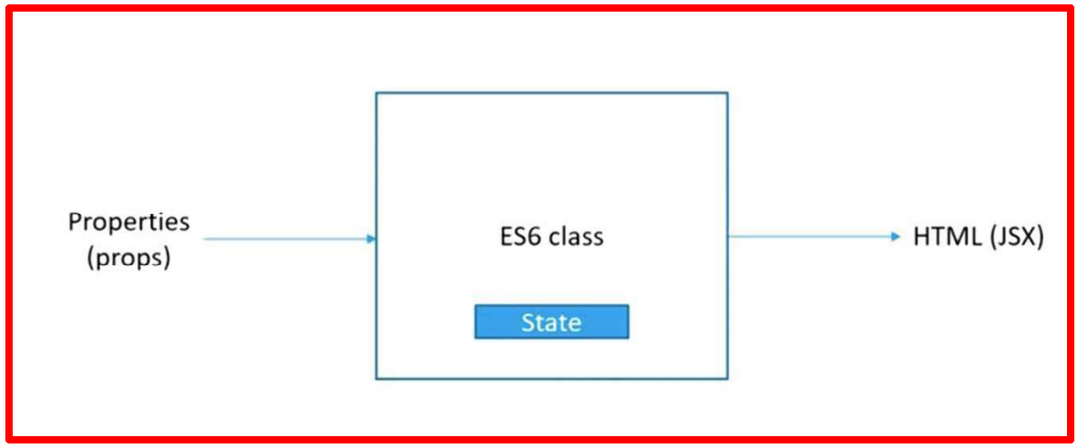
Output With Validation Error





The Class Based Components

- Class components are basically ES6 classes similar to a functional component
- A class component also can optionally receive props as input and return HTML apart from the props a class component can also maintain a private internal state.
- In simpler words it can maintain some information which is private to that component and use that information to describe the user interface



Stateful Components

- In contrast to the 'stateless' components shown above, 'stateful' components have a state object that can be
- updated with the setState method. The state must be initialized in the constructor before it can be set:

EXPLORER

LESSON03-HELLOWORLD

- my-app
  - node_modules
  - public
  - src
    - components
      - Greet.js U
      - Welcome.js U
    - App.css
    - App.js M

JS App.js M | JS Welcome.js U

my-app > src > components > JS Welcome.js > default

```
1  import React, { Component } from 'react';
2
3  class Welcome extends Component {
4      render() {
5          return <h1>Class Component</h1>
6      }
7  }
8
9  export default Welcome
```

JS App.js M | JS Welcome.js U

my-app > src > JS App.js > App

```
1  import logo from './logo.svg';
2  import './App.css';
3  import { Greet } from './components/Greet';
4  import Welcome from './components/Welcome';
5  function App() {
6    return (
7      <div className="App">
8        <Greet />
9        <Welcome />
10     </div>
11   );
12 }
13
14 export default App;
15
```

localhost:3000

LinkedIn Login, Sign... | Suriyanar Koil Templ... | Samsung Corporate... | Gmail | Recordings - OneDr... | How JSON Schema... | YouTube | Maps | Infinity Meta Jr | (22) Oracle JET UI c... | Add components to... | Printable A

Hello React !

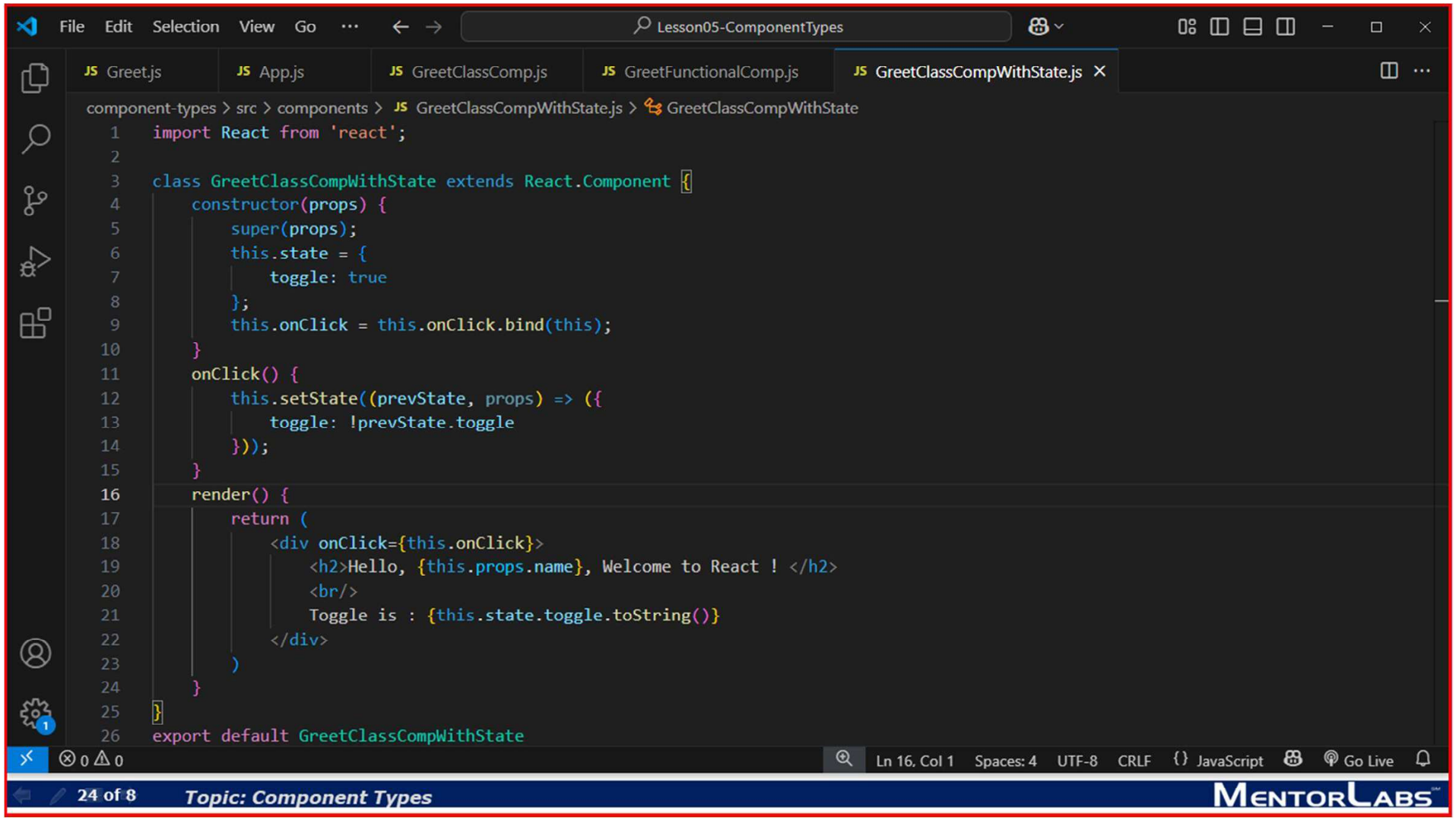
Class Component

Functional

- Simple functions
- Use Func components as much as possible
- Absence of ‘this’ keyword
- Solution without using state
- Mainly responsible for the UI
- Stateless/ Dumb/ Presentational

Class

- More feature rich
- Maintain their own private data - state
- Complex UI logic
- Provide lifecycle hooks
- Stateful/ Smart/ Container


File Edit Selection View Go
Lesson05-ComponentTypes
Greet.js
App.js
GreetClassComp.js
GreetFunctionalComp.js
GreetClassCompWithState.js
component-types > src > components > GreetClassCompWithState.js > GreetClassCompWithState
import React from 'react';
class GreetClassCompWithState extends React.Component {
constructor(props) {
super(props);
this.state = {
toggle: true
};
this.onClick = this.onClick.bind(this);
}
onClick() {
this.setState((prevState, props) => ({
toggle: !prevState.toggle
}));
}
render() {
return (

<h2>Hello, {this.props.name}, Welcome to React ! </h2>

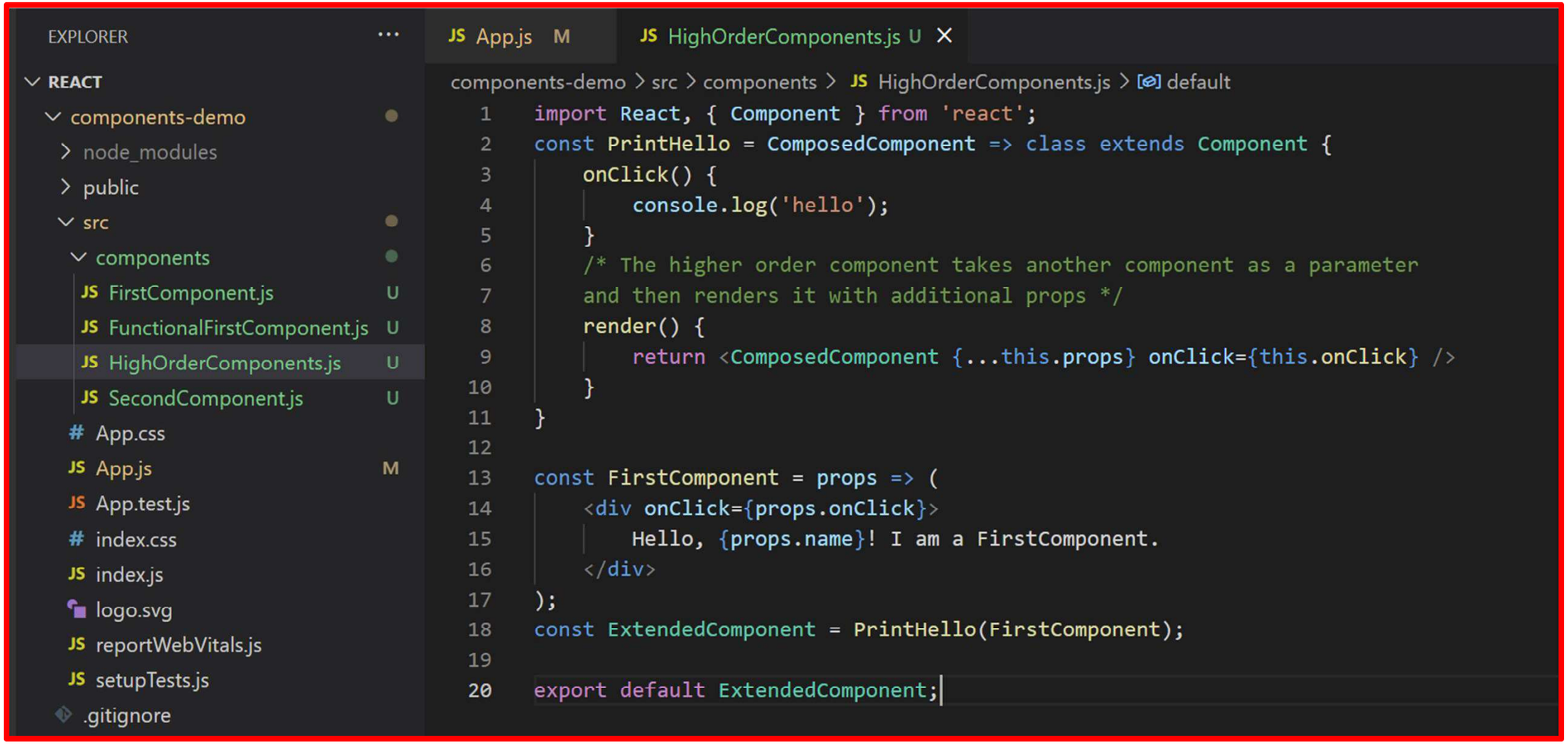
Toggle is : {this.state.toggle.toString()}

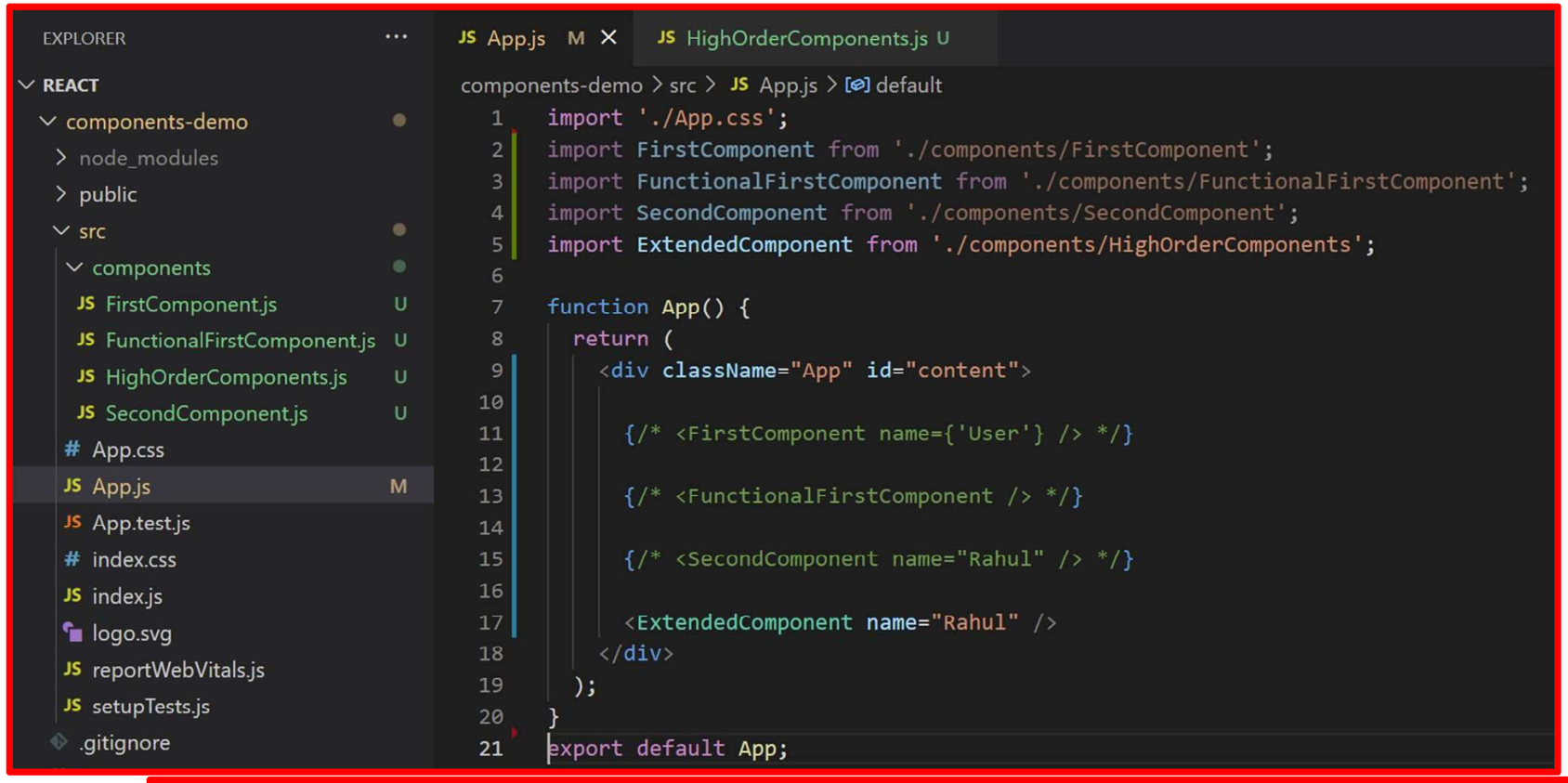
)
}
}
export default GreetClassCompWithState
Ln 16, Col 1
Spaces: 4
UTF-8
CRLF
JavaScript
Go Live
24 of 8
Topic: Component Types
MentorLabs

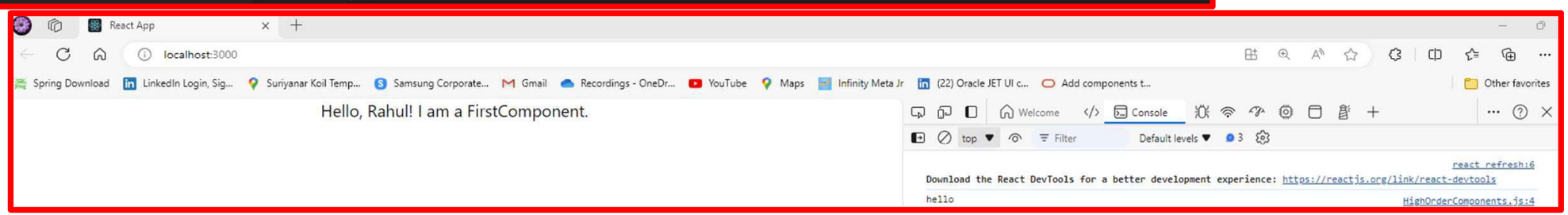
Higher Order Components

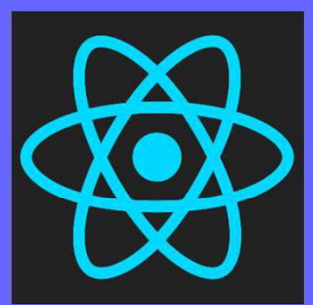
- Higher order components (HOC) allow to share component functionality.

- Higher order components are used when you want to share logic across several components regardless of how different they render.


EXPLORER
REACT
components-demo
node_modules
public
src
components
FirstComponent.js U
FunctionalFirstComponent.js U
HighOrderComponents.js U
SecondComponent.js U
App.css
App.js M
App.test.js
index.css
index.js
logo.svg
reportWebVitals.js
setupTests.js
.gitignore
App.js M
HighOrderComponents.js U
components-demo > src > components > HighOrderComponents.js > default
1 import React, { Component } from 'react';
2 const PrintHello = ComposedComponent => class extends Component {
3 onClick() {
4 console.log('hello');
5 }
6 /* The higher order component takes another component as a parameter
7 and then renders it with additional props */
8 render() {
9 return <ComposedComponent {...this.props} onClick={this.onClick} />
10 }
11 }
12
13 const FirstComponent = props => (
14
15 Hello, {props.name}! I am a FirstComponent.
16
17);
18 const ExtendedComponent = PrintHello(FirstComponent);
19
20 export default ExtendedComponent;


EXPLORER
REACT
components-demo
node_modules
public
src
components
FirstComponent.js U
FunctionalFirstComponent.js U
HighOrderComponents.js U
SecondComponent.js U
App.css
App.js M
App.test.js
index.css
index.js
logo.svg
reportWebVitals.js
setupTests.js
.gitignore
App.js M
HighOrderComponents.js U
components-demo > src > App.js > default
1 import './App.css';
2 import FirstComponent from './components/FirstComponent';
3 import FunctionalFirstComponent from './components/FunctionalFirstComponent';
4 import SecondComponent from './components/SecondComponent';
5 import ExtendedComponent from './components/HighOrderComponents';
6
7 function App() {
8 return (
9
10
11 {/* <FirstComponent name={'User'} /> */}
12
13 {/* <FunctionalFirstComponent /> */}
14
15 {/* <SecondComponent name="Rahul" /> */}
16
17 <ExtendedComponent name="Rahul" />
18
19);
20 }
21 export default App;


React App
localhost:3000
Hello, Rahul! I am a FirstComponent.
Welcome
Console
Download the React DevTools for a better development experience: https://reactjs.org/link/react-devtools
hello

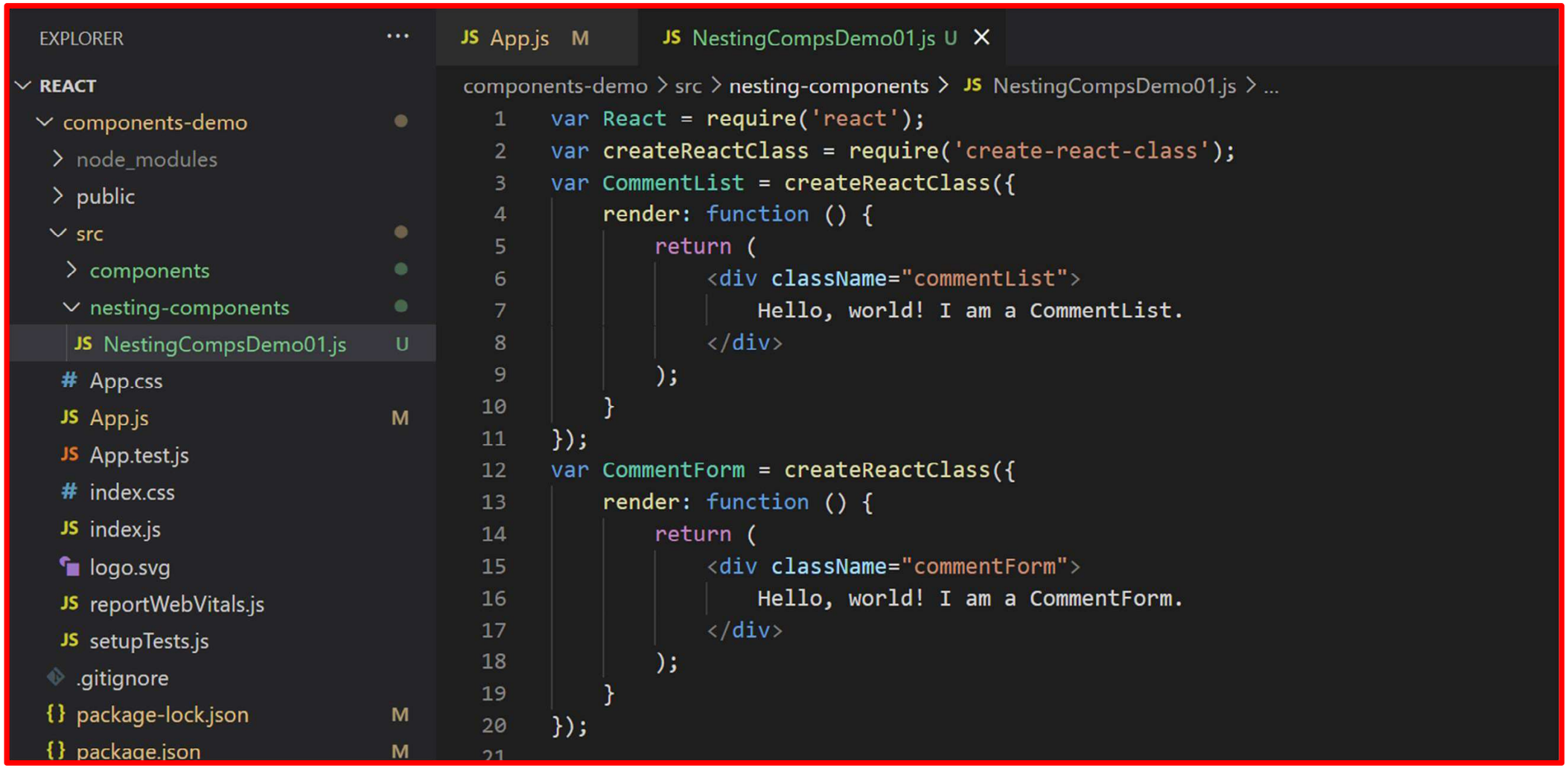


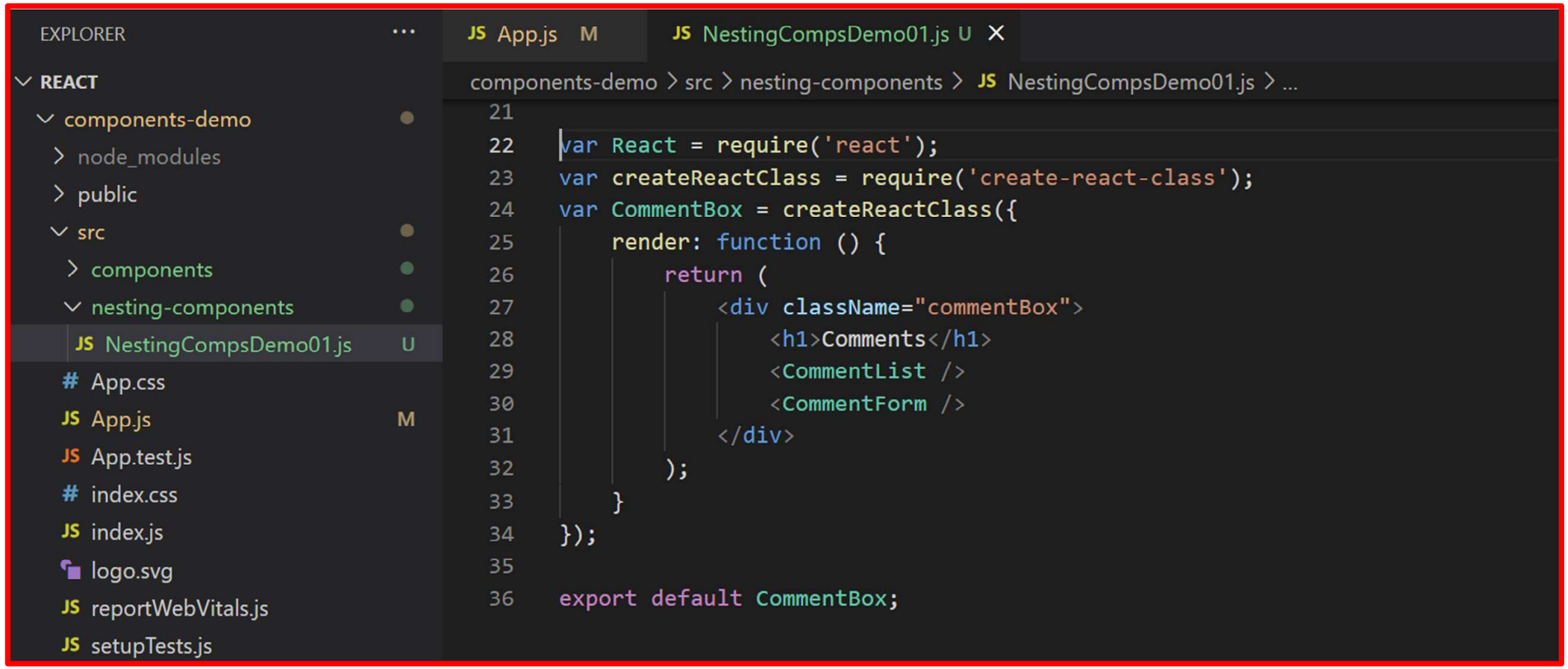
Nesting Components

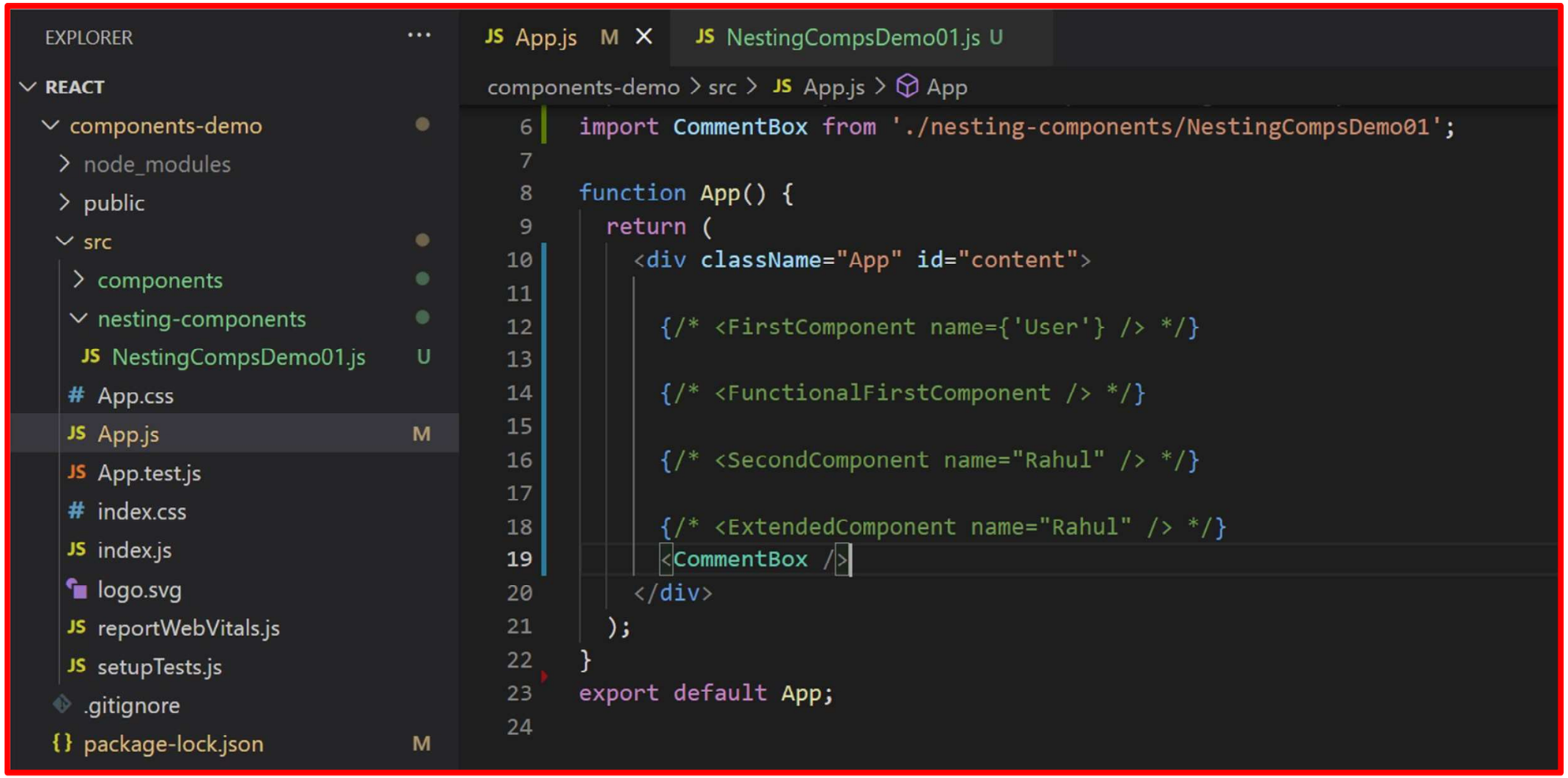
Nesting Components

- A lot of the power of ReactJS is its ability to allow nesting of components.

```
F:\Training\IQ Gateway\2024\Workspaces\React\components-demo>npm i create-react-class
```

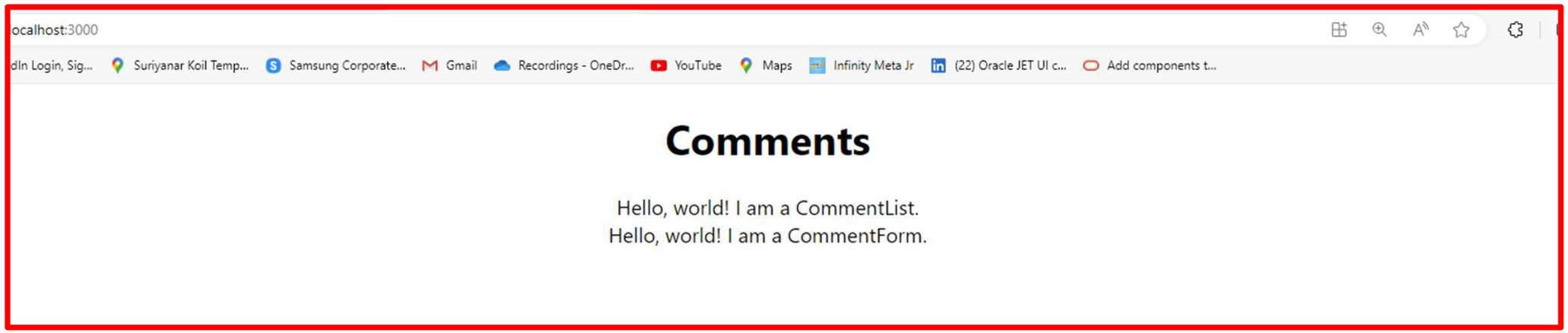

EXPLORER
REACT
components-demo
node_modules
public
src
components
nesting-components
NestingCompsDemo01.js U
App.css
App.js M
App.test.js
index.css
index.js
logo.svg
reportWebVitals.js
setupTests.js
.gitignore
package-lock.json M
package.json M
App.js M
NestingCompsDemo01.js U
components-demo > src > nesting-components > NestingCompsDemo01.js > ...
1 var React = require('react');
2 var createReactClass = require('create-react-class');
3 var CommentList = createReactClass({
4 render: function () {
5 return (
6
7 Hello, world! I am a CommentList.
8
9);
10 }
11 });
12 var CommentForm = createReactClass({
13 render: function () {
14 return (
15
16 Hello, world! I am a CommentForm.
17
18);
19 }
20 });

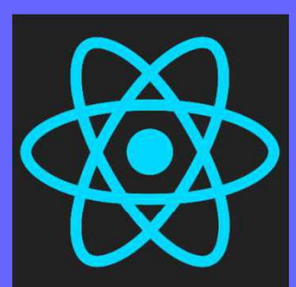

EXPLORER
REACT
components-demo
node_modules
public
src
components
nesting-components
NestingCompsDemo01.js U
App.css
App.js M
App.test.js
index.css
index.js
logo.svg
reportWebVitals.js
setupTests.js
App.js M
NestingCompsDemo01.js U
components-demo > src > nesting-components > NestingCompsDemo01.js > ...
21
22 var React = require('react');
23 var createReactClass = require('create-react-class');
24 var CommentBox = createReactClass({
25 render: function () {
26 return (
27
28 <h1>Comments</h1>
29 <CommentList />
30 <CommentForm />
31
32);
33 }
34 });
35
36 export default CommentBox;


EXPLORER
REACT
components-demo
node_modules
public
src
components
nesting-components
NestingCompsDemo01.js U
App.css
App.js M
App.test.js
index.css
index.js
logo.svg
reportWebVitals.js
setupTests.js
.gitignore
package-lock.json M
App.js M
NestingCompsDemo01.js U
components-demo > src > App.js > App
import CommentBox from './nesting-components/NestingCompsDemo01';
function App() {
return (

{/* <FirstComponent name={'User'} /> */}
{/* <FunctionalFirstComponent /> */}
{/* <SecondComponent name="Rahul" /> */}
{/* <ExtendedComponent name="Rahul" /> */}
<CommentBox />

);
}
export default App;


ocalhost:3000
dIn Login, Sig...
Suriyanar Koil Temp...
Samsung Corporate...
Gmail
Recordings - OneDr...
YouTube
Maps
Infinity Meta Jr
(22) Oracle JET UI c...
Add components t...
Comments
Hello, world! I am a CommentList.
Hello, world! I am a CommentForm.



Props

Props

- Props are a way to pass information into a React component, they can have any type including functions -
- sometimes referred to as callbacks.
- In JSX props are passed with the attribute syntax

```
<MyComponent userID={123} />
```

Inside the definition for MyComponent userID will now be accessible from the props object

```
// The render function inside MyComponent
render() {
    return (
        <span>The user's ID is {this.props.userID}</span>
    )
}
```

- It's important to define all props, their types, and where applicable, their default value:

```
// defined at the bottom of MyComponent
MyComponent.propTypes = {
    someObject: React.PropTypes.object,
    userID: React.PropTypes.number.isRequired,
    title: React.PropTypes.string
};

MyComponent.defaultProps = {
    someObject: {},
    title: 'My Default Title'
}
```

In this example the prop `someObject` is optional, but the prop `userID` is required. If you fail to provide `userID` to `MyComponent`, at runtime the React engine will show a console warning you that the required prop was not provided. Beware though, this warning is only shown in the development version of the React library, the production version will not log any warnings.

Summary

In this lesson, you should have learned how to:

- Creating Components
- Basic Component
- Nesting Components
- Props