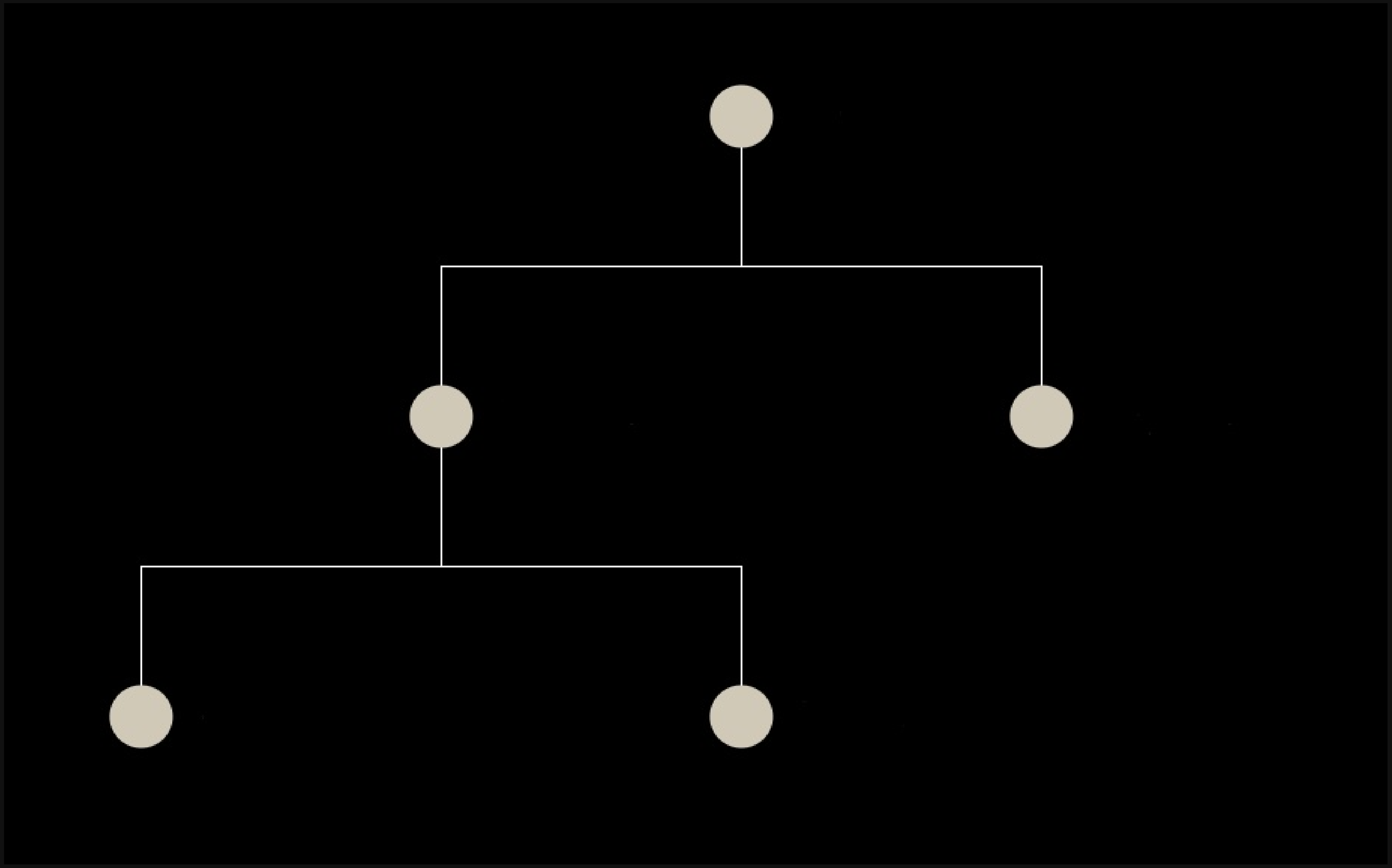
REACT CODE GENERATOR

FROM TREE

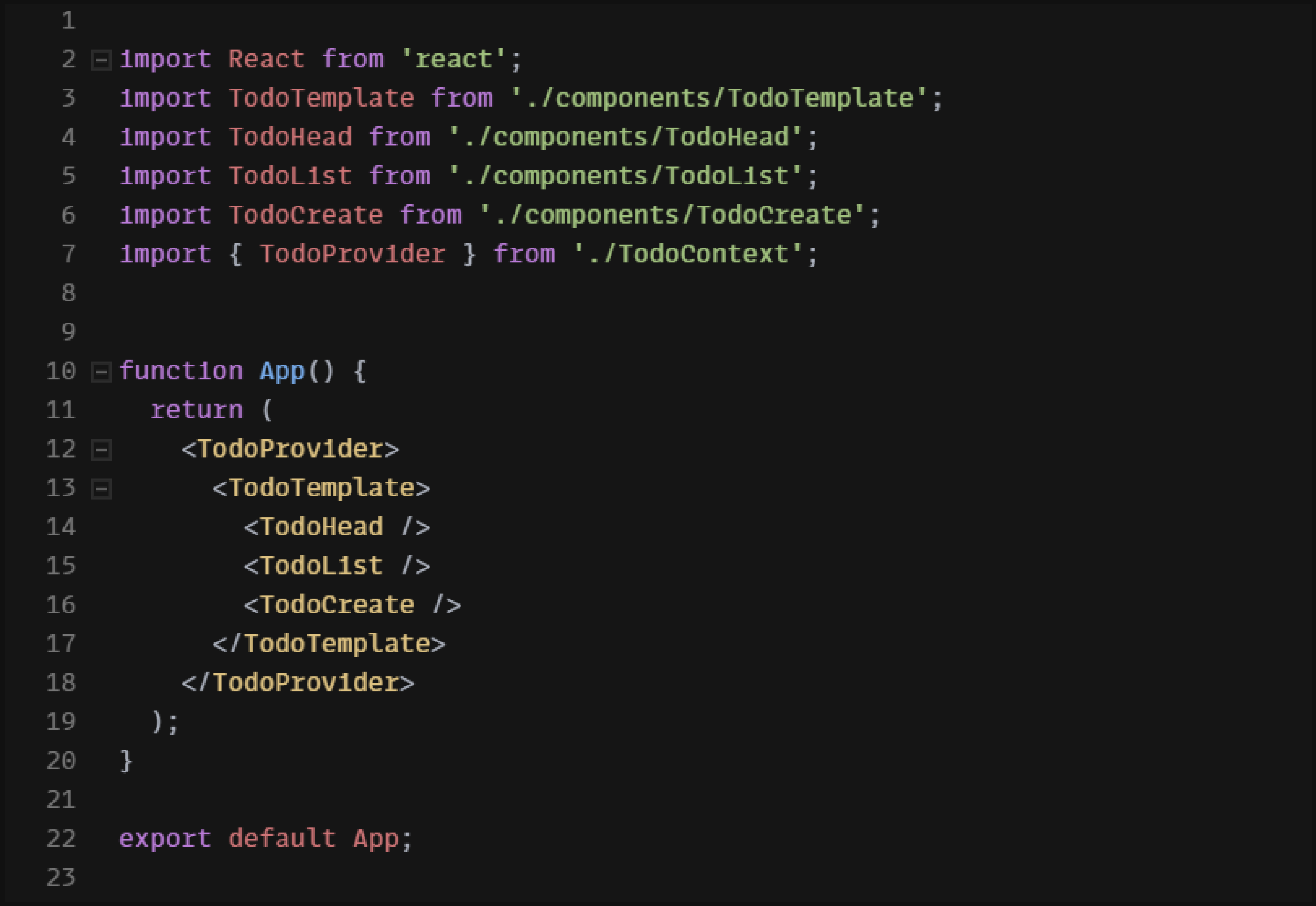
Open Source SW Project Team 1
Hercules Health Gym

OVERVIEW

FROM TREE

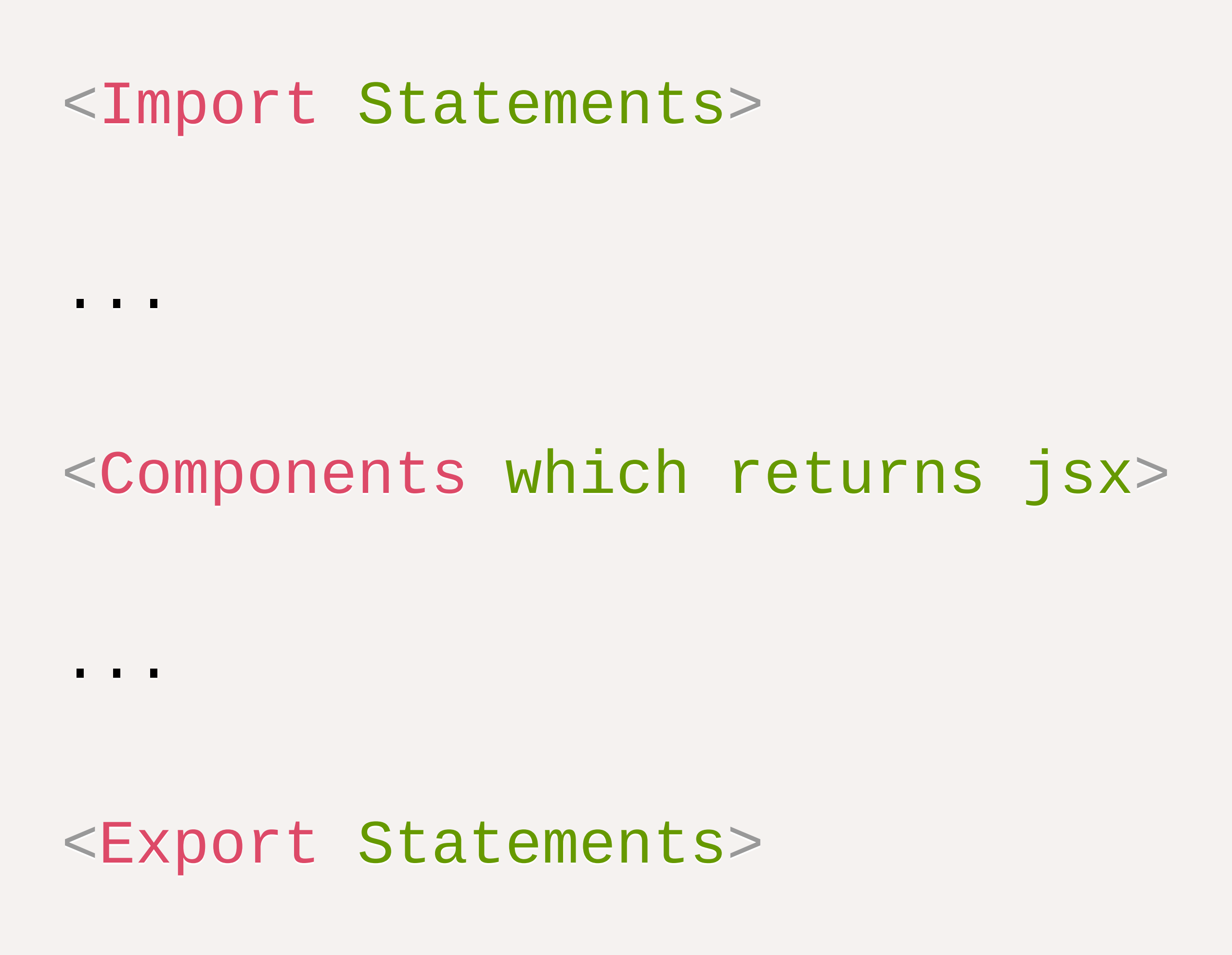


TO CODE



```

import React from 'react';
import TodoTemplate from './components/TodoTemplate';
import TodoHead from './components/TodoHead';
import TodoList from './components/TodoList';
import TodoCreate from './components/TodoCreate';
import { TodoProvider } from './TodoContext';


function App() {
  return (
    <TodoProvider>
      <TodoTemplate>
        <TodoHead />
        <TodoList />
        <TodoCreate />
      </TodoTemplate>
    </TodoProvider>
  );
}

export default App;

```

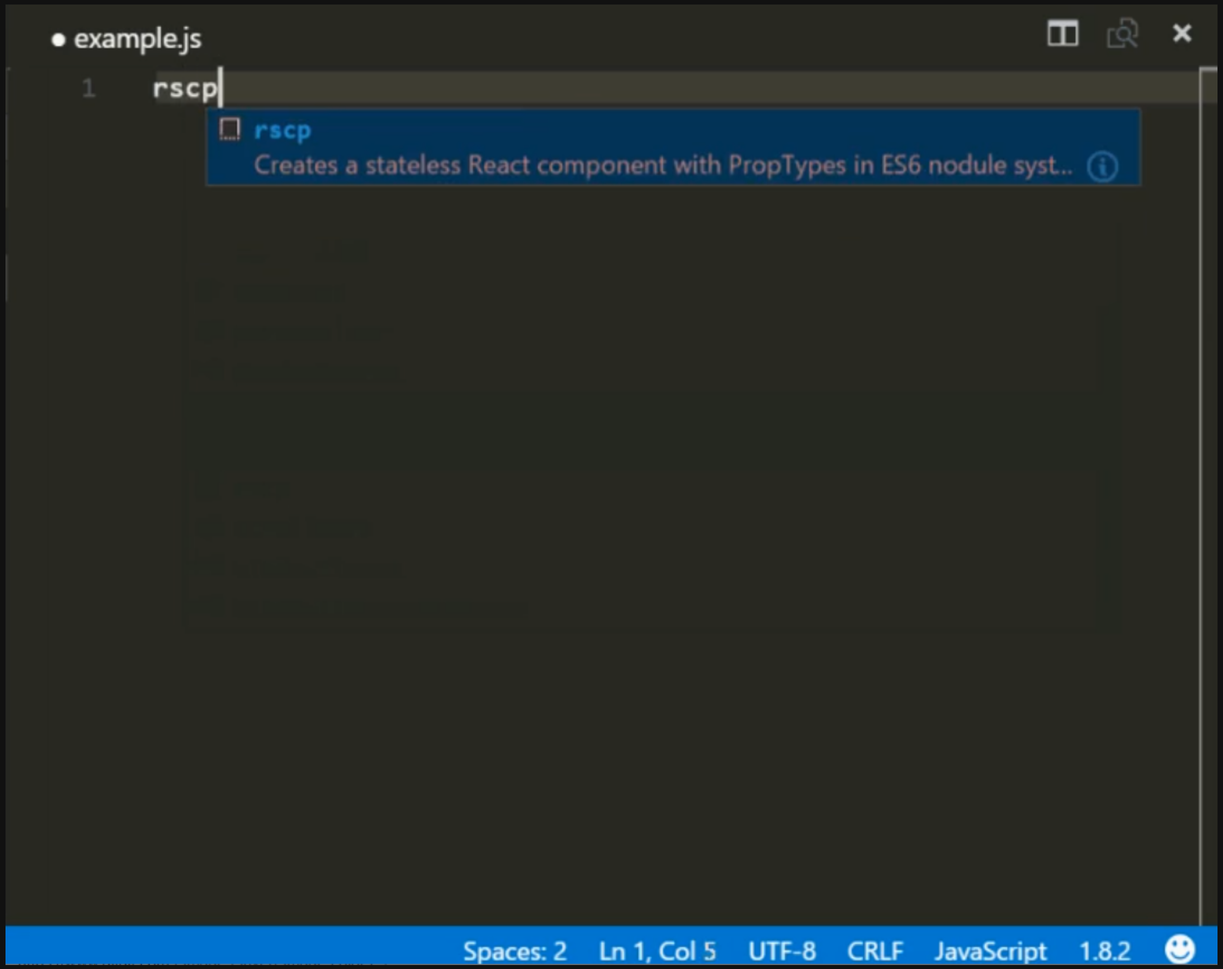
NECESSITY



```
<Import Statements>

...

<Components which returns jsx>

...

<Export Statements>
```

LOTS OF REPEATED WORK

LOTS OF CODE SNIPPETS

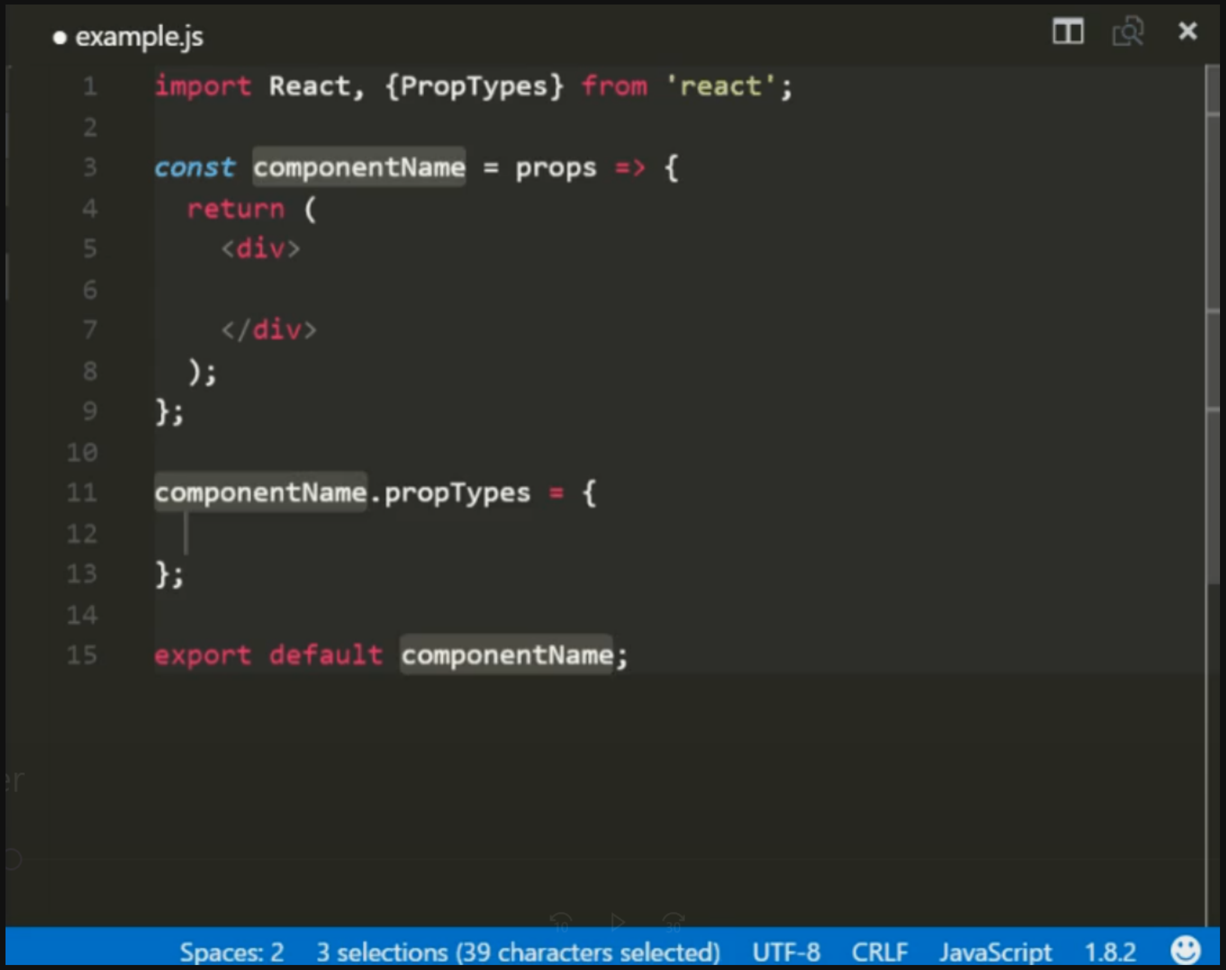


```
rscp
```

rscp

Creates a stateless React component with PropTypes in ES6 nodule syst..

LOTS OF CODE SNIPPETS



```
import React, {PropTypes} from 'react';

const componentName = props => {
  return (
    <div>

    </div>
  );
};

componentName.propTypes = {

};

export default componentName;
```

Spaces: 2   3 selections (39 characters selected)   UTF-8   CRLF   JavaScript   1.8.2

SIMPLIFY

THOSE PROCESSES

WITH OUR SW

DIFFERENTIATION ?

DIFFERENTIATION ?

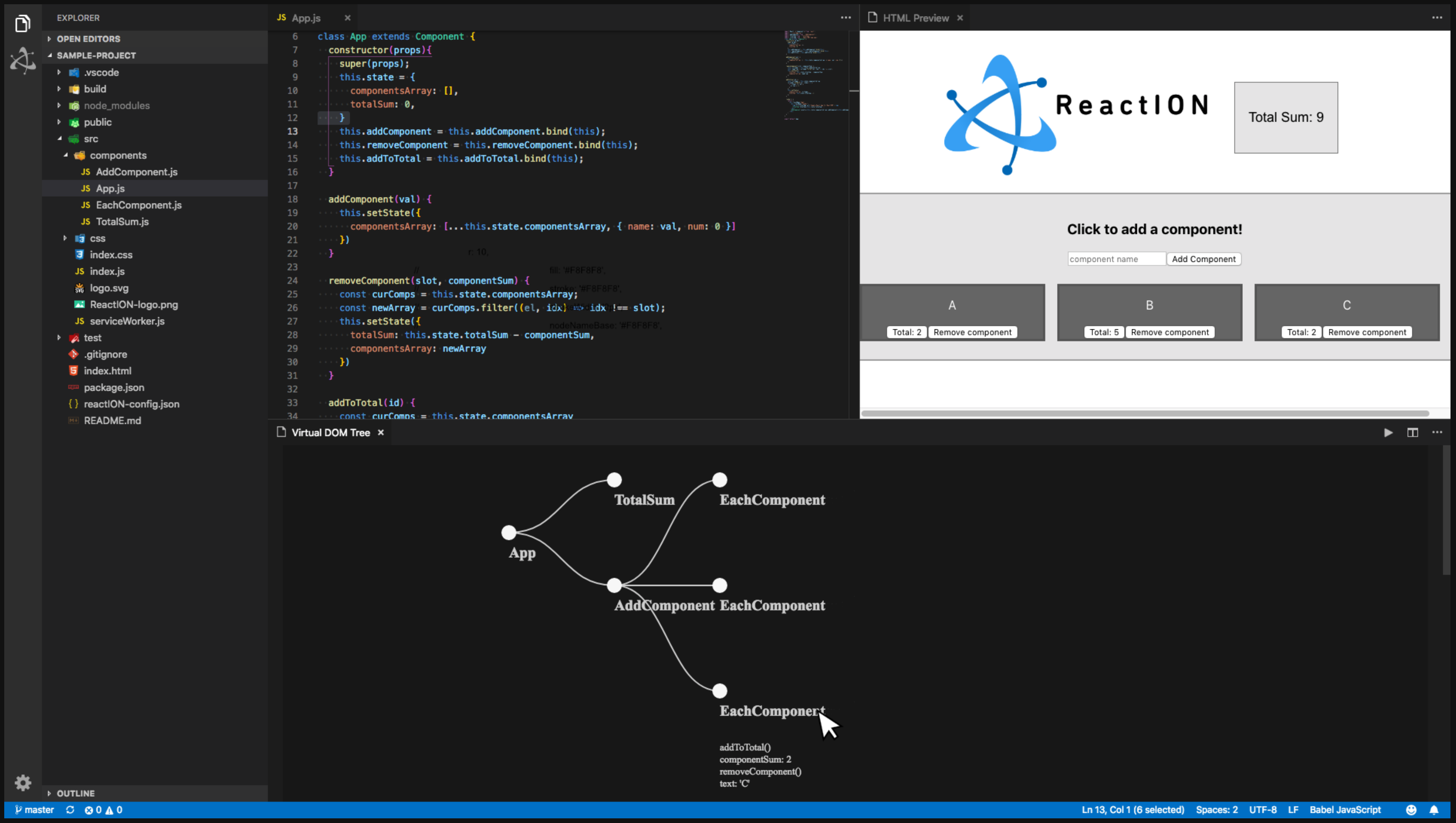
NONE OF OPEN SOURCE TOOL FOR THIS

DIFFERENTIATION ?

ONLY FOR OPPOSING ACTION

DIFFERENTIATION ?

ONLY FOR OPPOSING ACTION



```
class App extends Component {
  constructor(props){
    super(props);
    this.state = {
      componentsArray: [],
      totalSum: 0,
    }
    this.addComponent = this.addComponent.bind(this);
    this.removeComponent = this.removeComponent.bind(this);
    this.addToTotal = this.addToTotal.bind(this);
  }

  addComponent(val) {
    this.setState({
      componentsArray: [...this.state.componentsArray, { name: val, num: 0 }]
    })
  }

  removeComponent(slot, componentSum) {
    const curComps = this.state.componentsArray;
    const newArray = curComps.filter((el, idx) => idx !== slot);
    this.setState({
      totalSum: this.state.totalSum - componentSum,
      componentsArray: newArray
    })
  }

  addToTotal(id) {
    const curComps = this.state.componentsArray
```

Total Sum: 9

Click to add a component!

A — Total: 2

B — Total: 5

C — Total: 2

EachComponent

addToTotal()
componentSum: 2
removeComponent()
text: 'C'

Ln 13, Col 1 (6 selected) Spaces: 2 UTF-8 LF Babel JavaScript

ReactION-js/ReactION

DEVELOPMENT PLAN

DEVELOPMENT PLAN

LANGUAGE

TypeScript / React

DEVELOPMENT PLAN

COLLABORATION

Git

DEVELOPMENT PLAN

COLLABORATION

GitHub

DEVELOPMENT PLAN

COLLABORATION

Git Flow

DEVELOPMENT PLAN

COLLABORATION

Agile

DEVELOPMENT PLAN

COLLABORATION

Agile Khanban

DEVELOPMENT PLAN

MINIMUM VIABLE PRODUCT

DEVELOPMENT SCOPE

VSCode Extension

DEVELOPMENT PLAN

MINIMUM VIABLE PRODUCT

DEVELOPMENT SCOPE

Just From Tree To Code

DEVELOPMENT PLAN

MINIMUM VIABLE PRODUCT

DEVELOPMENT SCOPE

Focus on Basic Feature

DEVELOPMENT PLAN

DISTRIBUTING ROLES

DEVELOPMENT PLAN

DISTRIBUTING ROLES

PROJECT MANAGEMENT

임동영

DEVELOPMENT PLAN

DISTRIBUTING ROLES

UI DEVELOPMENT

정용준 / 조원희

DEVELOPMENT PLAN

DISTRIBUTING ROLES

LOGIC DEVELOPMENT

문법식 / 박준수

OPEN SOURCE SW

FOR REFERENCE OR USE

OPEN SOURCE SW

FOR REFERENCE OR USE

< REACT >

UI IMPLEMENTATION

https://github.com/facebook/react

OPEN SOURCE SW

FOR REFERENCE OR USE

< REACT D3 TREE >

IMPLEMENT TREE USING REACT

https://github.com/bkrem/react-d3-tree

OPEN SOURCE SW

FOR REFERENCE OR USE

< GENERATE REACT CODE >

REFERENCE CODE GENERATING MODULE

https://github.com/JPStrydom/Generate-React-Code

OPEN SOURCE SW

FOR REFERENCE OR USE

< REACTION >

REFERENCE THE IDEA OF PROJECT

which analyze react project and generate tree

https://github.com/ReactION-js/ReactION

LICENSE

MIT LICENSE

FIN

Q & A