React

JS Library for creating UI

react.dev

card component ⟷ function

reusable

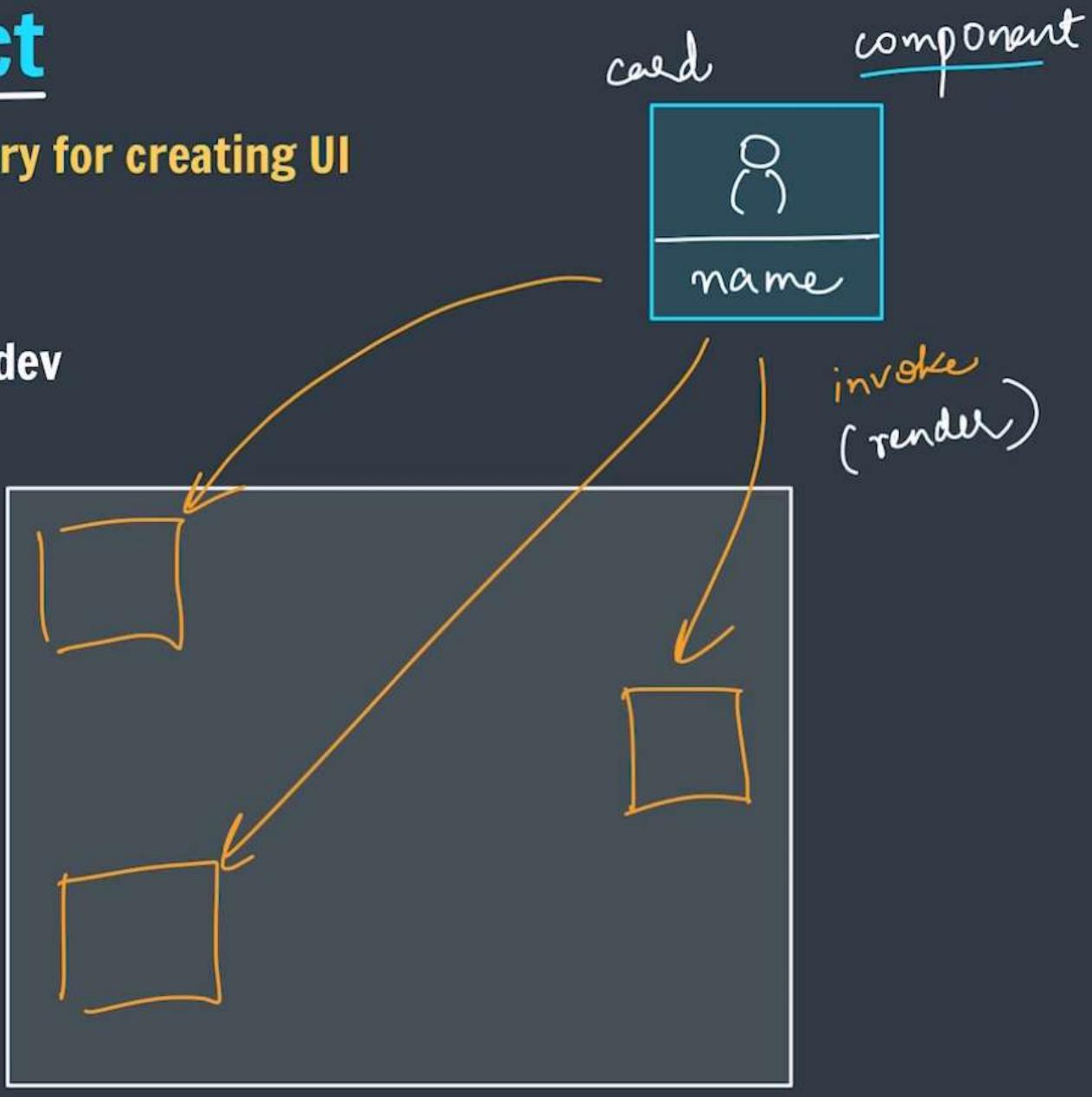
```
func sum ( a , b) {
    return a + b ;
}
```

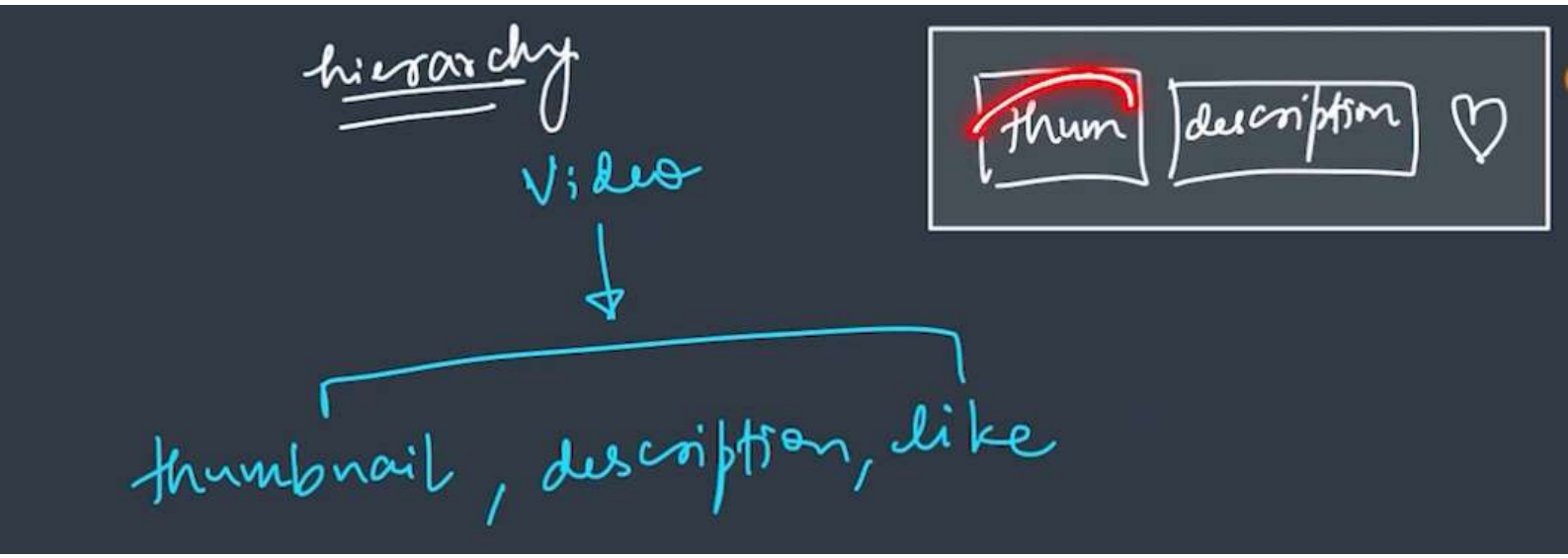
invoke (call)

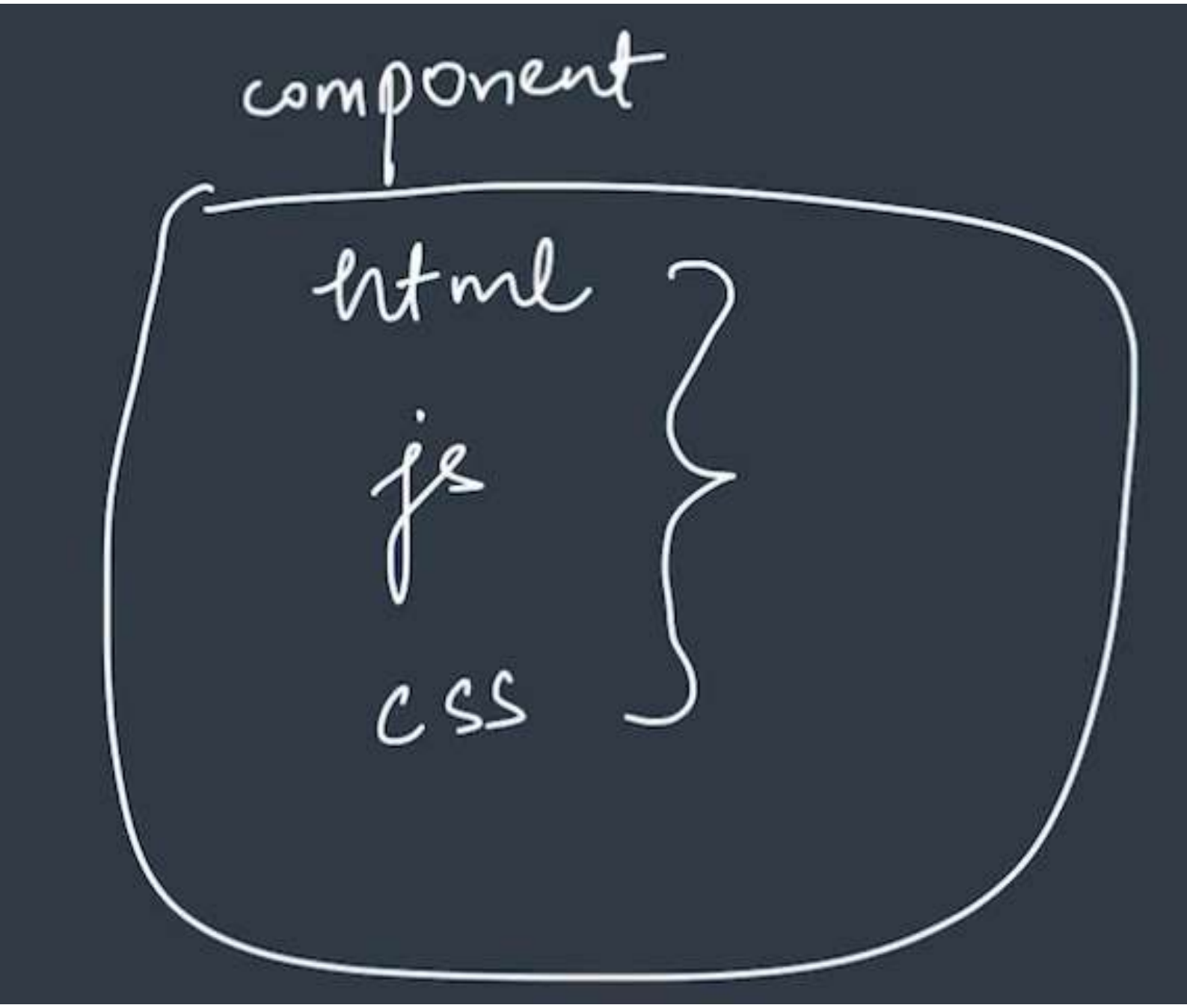
React

JS Library for creating UI

react.dev




hierarchy
Video
thumbnail, description, like
thum
description


component
html
js
css

React

JS Library for creating UI

react.dev

JSX

JavaScript Extension Syntax

It lets us write HTML directly inside JS

JSX

not real JS

JSX → Babel → JS

JavaScript Extension Syntax

It lets us write HTML directly inside JS

transpile (Babel)

JS

Set up Local Environment

React

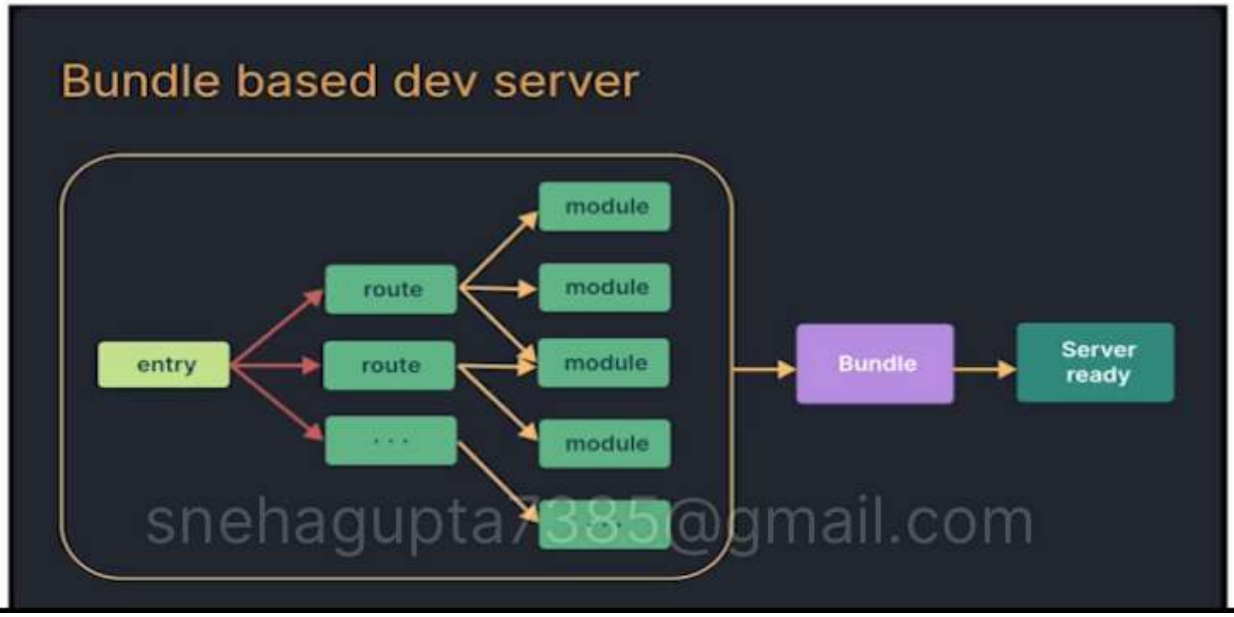
Create-React-App v/s Vite

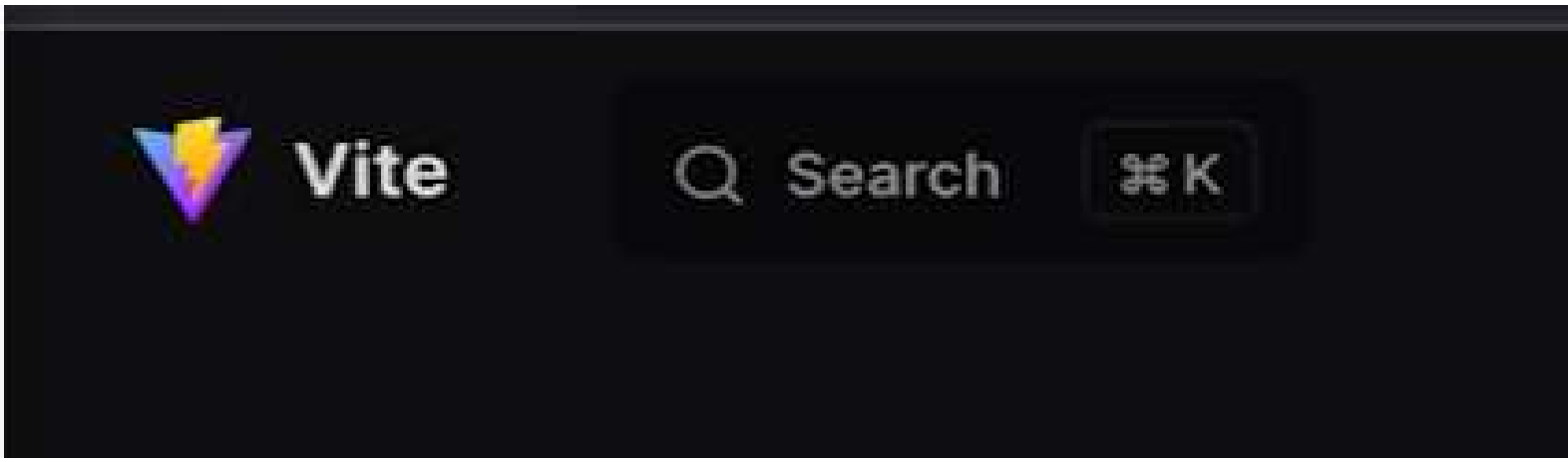
```
npm create vite@latest

npm run dev        // to start the server
```

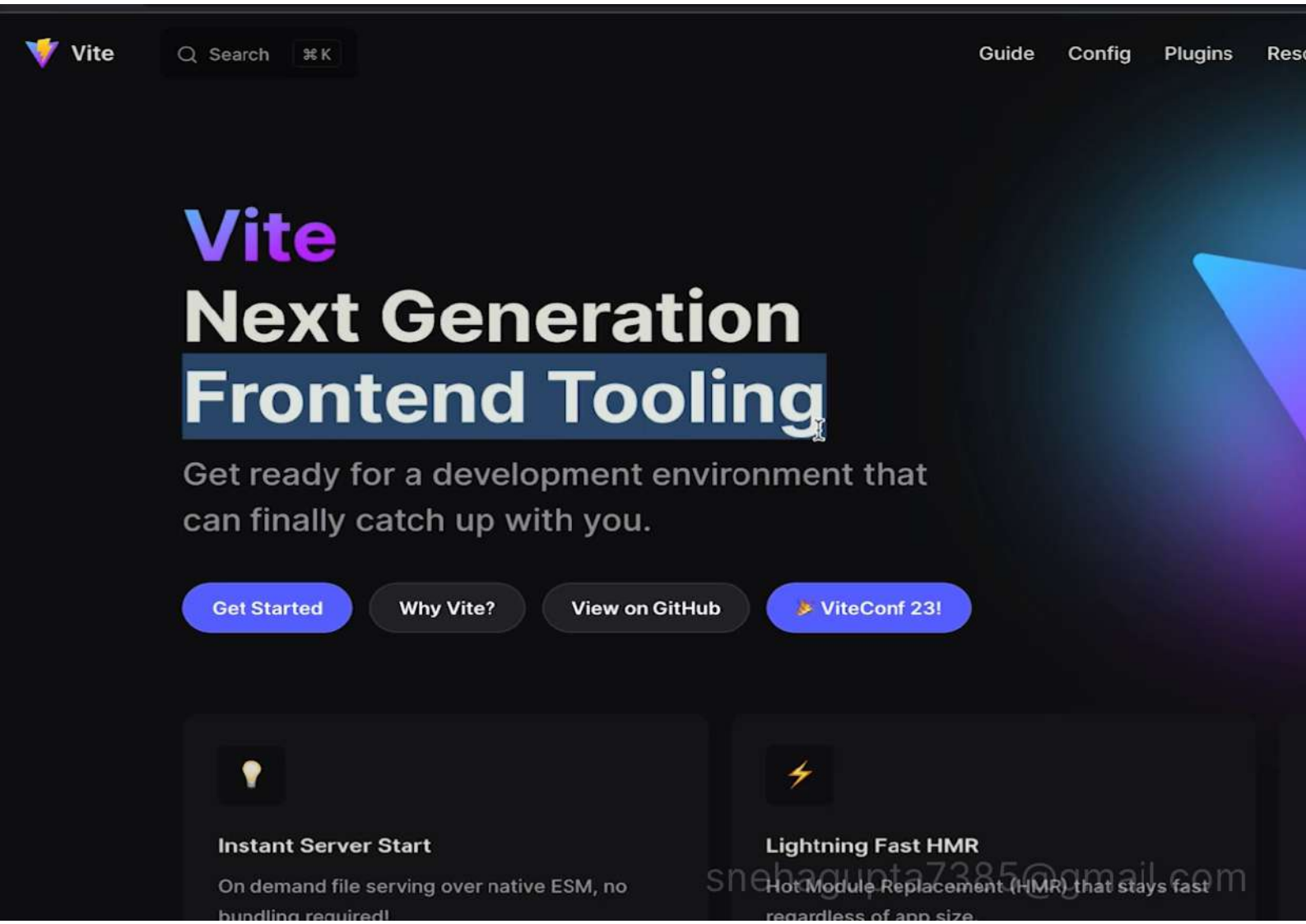
Why is CRA slow?

CRA uses a webpack under the hood. The webpack bundles the entire application code before it can be served. With a large codebase, it takes more time to spin up the development server and reflecting the changes made takes a long time.

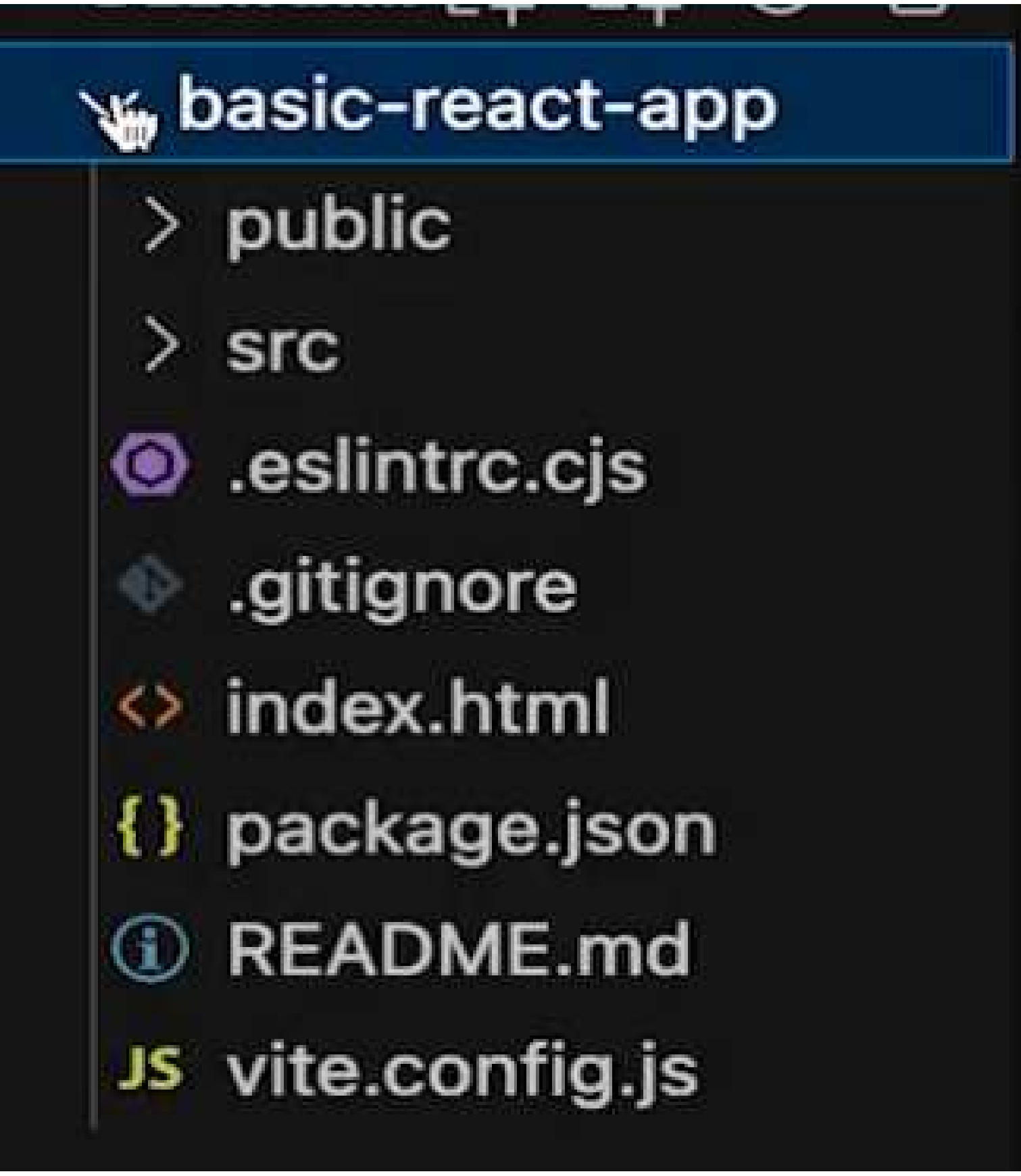
The diagram below shows how all the code must be bundled in order to start a bundle-based dev server.




Vite
Search
⌘K



```
shradhakhapra@Shradhas-MacBook-Air DeltaReact % npm create vite@latest
✔ Project name: … basic-react-app
✔ Select a framework: › React
✔ Select a variant: › JavaScript

Scaffolding project in /Users/shradhakhapra/WebClassroom/DeltaReact

Done. Now run:

  cd basic-react-app
  npm install
  npm run dev
```


basic-react-app
public
src
.eslintrc.cjs
.gitignore
index.html
package.json
README.md
vite.config.js

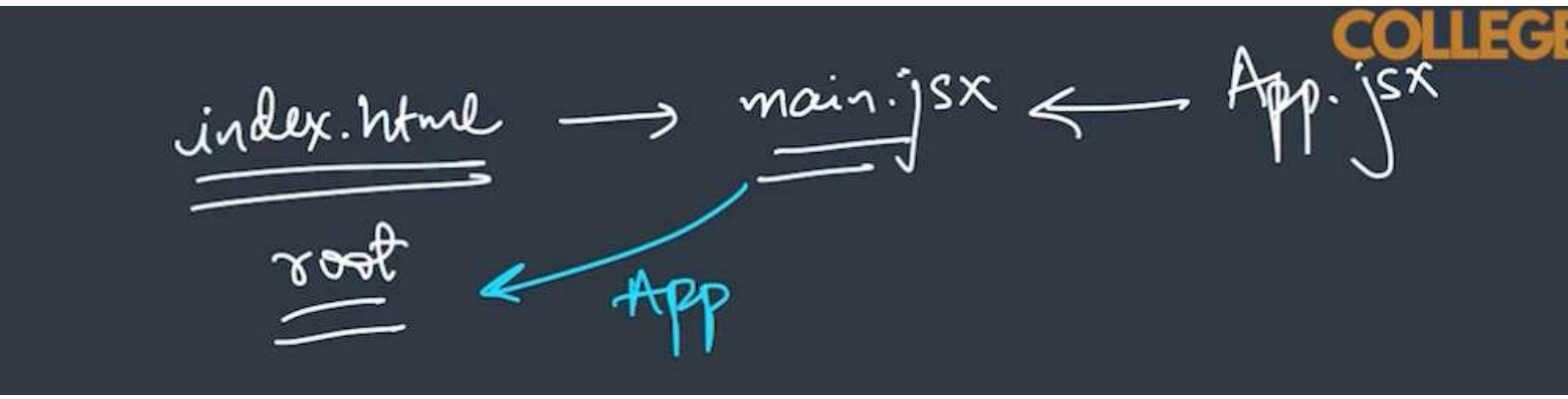
```
shradhakhapra@Shradhas-MacBook-Air DeltaReact % cd basic-react-app
shradhakhapra@Shradhas-MacBook-Air basic-react-app % npm install

added 269 packages, and audited 270 packages in 11s

97 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

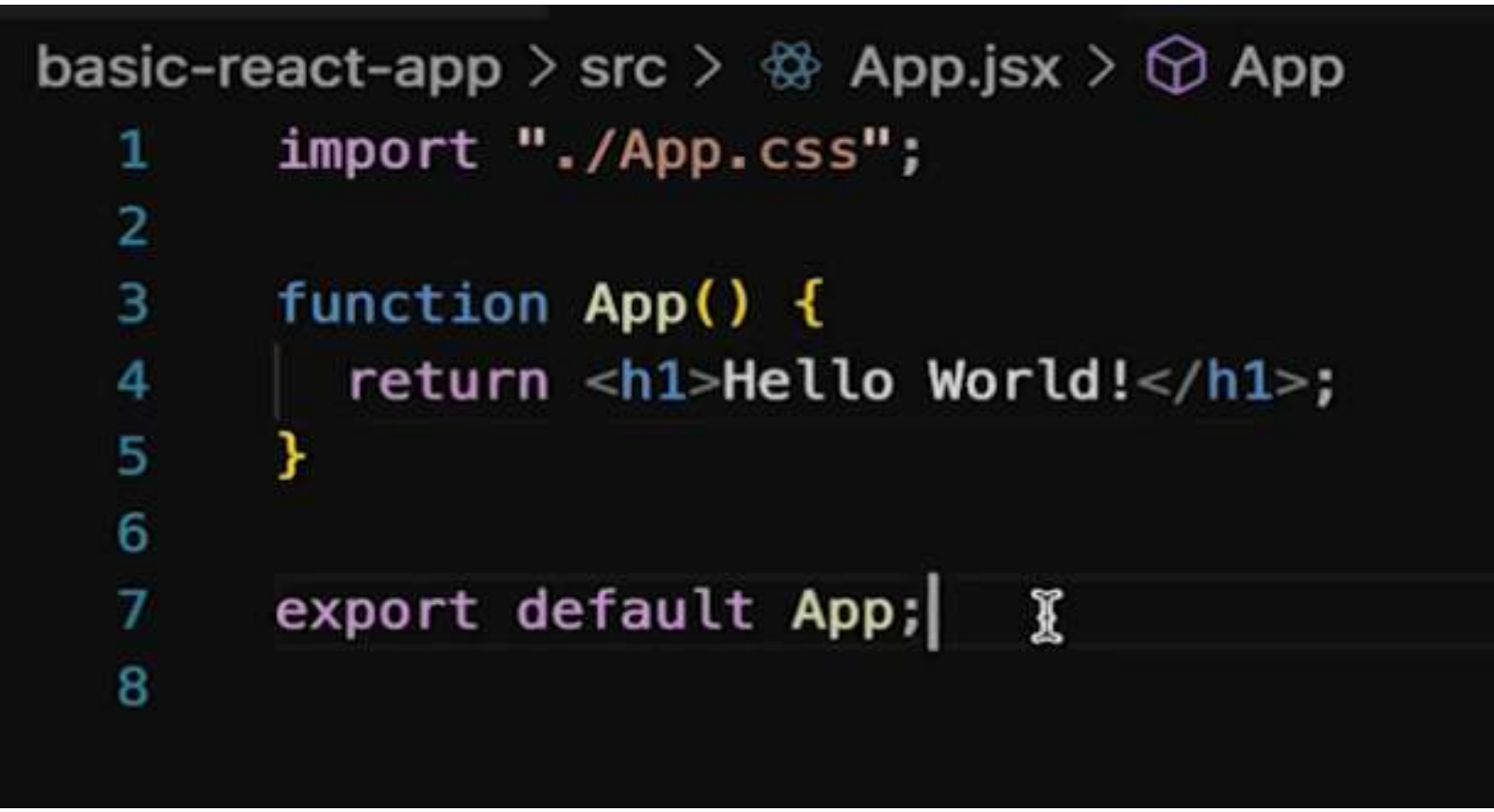
Set up Local Environment

Create-React-App v/s Vite

```
npm create vite@latest

npm run dev        // to start the server
```

Understanding our App


index.html
main.jsx
App.jsx
root
App


basic-react-app > src > App.jsx > App
1 import "./App.css";
2
3 function App() {
4 return <h1>Hello World!</h1>;
5 }
6
7 export default App;
8


Hello World!


return <button>Hello World</button>;


Hello World

Our 1st Component

Component is a reusable & independent piece of code.

Creating a component

```
function Title() {
  return(
    <h1>Hello World! </h1>
  );
}
```

Rendering a component

```
<Title></Title>
```

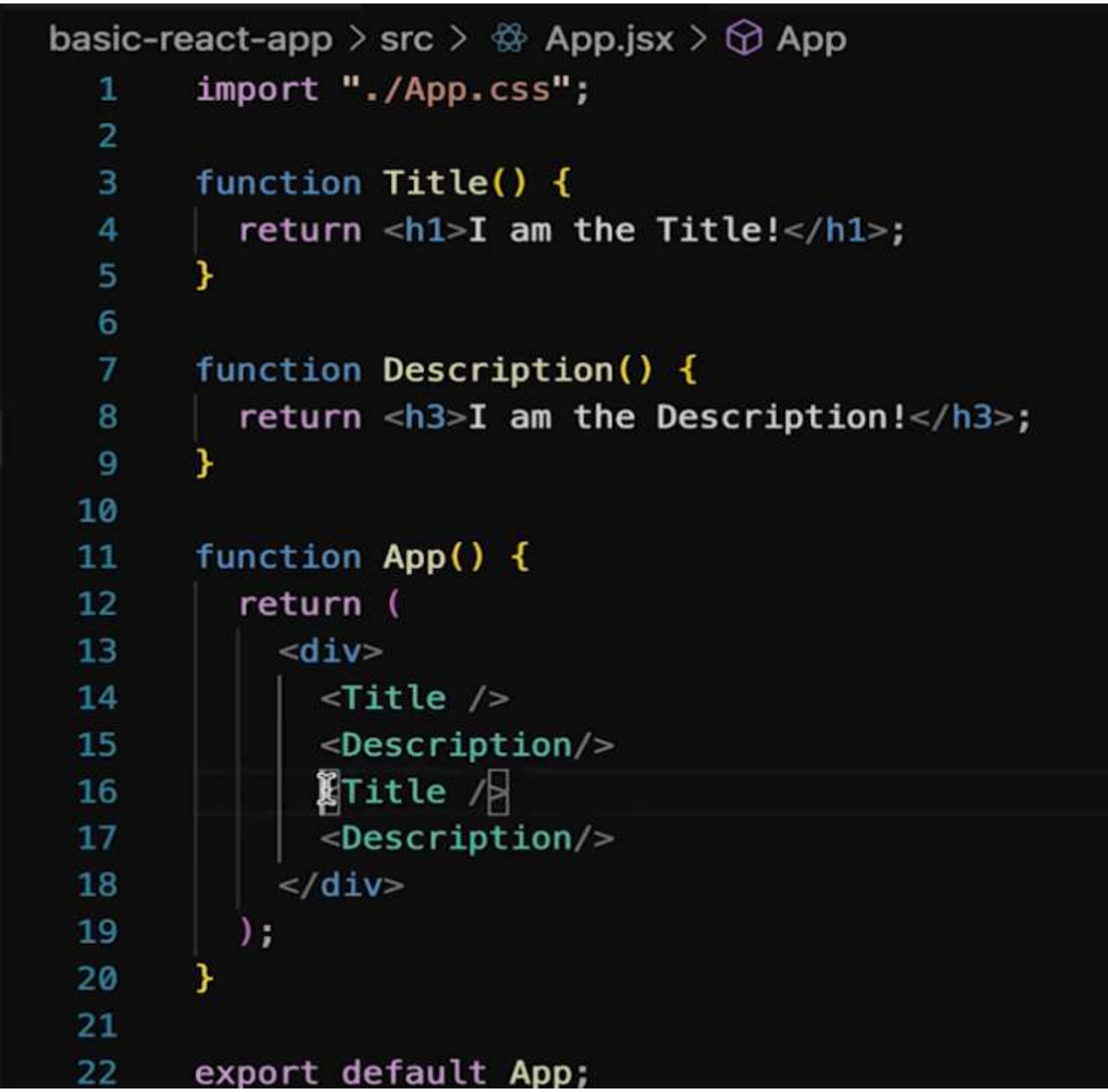
```
<Title/>
```


basic-react-app > src > App.jsx > App
1 import "./App.css";
2
3 function Title() {
4 return <h1>I am the Title!</h1>;
5 }
6
7 function App() {
8 return <Title />;
9 }
10
11 export default App;
12


I am the Title!

basic-react-app > src > App.jsx > App

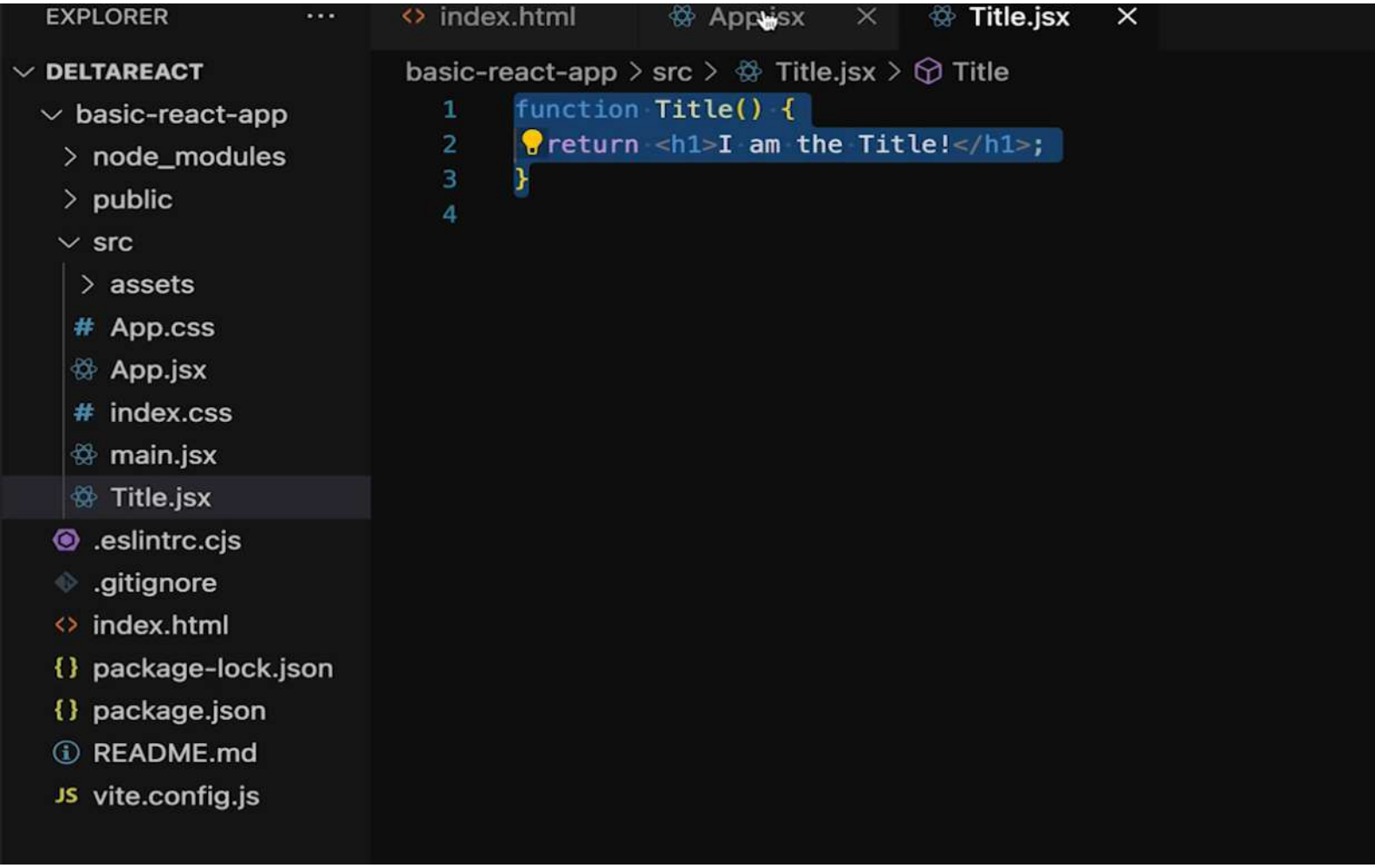
```jsx
import "./App.css";

function Title() {
  return <h1>I am the Title!</h1>;
}

function App() {
  return (
    <div>
      <h1>This is my app component</h1>
      <p>inside app component we have : </p>
      <Title />
    </div>
  );
}

export default App;

```


basic-react-app > src > App.jsx > App
import "./App.css";
function Title() {
return <h1>I am the Title!</h1>;
}
function Description() {
return <h3>I am the Description!</h3>;
}
function App() {
return (

<Title />
<Description/>
<Title />
<Description/>

);
}
export default App;


I am the Title!
I am the Description!
I am the Title!
I am the Description!


EXPLORER
index.html
App.jsx
Title.jsx
DELTAREACT
basic-react-app
node_modules
public
src
assets
App.css
App.jsx
index.css
main.jsx
Title.jsx
.eslintrc.cjs
.gitignore
index.html
package-lock.json
package.json
README.md
vite.config.js
basic-react-app > src > Title.jsx > Title
function Title() {
return <h1>I am the Title!</h1>;
}

Import-Export

Import

```
import Title from "./Title.jsx";
```

Default Export

```
export default Title;
```

Named Export

```
export { Title };
```

```
import { Title } from "./Title.jsx";
```

In summary, named exports are useful when you want to export multiple values and import them with their specific names, while default exports are handy for exporting a single value and giving it a custom name when importing. The choice between the two depends on the structure and requirements of your codebase.


index.html
App.jsx
Title.jsx
basic-react-app > src > Title.jsx > Title
1 function Title() {
2 return <h1>I am the Title!</h1>;
3 }
4
5 // export default Title;
6 export { Title };
7

basic-react-app > src > App.jsx > ...

```jsx
import "./App.css";
import {Title} from "./Title.jsx";

function Description() {
  return <h3>I am the Description!</h3>;
}

function App() {
  return (
    <div>
      <Title />
      <Description />
      <Title />
      <Description />
    </div>
  );
}

export default App;
```


I am the Title!
I am the Description!
I am the Title!
I am the Description!

Writing Markup in JSX

1. Return a single root element
2. Close all the tags
3. camelCase most of the things

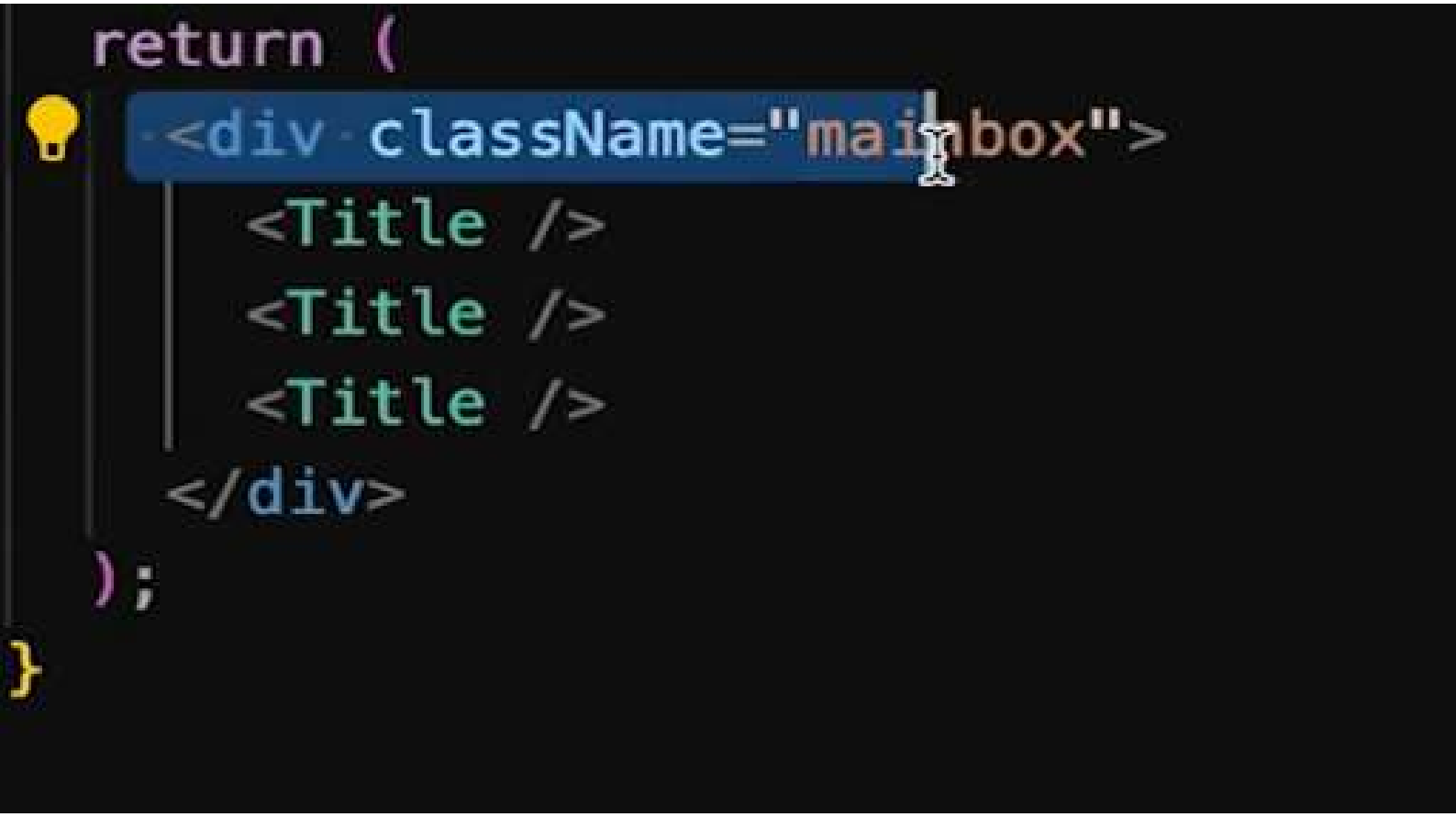
3. camelCase ~~all~~ most of the things!

JSX turns into JavaScript and attributes written in JSX become keys of JavaScript objects. In your own components, you will often want to read those attributes into variables. But JavaScript has limitations on variable names. For example, their names can't contain dashes or be reserved words like `class`.

This is why, in React, many HTML and SVG attributes are written in camelCase. For example, instead of `stroke-width` you use `strokeWidth`. Since `class` is a reserved word, in React you write `className` instead, named after the corresponding DOM property:

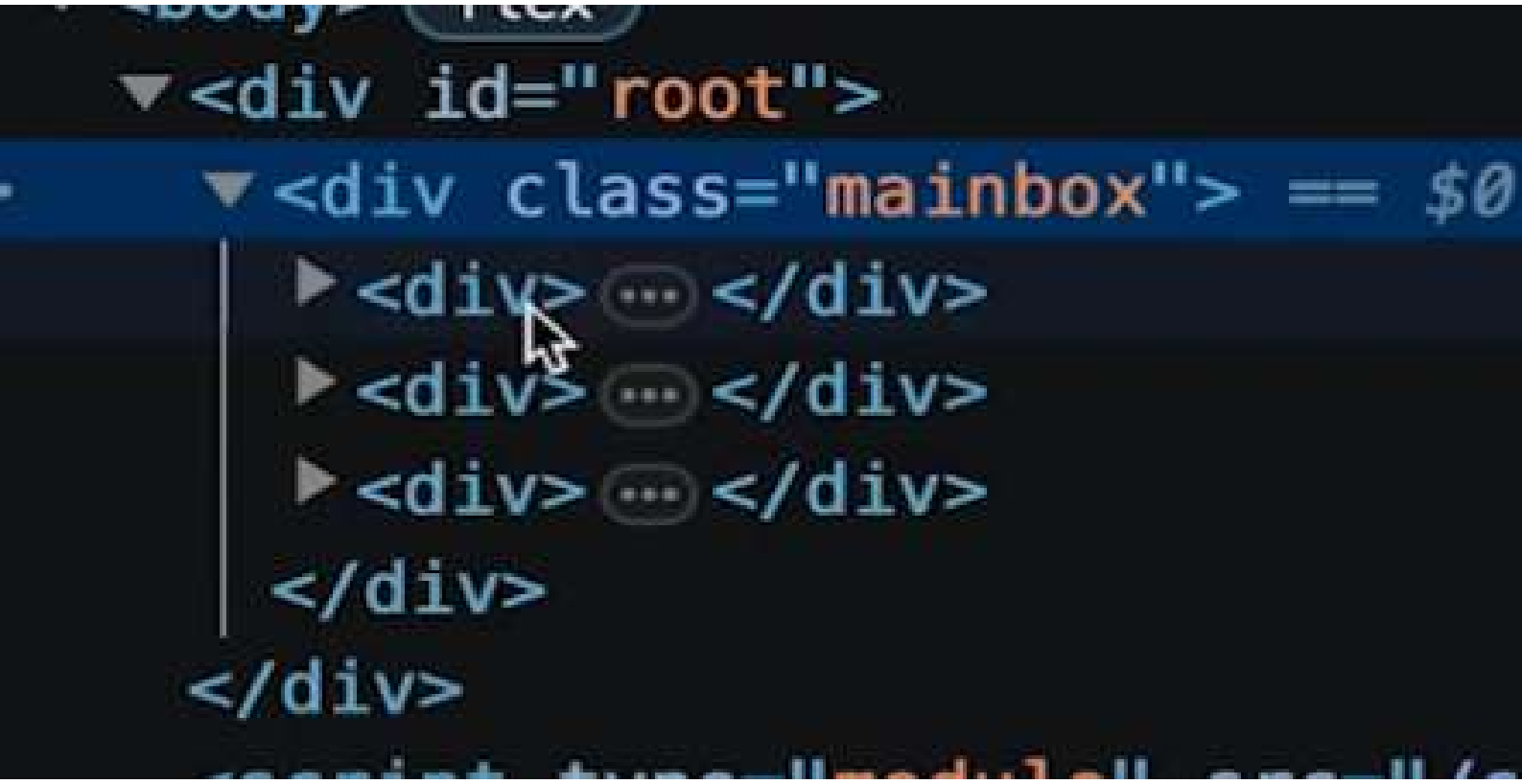
React Fragment

Fragments let you group a list of children without adding extra nodes to the DOM.


return (

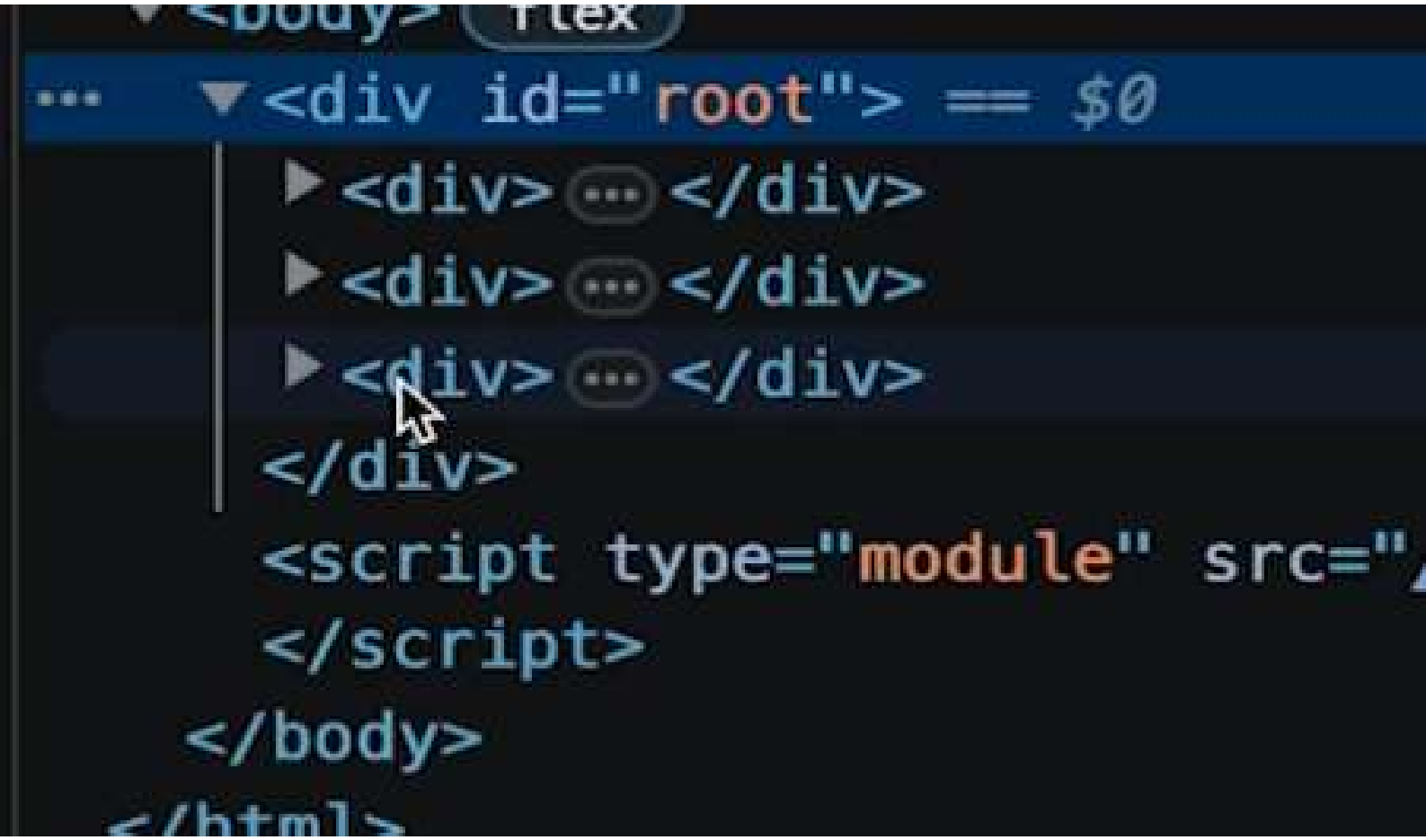
<Title />
<Title />
<Title />

);
}



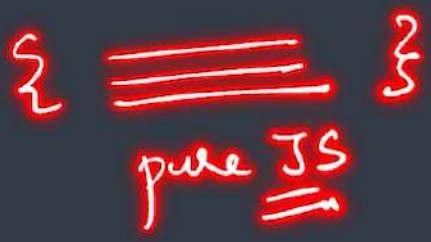
 == $0
 ...
 ...
 ...


function App() {
return (
<>
<Title />
<Title />
<Title />
</>
);
}


<body> flex
 == $0
 ...
 ...
 ...

<script type="module" src="
</script>
</body>
</html>

JSX with Curly Braces




```
function Title() {
  return <p>2 * 2 = {2 * 2}</p>;
}
```

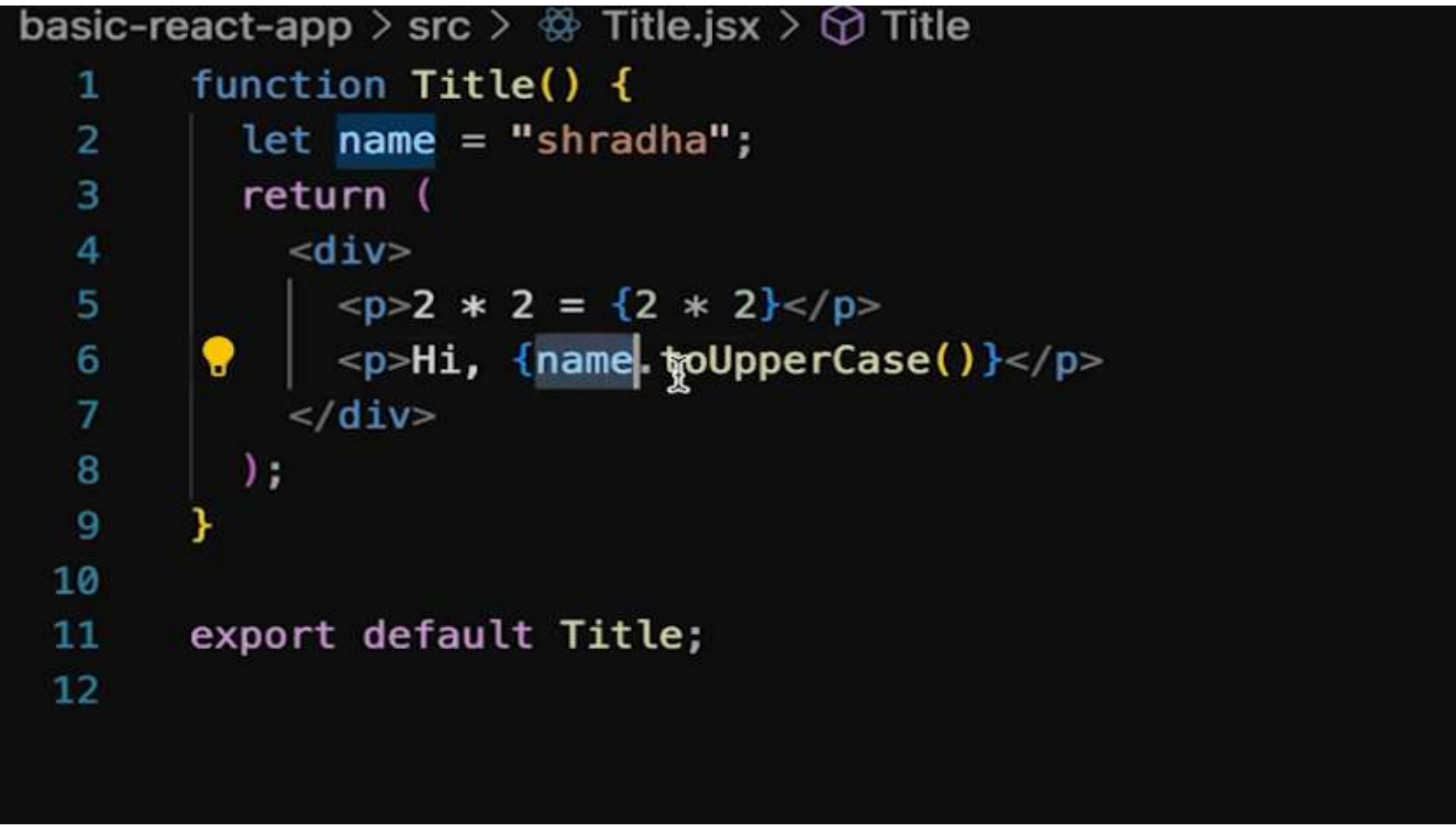
```
function Title() {
  let msg = "hello world!";
  return <p>message says {msg}</p>;
}
```

JavaScript in JSX with Curly Braces

JSX lets you write HTML-like markup inside a JavaScript file, keeping rendering logic and content in the same place. Sometimes you will want to add a little JavaScript logic or reference a dynamic property inside that markup. In this situation, you can use curly braces in your JSX to open a window to JavaScript.


index.html
App.jsx
Title.jsx
basic-react-app > src > Title.jsx > Title
1 function Title() {
2 return (
3
4 2 * 2 = {2 * 2}
5
6);
7 }
8
9 export default Title;
10


basic-react-app > src > Title.jsx > Title
1 function Title() {
2 let name = "shradha";
3 return (
4
5 2 * 2 = {2 * 2}
6 Hi, {name}
7
8);
9 }
10
11 export default Title;
12

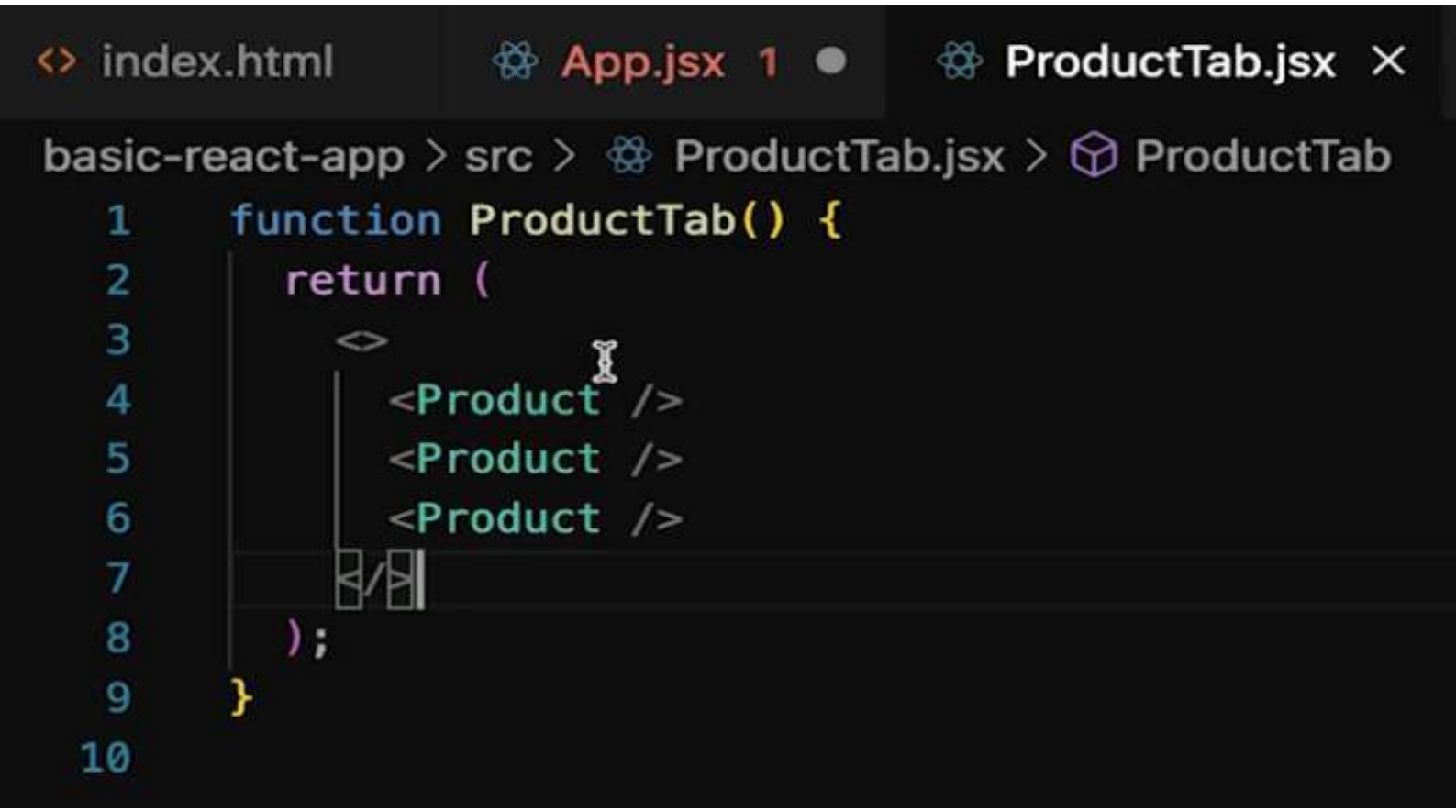

basic-react-app > src > Title.jsx > Title
1 function Title() {
2 let name = "shradha";
3 return (
4
5 2 * 2 = {2 * 2}
6 Hi, {name.toUpperCase()}
7
8);
9 }
10
11 export default Title;
12

Structuring Components

Jira Software
Project and issue tracking

Jira Product Discovery
Prioritization and roadmapping

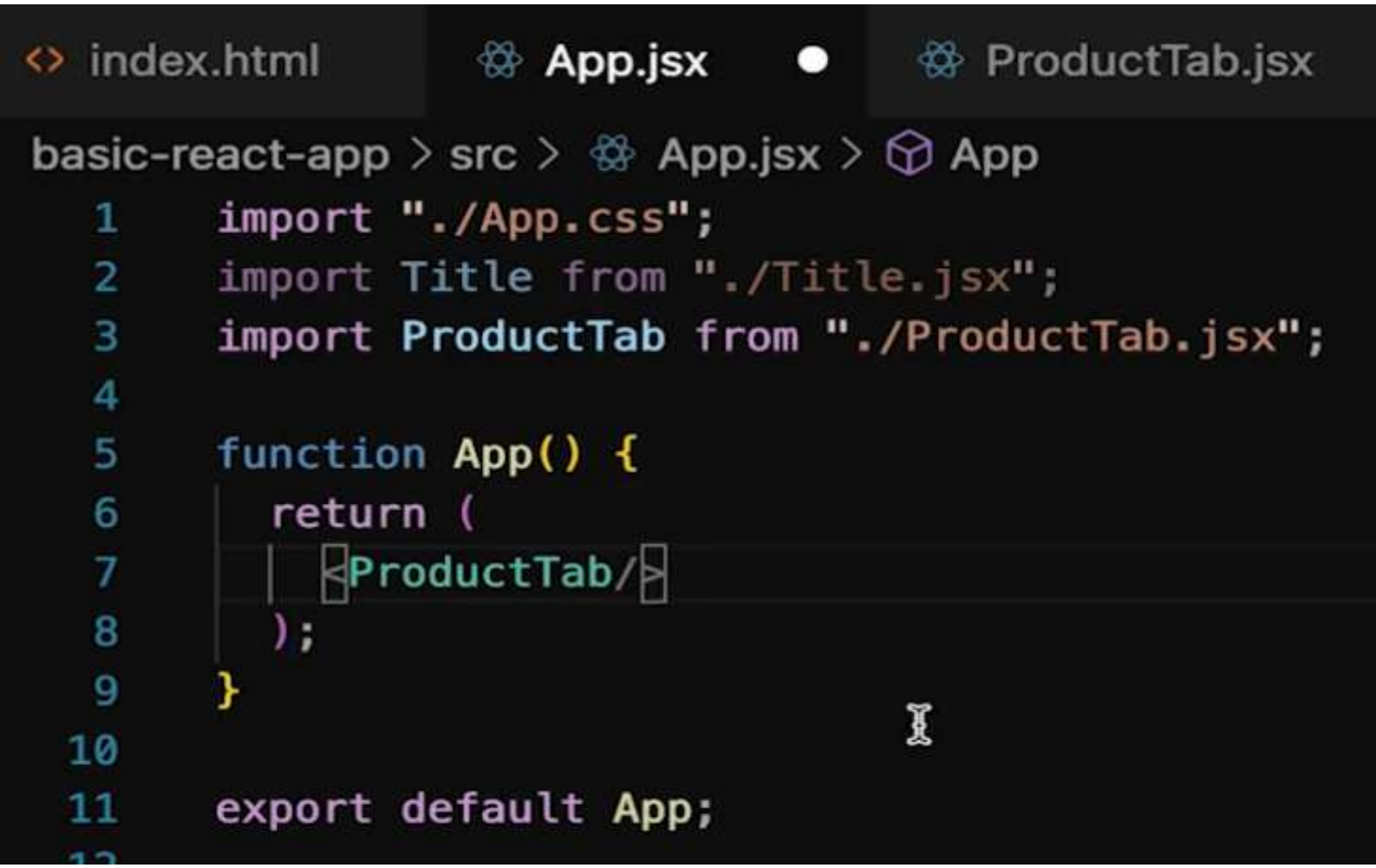
Jira Align
Enterprise Agile planning


index.html
App.jsx 1
ProductTab.jsx
basic-react-app > src > ProductTab.jsx > ProductTab
1 function ProductTab() {
2 return (
3 <>
4 <Product />
5 <Product />
6 <Product />
7 </>
8);
9 }
10

index.html | App.jsx 1 | ProductTab.jsx

basic-react-app > src > ProductTab.jsx > default

```jsx
import Product from "./Product.jsx";

function ProductTab() {
  return (
    <>
      <Product />
      <Product />
      <Product />
    </>
  );
}

export default ProductTab;
```


index.html
App.jsx
ProductTab.jsx
basic-react-app > src > App.jsx > App
1 import "./App.css";
2 import Title from "./Title.jsx";
3 import ProductTab from "./ProductTab.jsx";
4
5 function App() {
6 return (
7 <ProductTab/>
8);
9 }
10
11 export default App;

index.html | App.jsx | ProductTab.jsx | Product.jsx

basic-react-app > src > Product.jsx > ...

```jsx
function Product() {
  return (
    <div>
      <h3>Product Title</h3>
      <h5>Product Description</h5>
    </div>
  );
}

export default Product;
```

Style Components

```
import "./Product.css";

export default function Product() {
  return (
    <div className="Product">
      <h4>Product Title</h4>
      <p>Product Description</p>
    </div>
  );
}
```

```
.Product {
  border: 1px solid white;
  margin-bottom: 5px;
  border-radius: 14px;
}
```

App

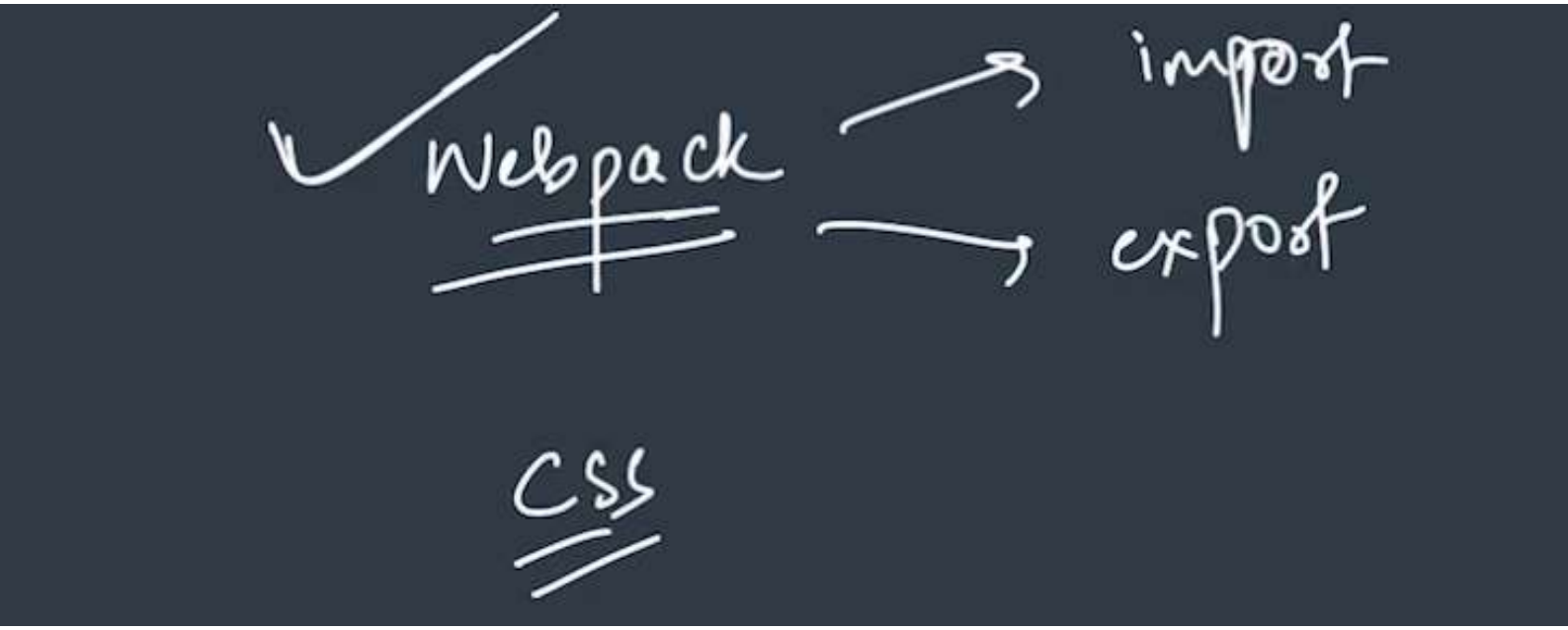
App.css

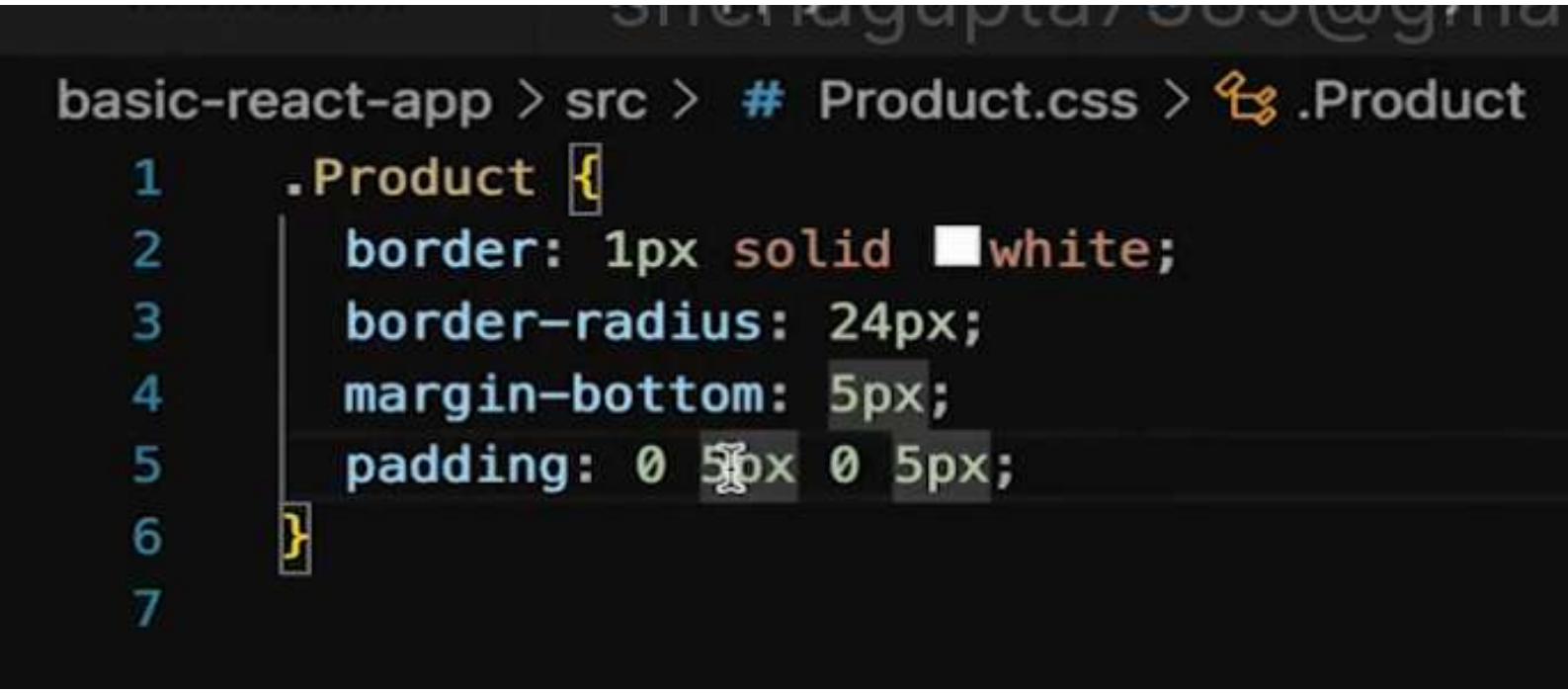
Product

Product.css

ProductTab

ProductTab.css


Webpack
import
export
CSS


basic-react-app > src > # Product.css > .Product
1 .Product {
2 border: 1px solid white;
3 border-radius: 24px;
4 margin-bottom: 5px;
5 padding: 0 5px 0 5px;
6 }
7

Product Title

Product Description

Product Title

Product Description

Product Title

Product Description