React.JS

(·) You Should Already know

- ↳ JS basic
- ↳ Basic Command-line Interface.
- ↳ React basic.

(·) Installing Node.JS & NPM

- ↳ check if it is installed or not
  - ↳ Open terminal
    - ↳ node -v or node --version
    - ↳ npm -v or npm --version

(·) why React

(i) It makes creating a website much simpler & faster without taking away any of the fine grained control from the programmer.

(ii) It allows us to create modular reusable components that can be arranged into a fully functional website instead of hand coding each individual page in our site.

React.TS

⊙ Project set-up

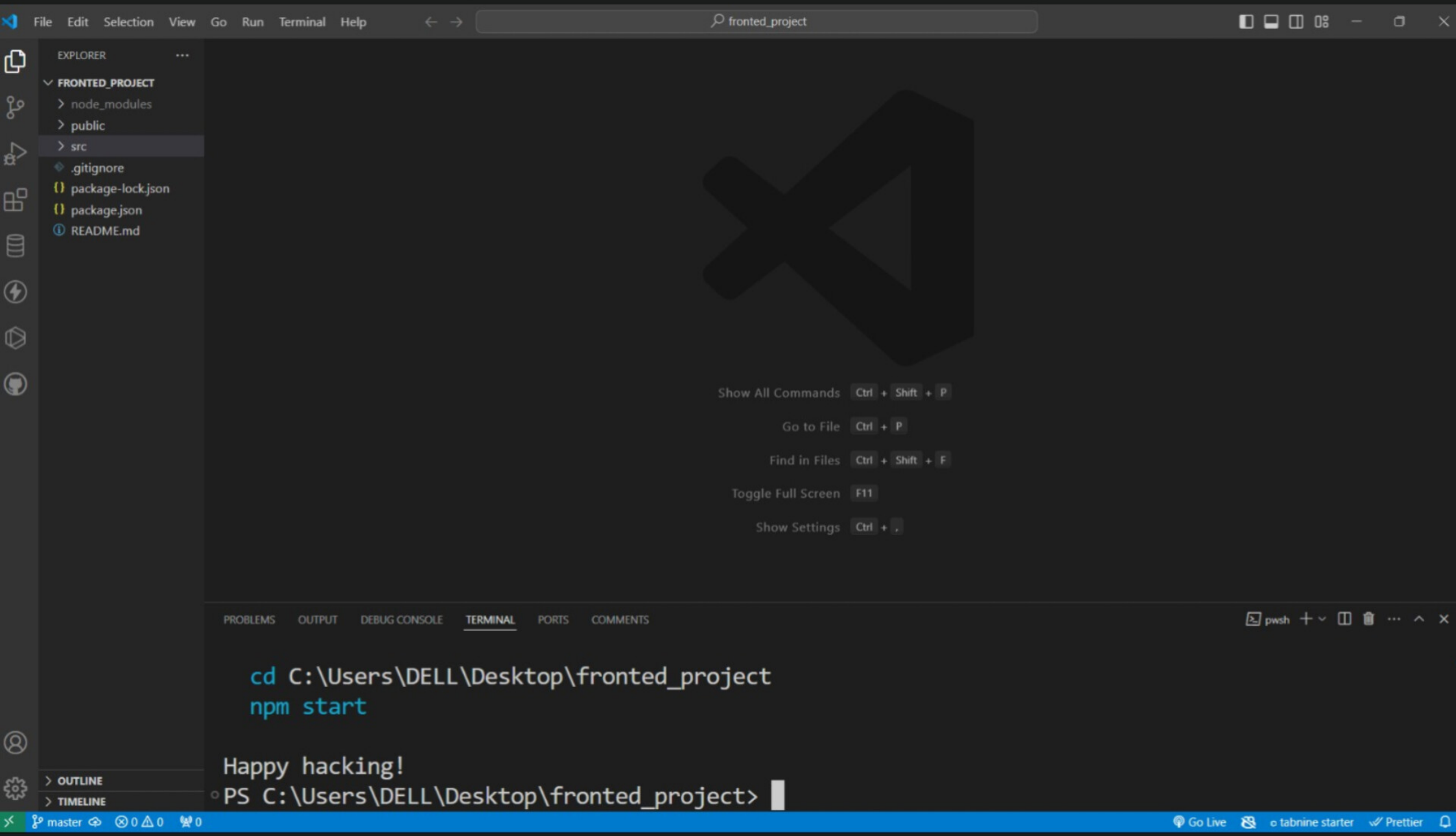
↳ Create directory

↳ npx create-react-app .

⊙ To run react app

↳ npm start

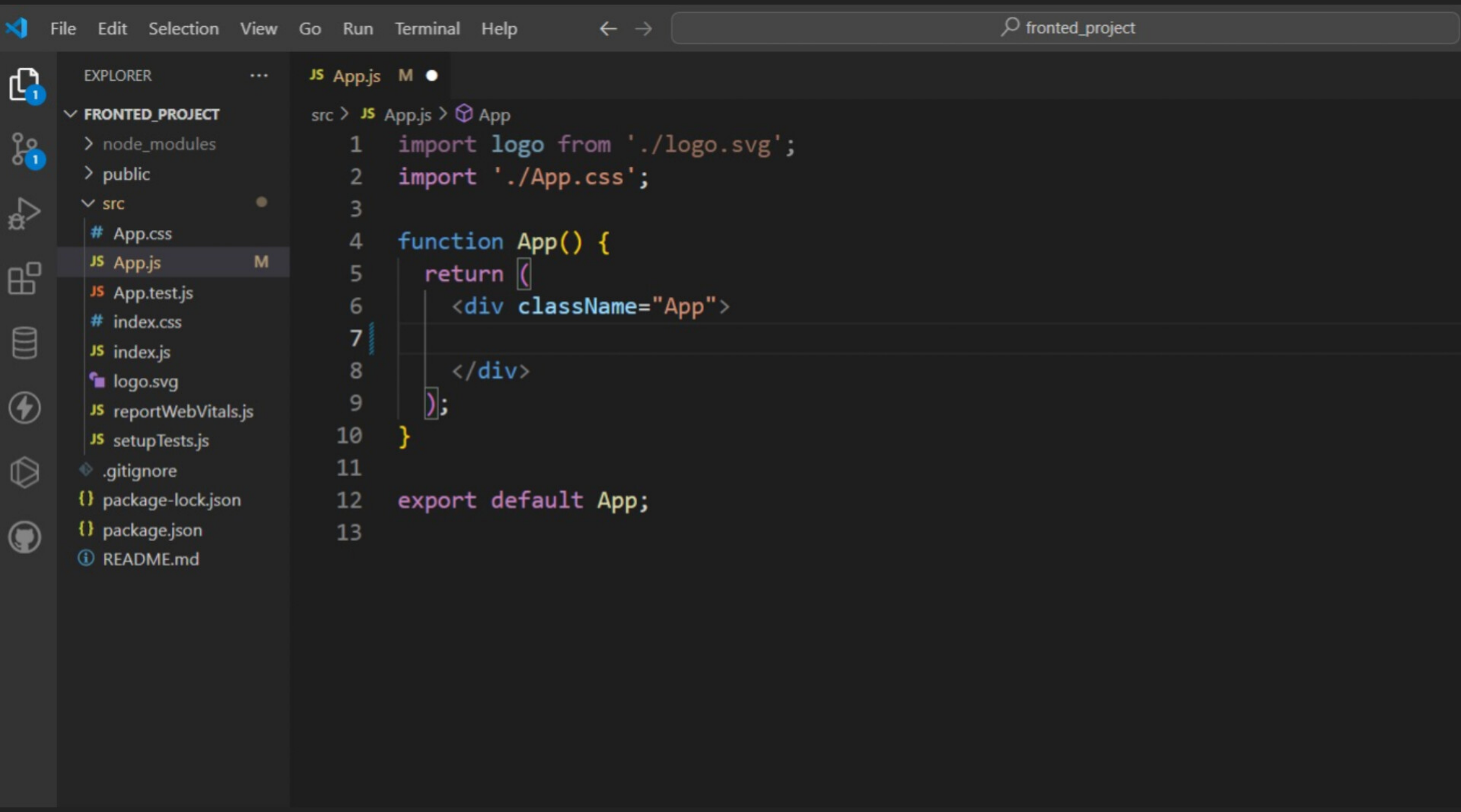
Preject structure of React.



```
  cd C:\Users\DELL\Desktop\fronted_project
  npm start

Happy hacking!
PS C:\Users\DELL\Desktop\fronted_project>
```

Start building site using src directory

App.jsx is the root component of react



```
import logo from './logo.svg';
import './App.css';

function App() {
  return (
    <div className="App">

    </div>
  );
}

export default App;
```

We can control other components through App.jsx

React.JS

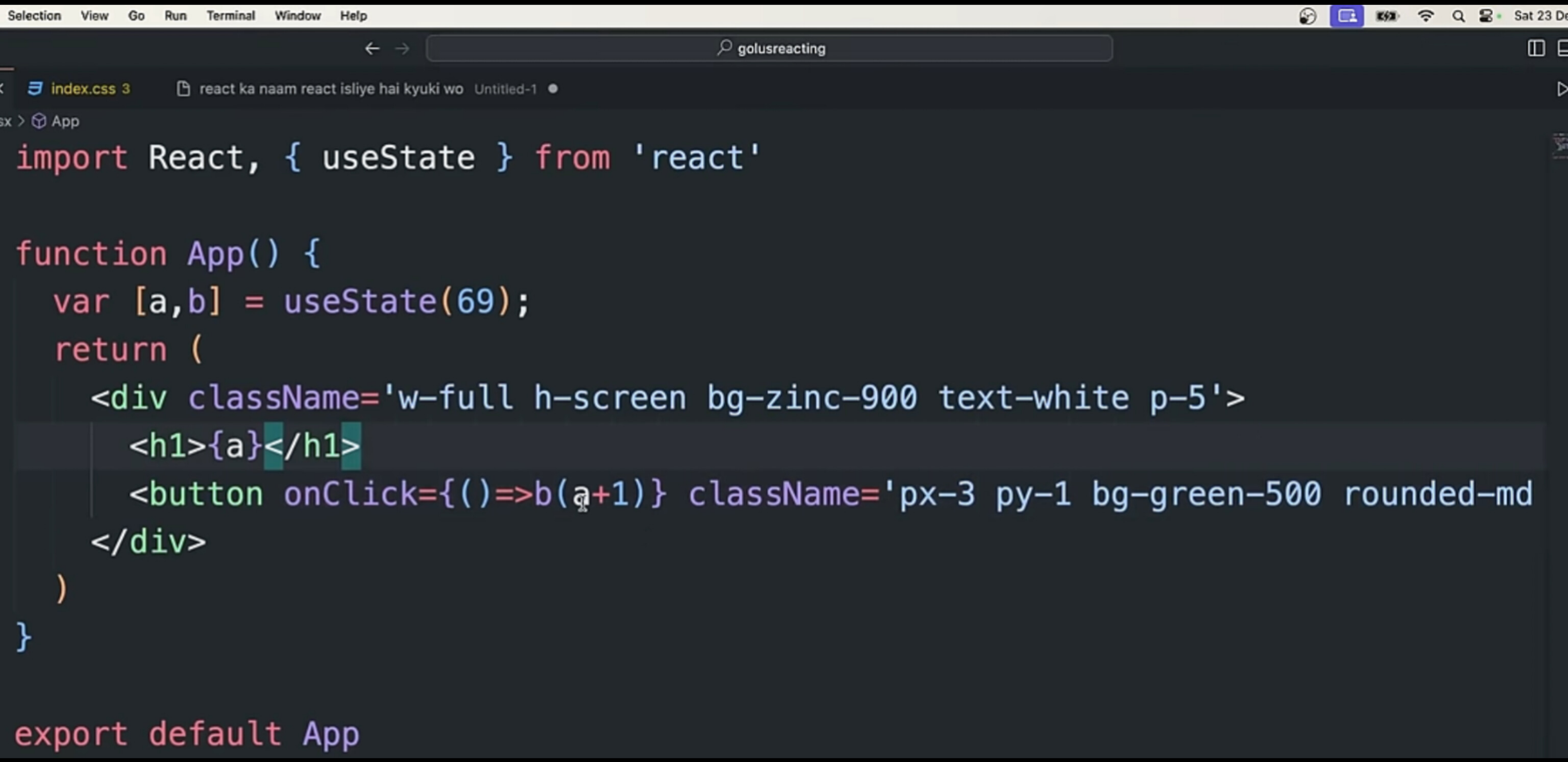
(i) Explanation about project directory:-

- → package.json:- It Contains info. about project.
- → package.lock.json:- It Contains all of the dependencies of our project.
- → public:- It is default directory created by react which Contains index.html and so on.
- → src:- i) Inside src directory you can see App.js.

ii) App.js is a root Component for our blog site. This is basically the component in React that is going to Contain the rest of the components that we build.

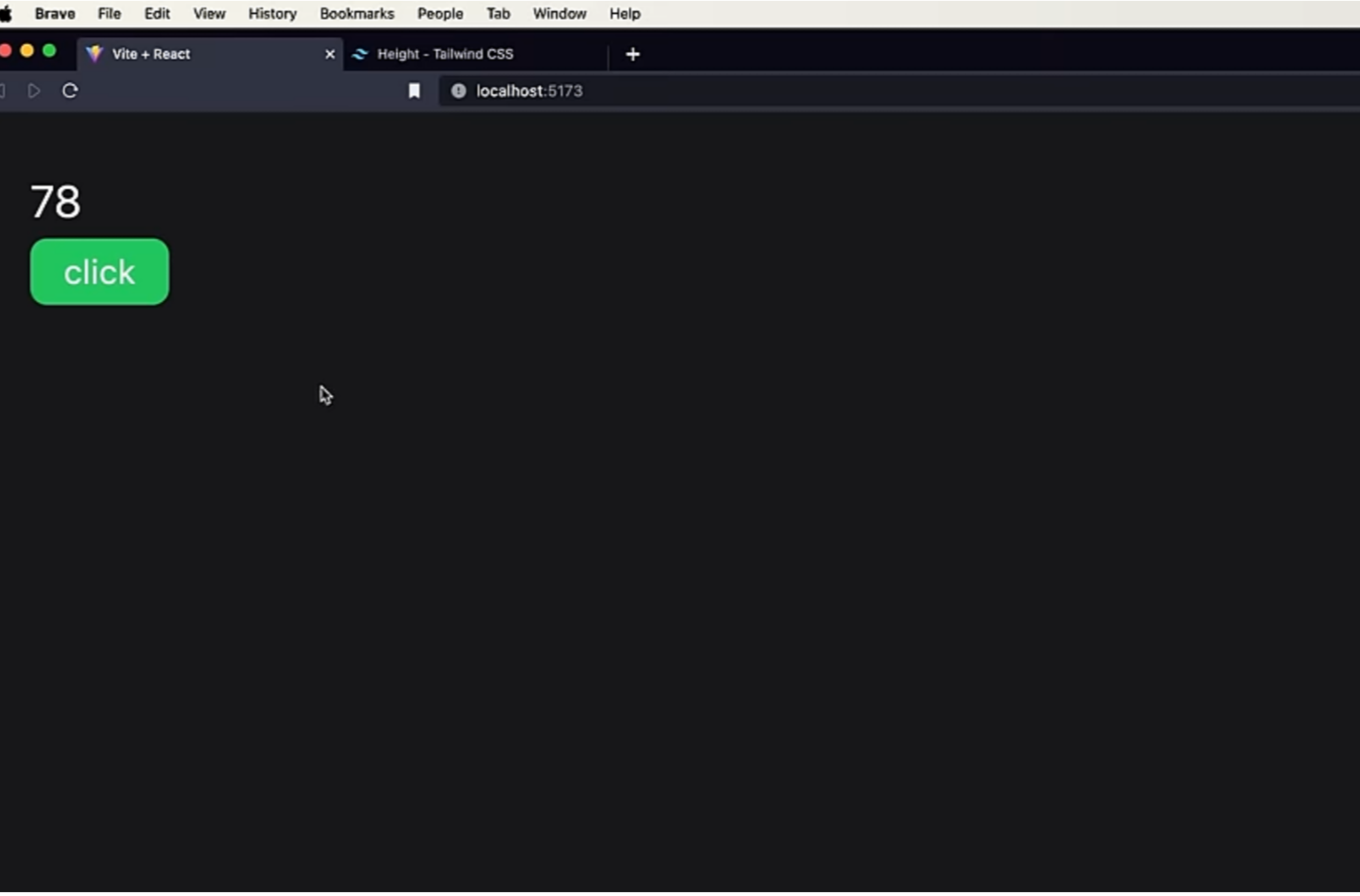
iii) And we have many other files inside the src directory like App.css, App.test.js etc.

How to change state in React



```
import React, { useState } from 'react'

function App() {
  var [a,b] = useState(69);
  return (
    <div className='w-full h-screen bg-zinc-900 text-white p-5'>
      <h1>{a}</h1>
      <button onClick={()=>b(a+1)} className='px-3 py-1 bg-green-500 rounded-md
    </div>
  )
}

export default App
```

'a' pointing value and 'b' is the updater

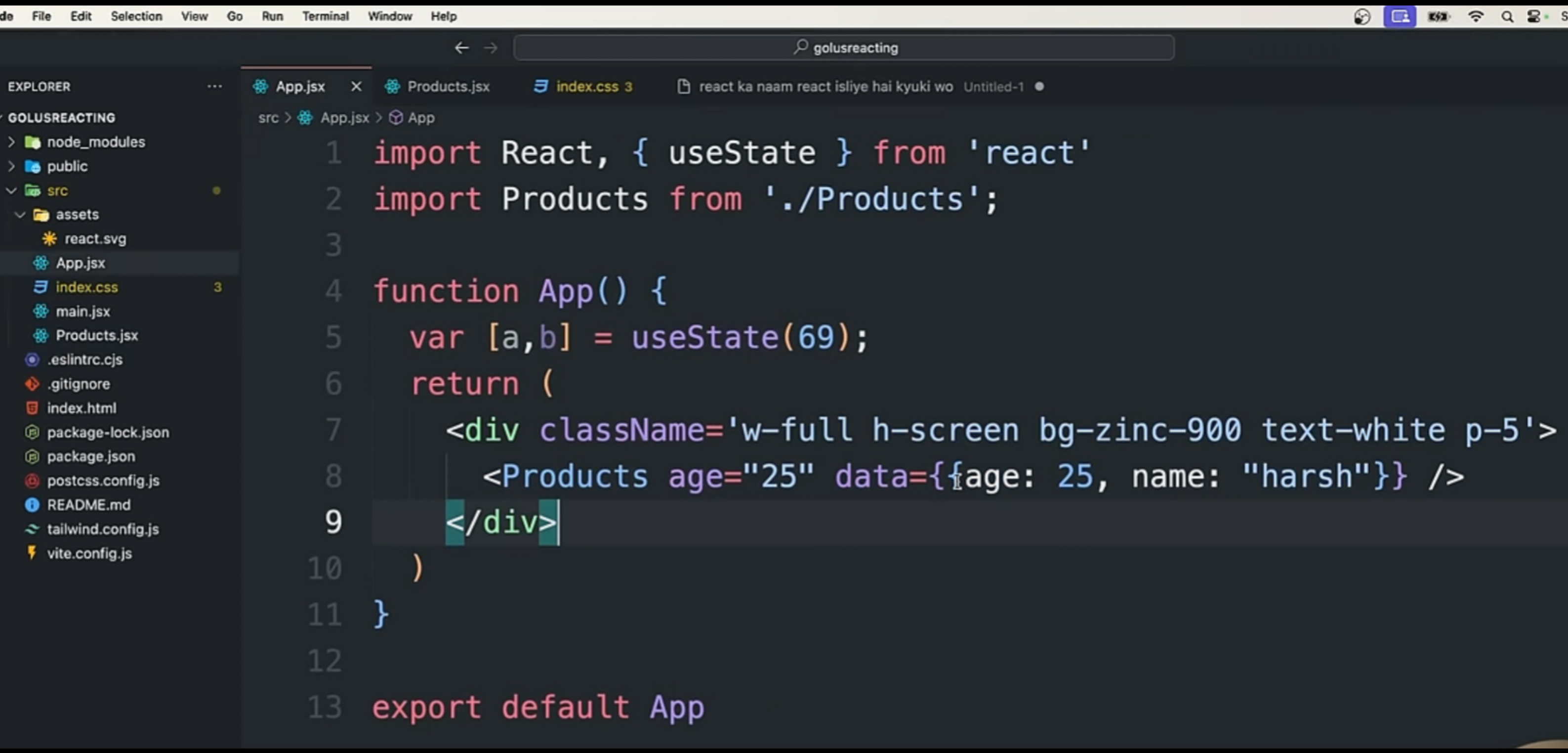


Brave File Edit View History Bookmarks People Tab Window Help
Vite + React
Height - Tailwind CSS
localhost:5173

78

click

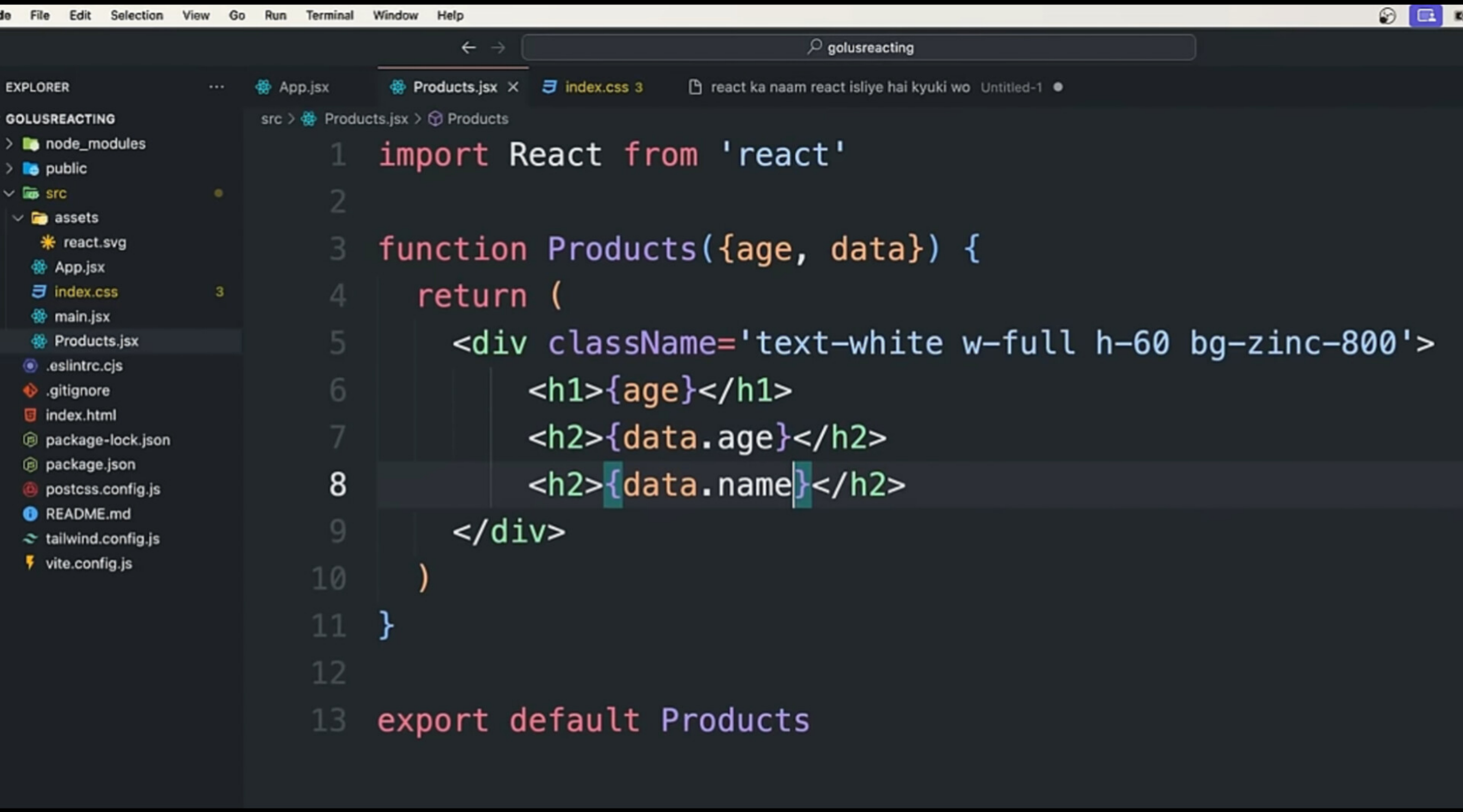
Sending data one component to other component.



```jsx
import React, { useState } from 'react'
import Products from './Products';

function App() {
  var [a,b] = useState(69);
  return (
    <div className='w-full h-screen bg-zinc-900 text-white p-5'>
      <Products age="25" data={{age: 25, name: "harsh"}} />
    </div>
  )
}

export default App
```

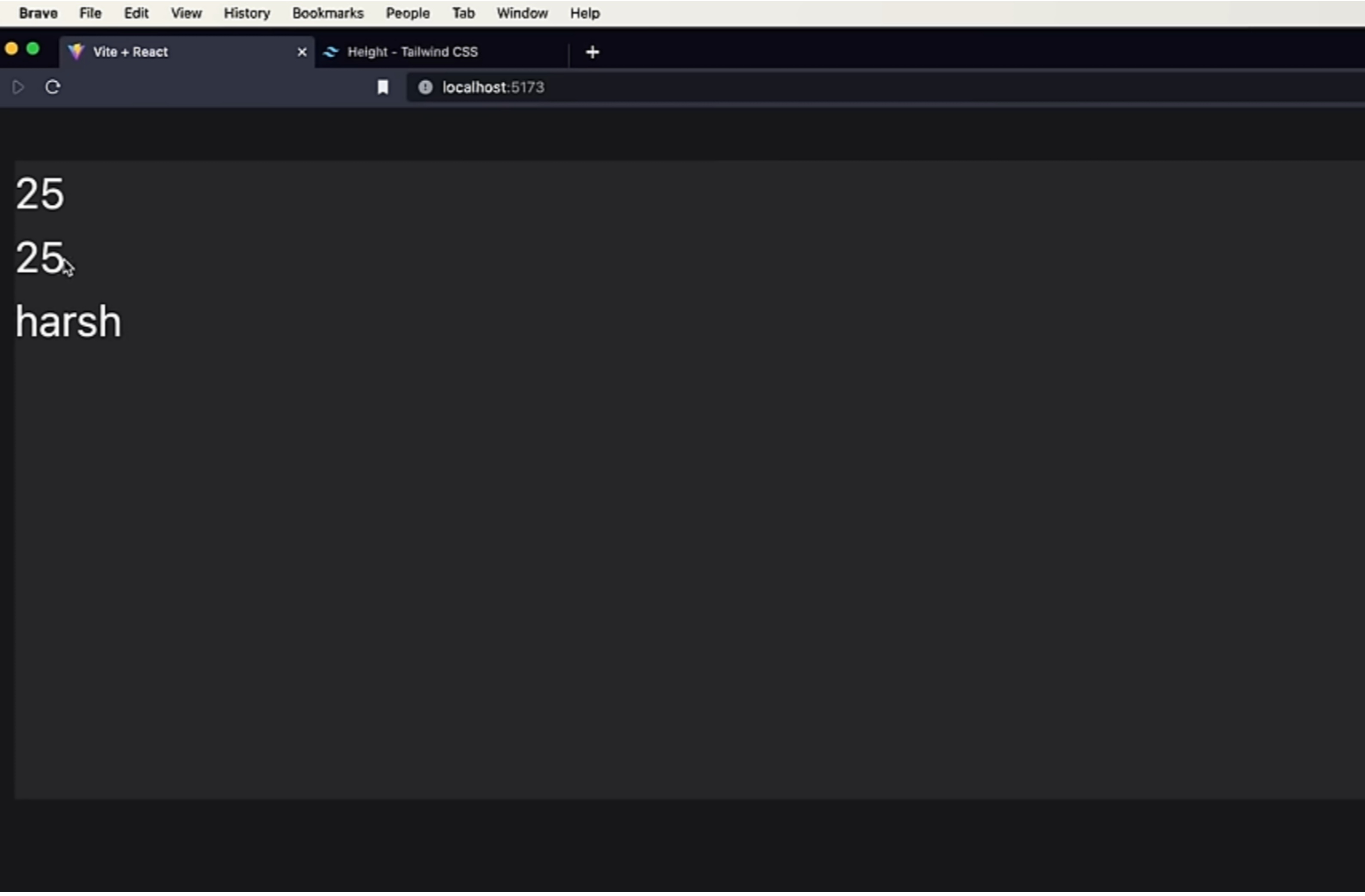
👇👇👇👇

Receiving data from app.jsx component.



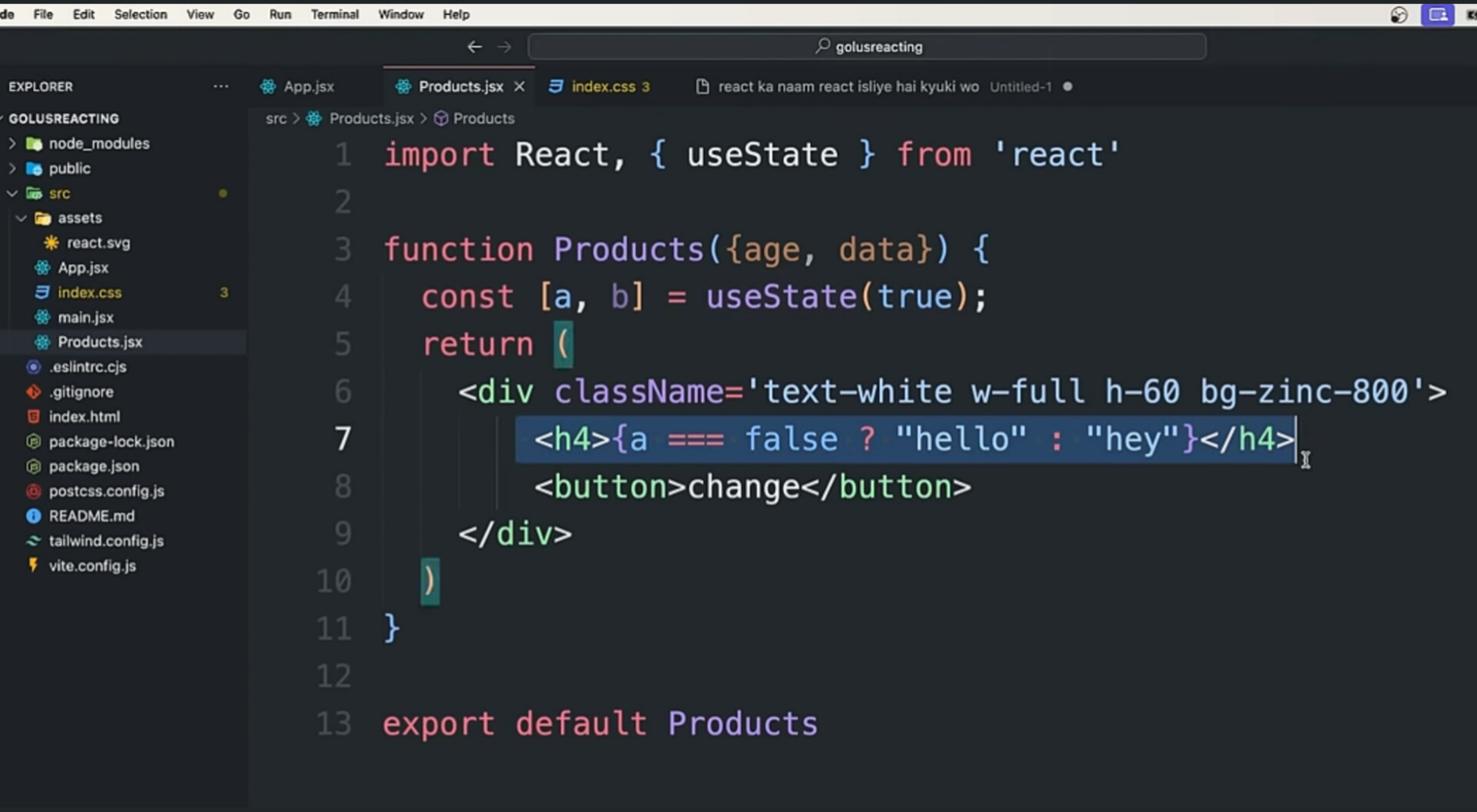
```jsx
import React from 'react'

function Products({age, data}) {
  return (
    <div className='text-white w-full h-60 bg-zinc-800'>
        <h1>{age}</h1>
        <h2>{data.age}</h2>
        <h2>{data.name}</h2>
    </div>
  )
}

export default Products
```

Output 👇


Brave
File
Edit
View
History
Bookmarks
People
Tab
Window
Help
Vite + React
Height - Tailwind CSS
localhost:5173

```
25
25
harsh
```

Creating data using useState.



```jsx
import React, { useState } from 'react'

function Products({age, data}) {
  const [a, b] = useState(true);
  return (
    <div className='text-white w-full h-60 bg-zinc-800'>
      <h4>{a === false ? "hello" : "hey"}</h4>
      <button>change</button>
    </div>
  )
}

export default Products
```

We can create all types of data using useState.