**Day 54**





**“Web Development + Security”**

**Components, Props and JSX in React:**

**What Are Components?**

A component in React is a reusable, independent piece of UI. Think of components as building blocks — like Lego pieces of your website.

**Why Components?**

* Reusable
* Organized
* Easier to maintain
* Scalable

**Types of Components**

| **Type** | **Syntax** | **Example** |
| --- | --- | --- |
| **Functional Component** | Uses function + hooks | function App() { return <h1>Hi</h1>; } |
| **Class Component** | Uses class + lifecycle methods (old way) | class App extends React.Component { render() { return <h1>Hi</h1>; } } |

**What Are Props (Properties)?**

Props are like *function parameters* — they let you pass data from a parent component to a child component.

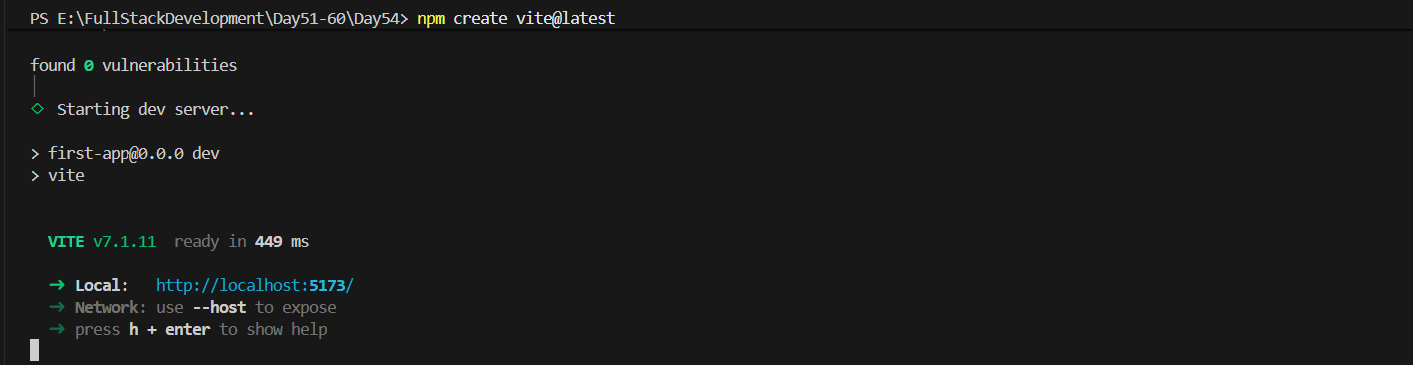
**What Is JSX? (JavaScript XML)**

JSX is a syntax extension for JavaScript that looks like HTML, but is actually converted into React elements behind the scenes.

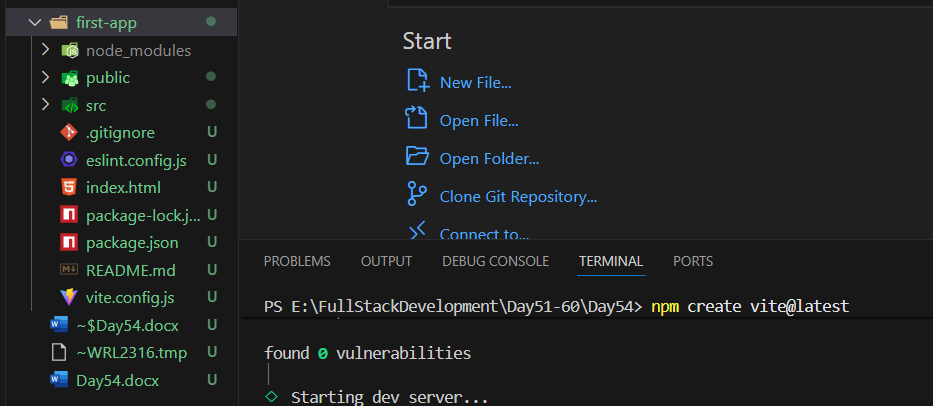
**In Short**

* Components → building blocks of UI.
* Props → pass data between components.
* JSX → HTML-like syntax to write UI inside JS.
* Together, they make React reusable, readable, and dynamic.

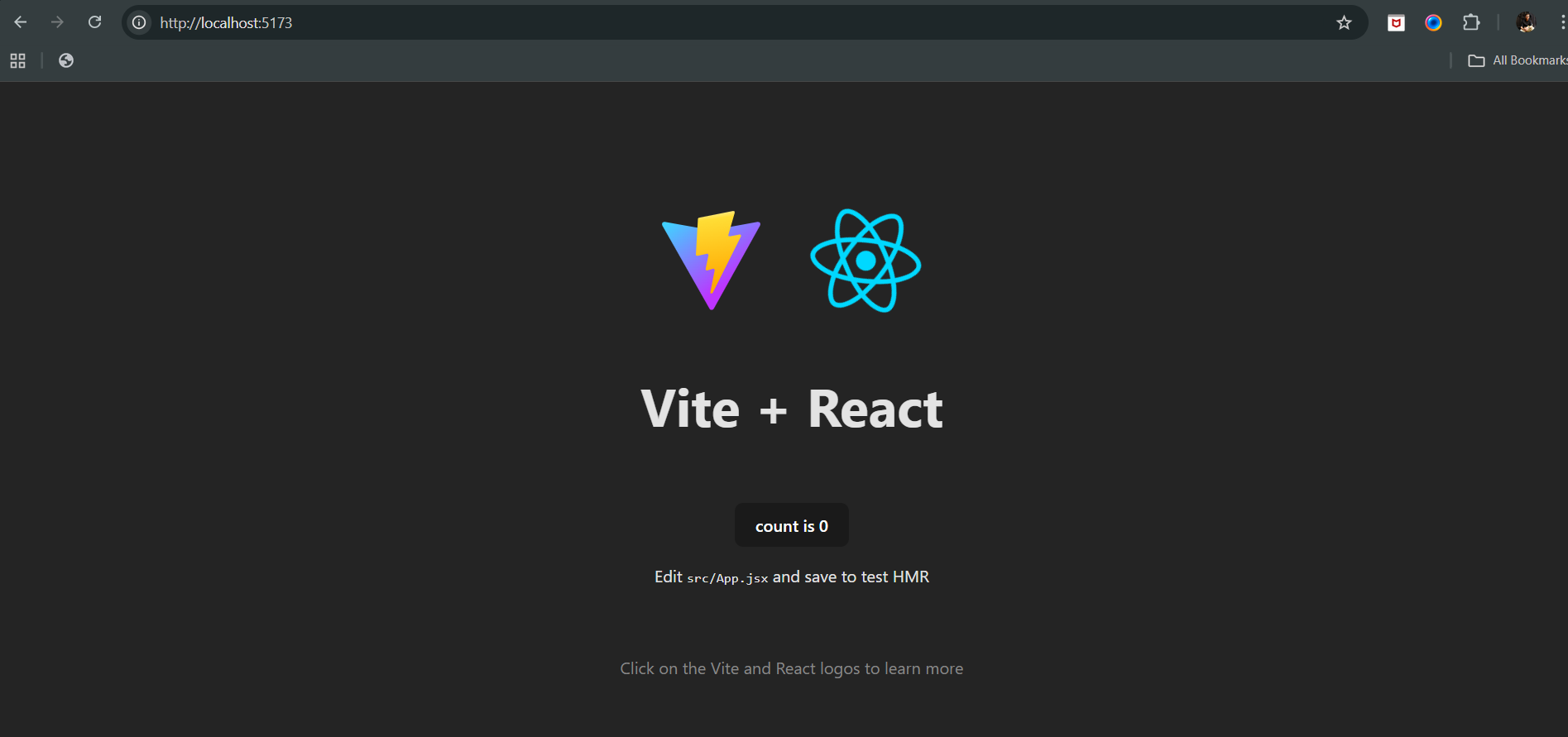
Now, to understand this we will start to use, Vite: to install



When we run this, it will automatically creates a folder named: first-app

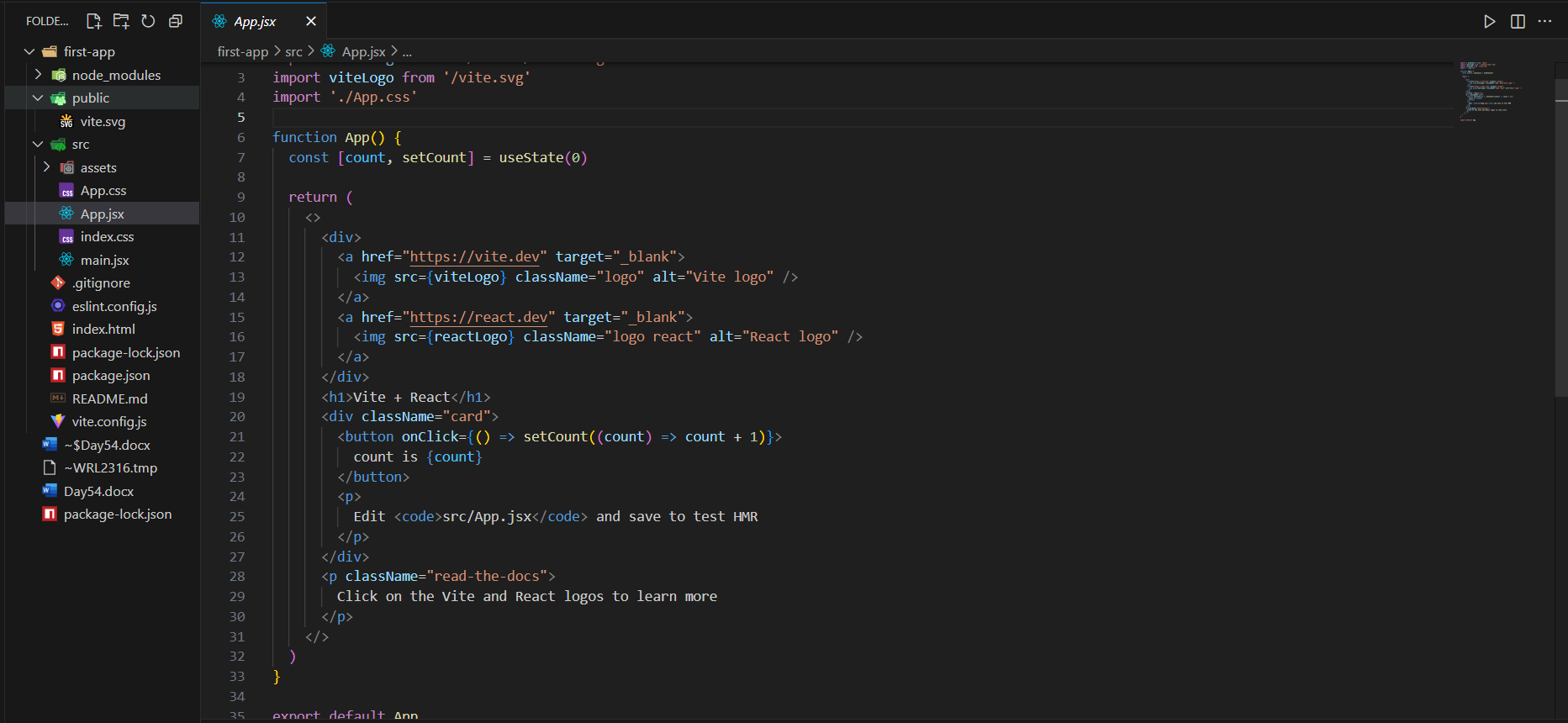


Since, we had started it, it will give output as:

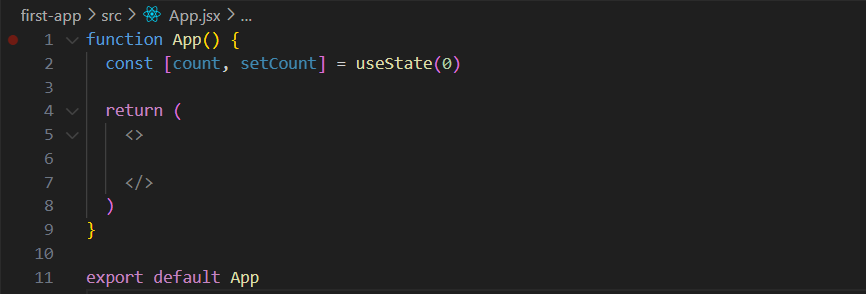


Now, since we had created a counter on last day, we can see that it is there on the screen too. So, now we will edit App.jsx and remove that part:

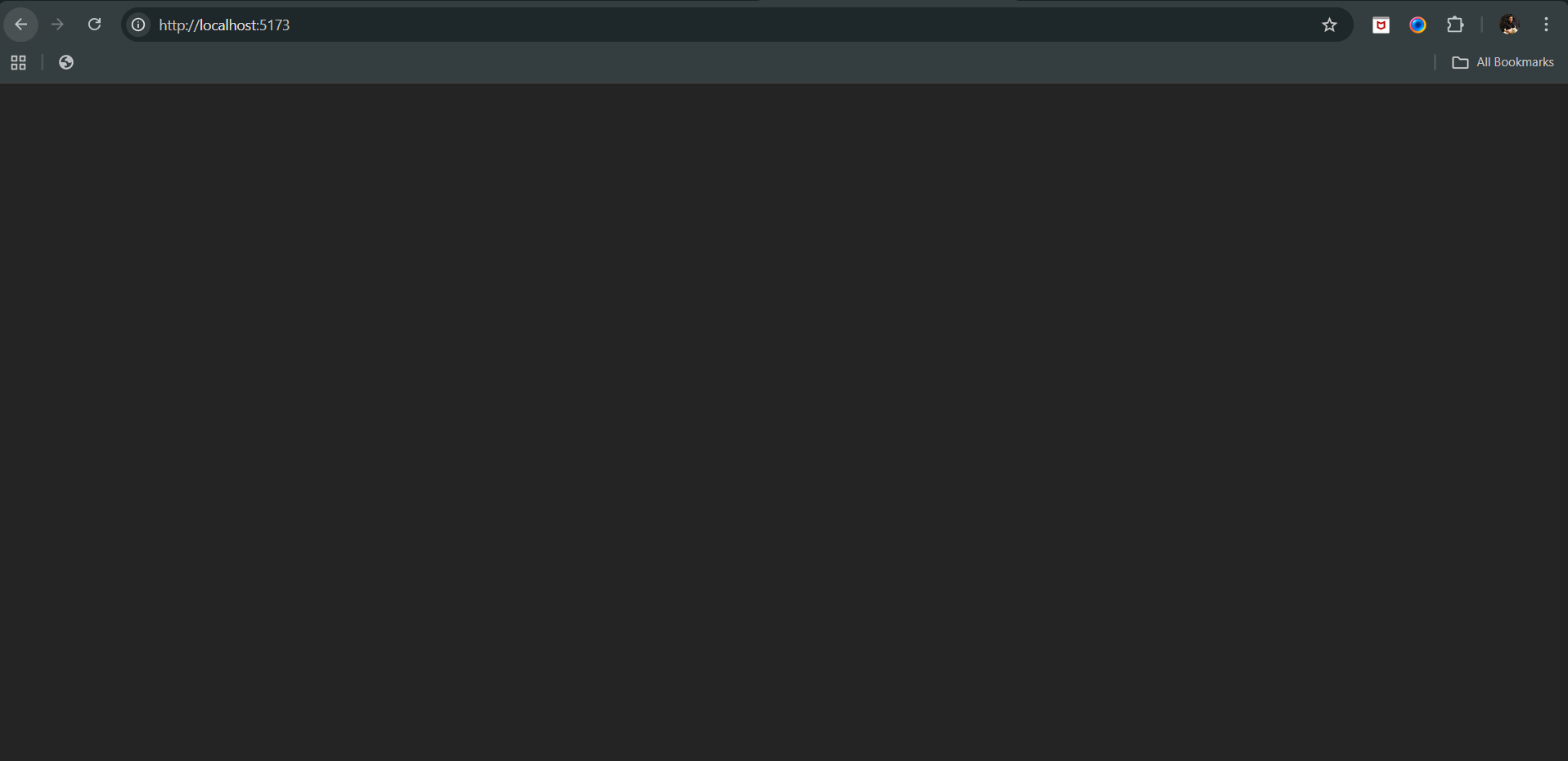
Old: App.jsx



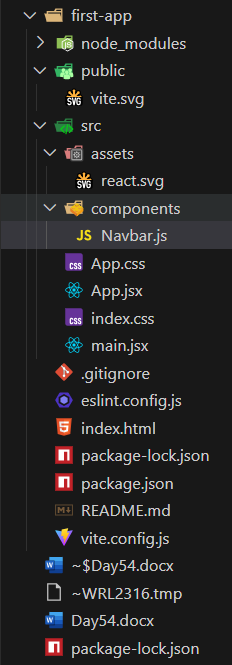
New: App.jsx



Now, as we changed the App.jsx. our output will change as well: it is blank as App.jsx is blank



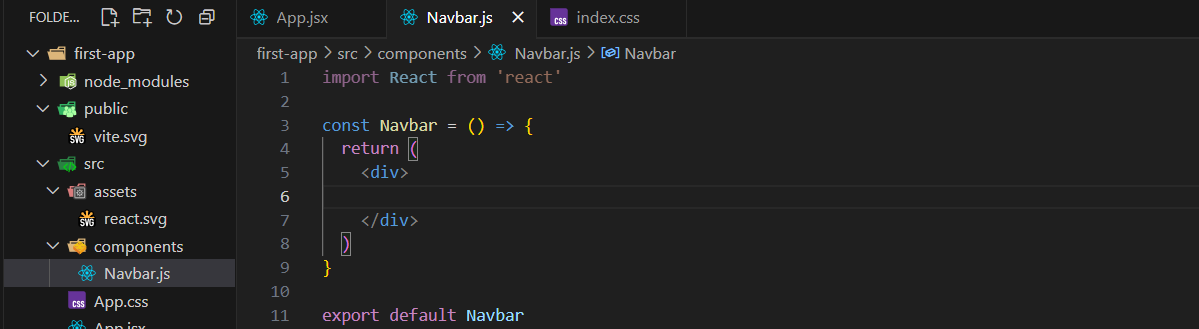
Now, we will create a nav bar, for which we will first create a folder inside the “src”:



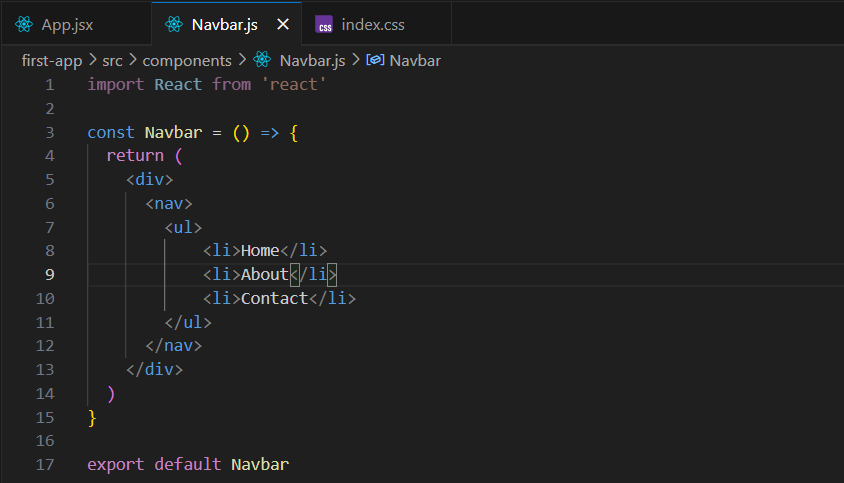
Inside the Navbar.js, we will write this template:



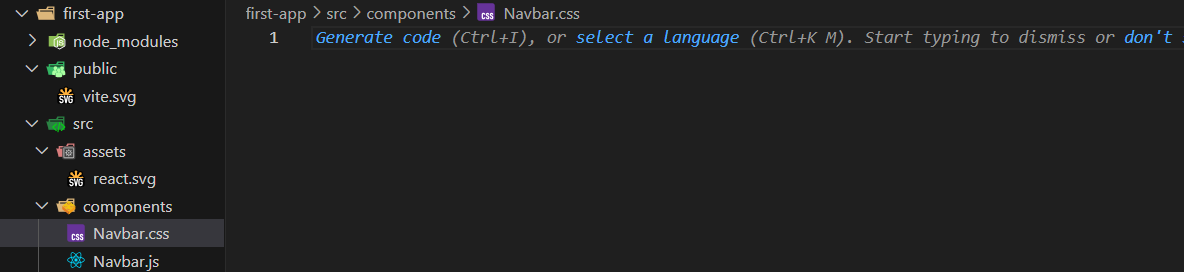
Change it to jsx:



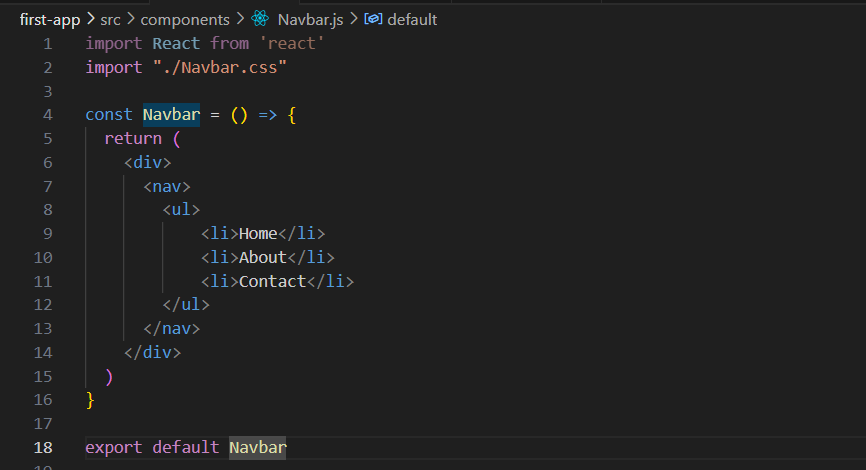
Now, we give the basic layout of the Navbar:



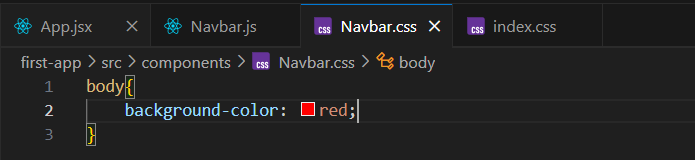
Now, write the css for this nav bar, for which we will create Navbar.css in the same components folder as shown below:



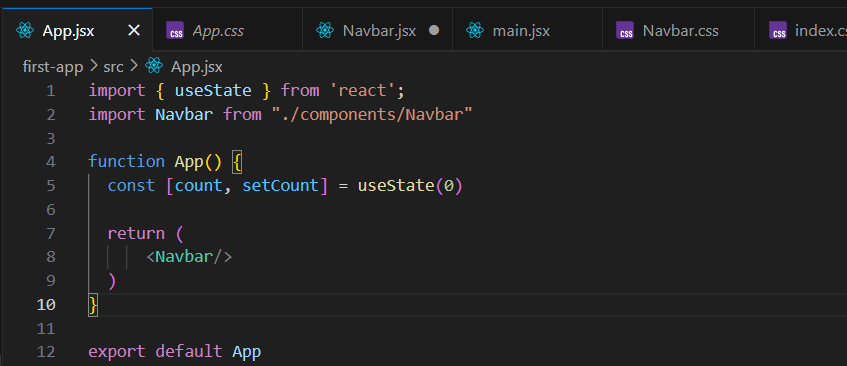
Now, we will import that .css in the Navbar.js as shown below:



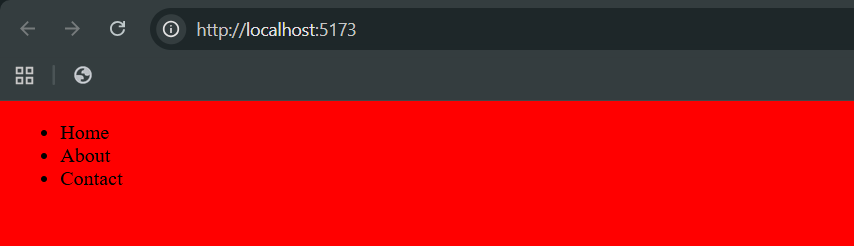
Now, we will write the Navbar.css as shown below:



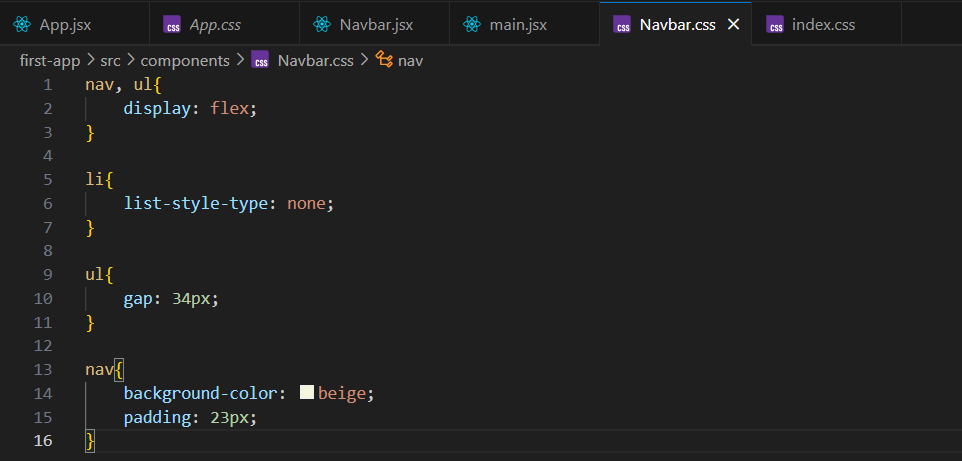
Also, we will import this in the App.jsx as shown below:



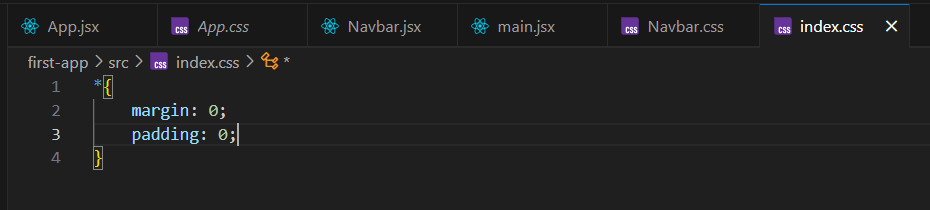
Output:



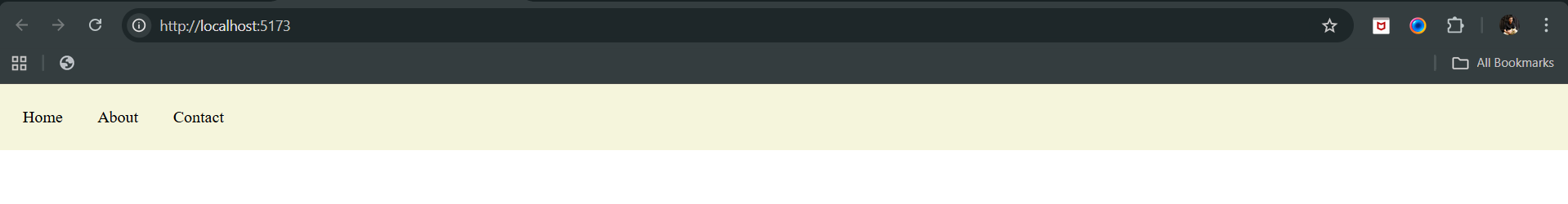
Now, we will arrange this <ul> in such a way that it will work as a Navbar, for which we will change the Navbar.css as:



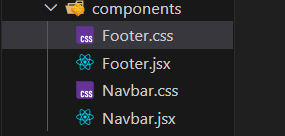
Index.css:



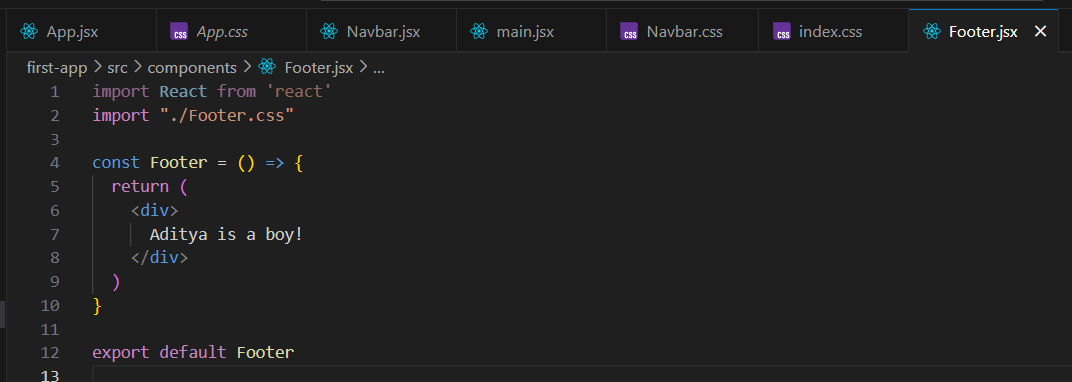
Output:



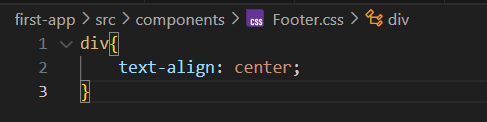
Now, suppose we want to add the footer, then for it, we will create the footer.jsx and footer.css in the components folder:



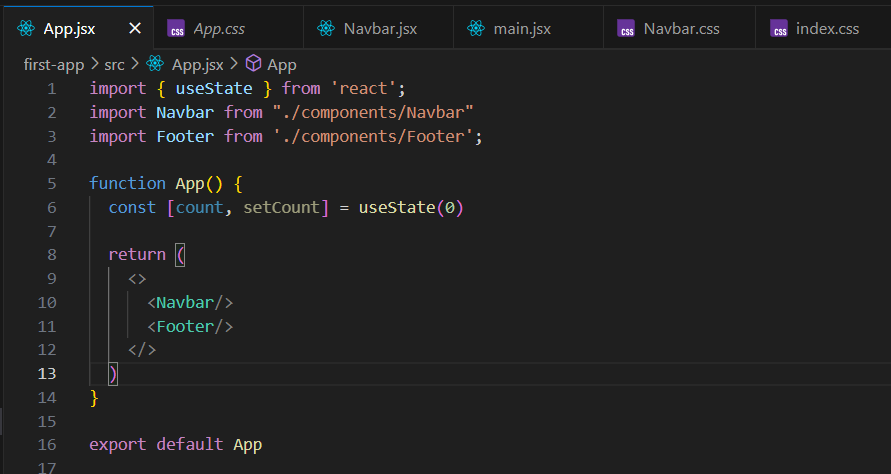
Now, we will write code for the footer, first we will write the Footer.jsx:



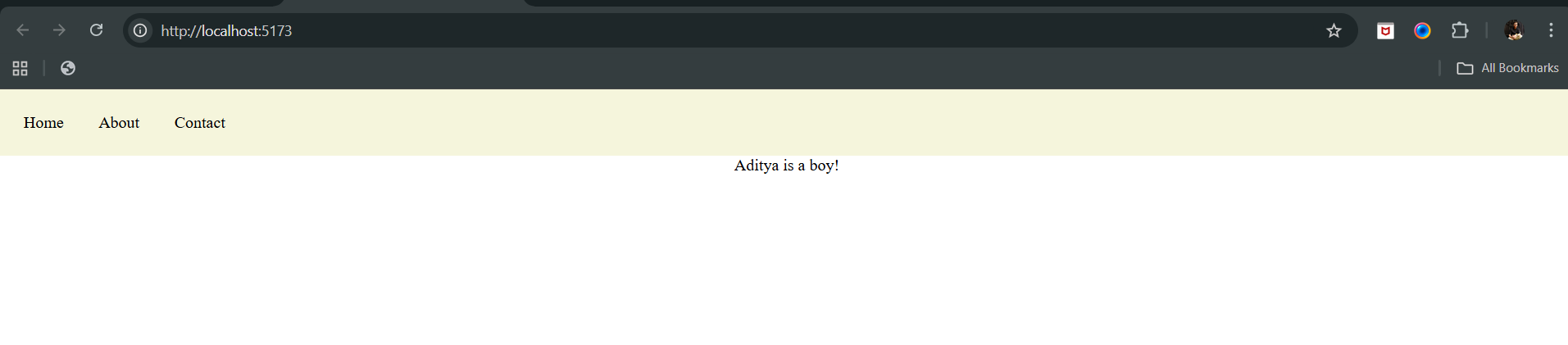
Footer.css:



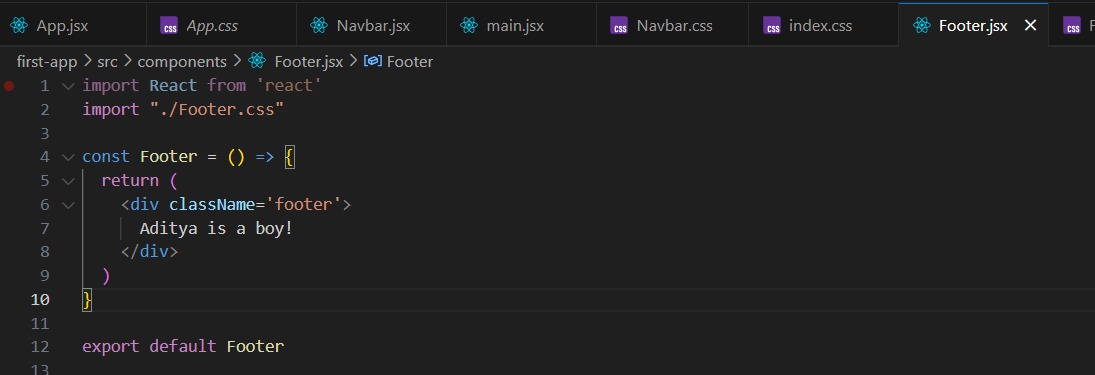
Also, include the Footer in the App.jsx:



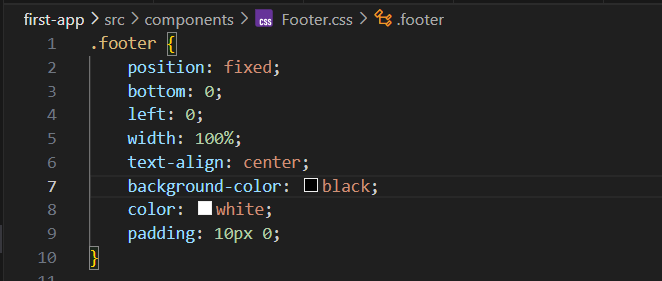
Output: as expected footer text is at the center.



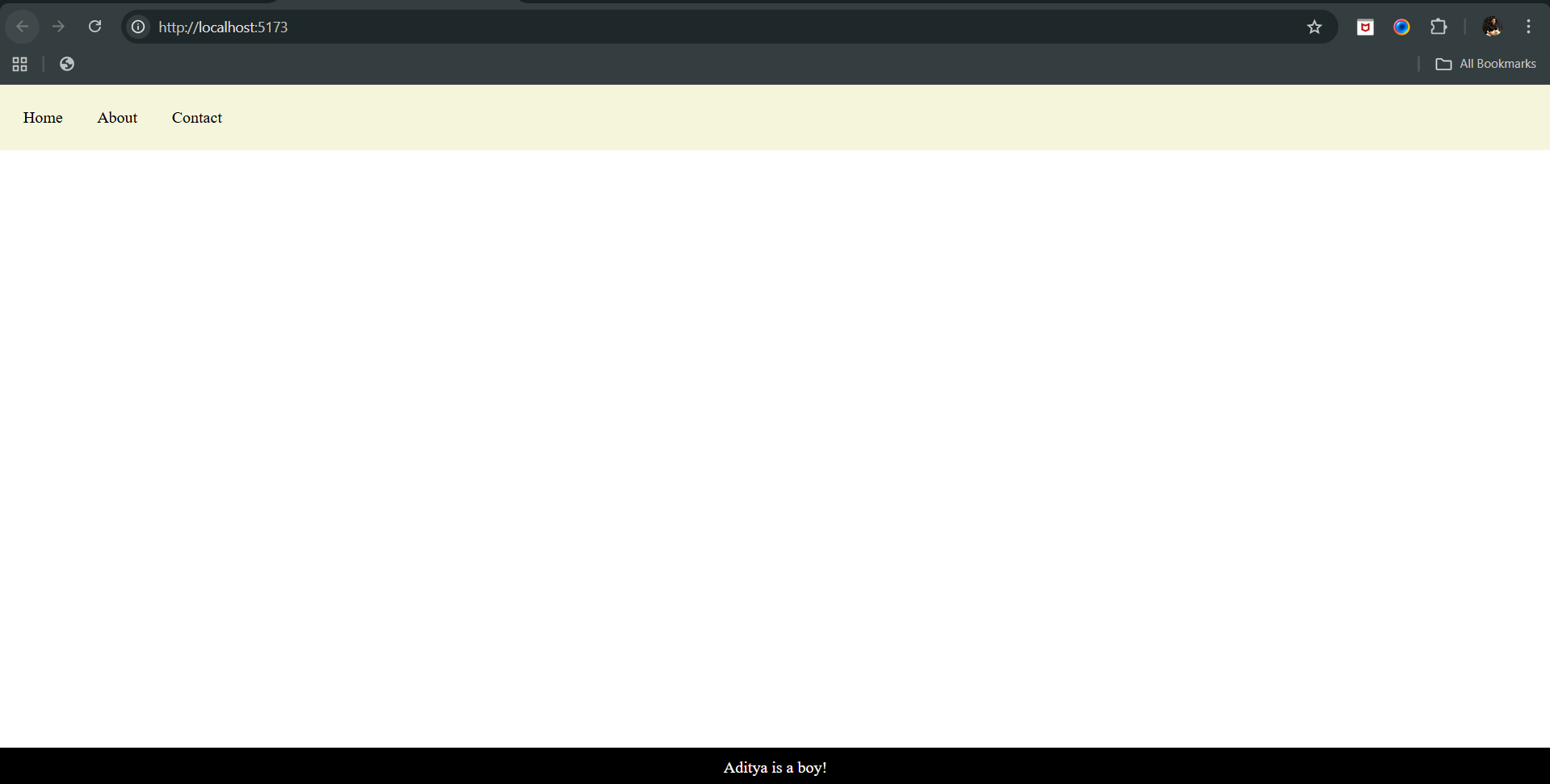
Now, since we want to target the <div> of the Footer.jsx specifically, we need to assign the class name to it as shown below:



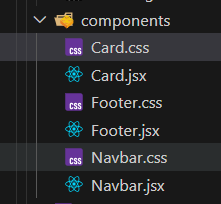
Now, redefining the Footer.css:

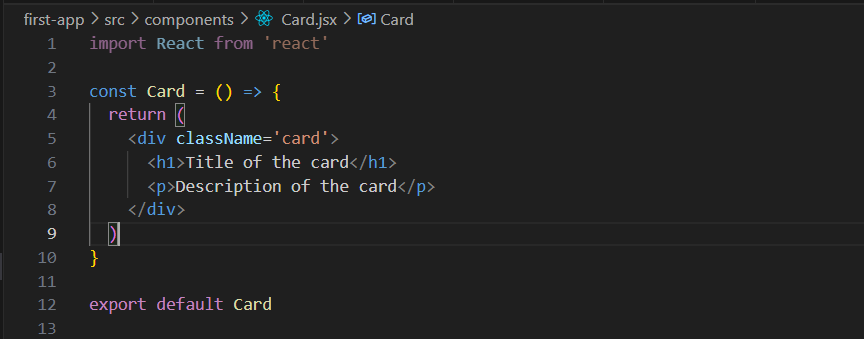


Output:

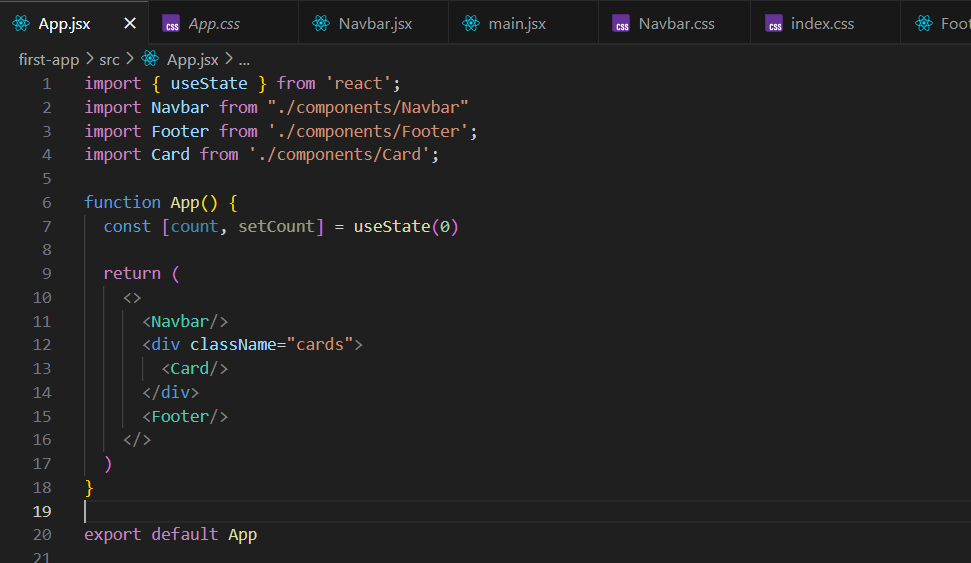


Now, let’s add the card component to it: for that create Card.jsx as well as Card.css in the components folder

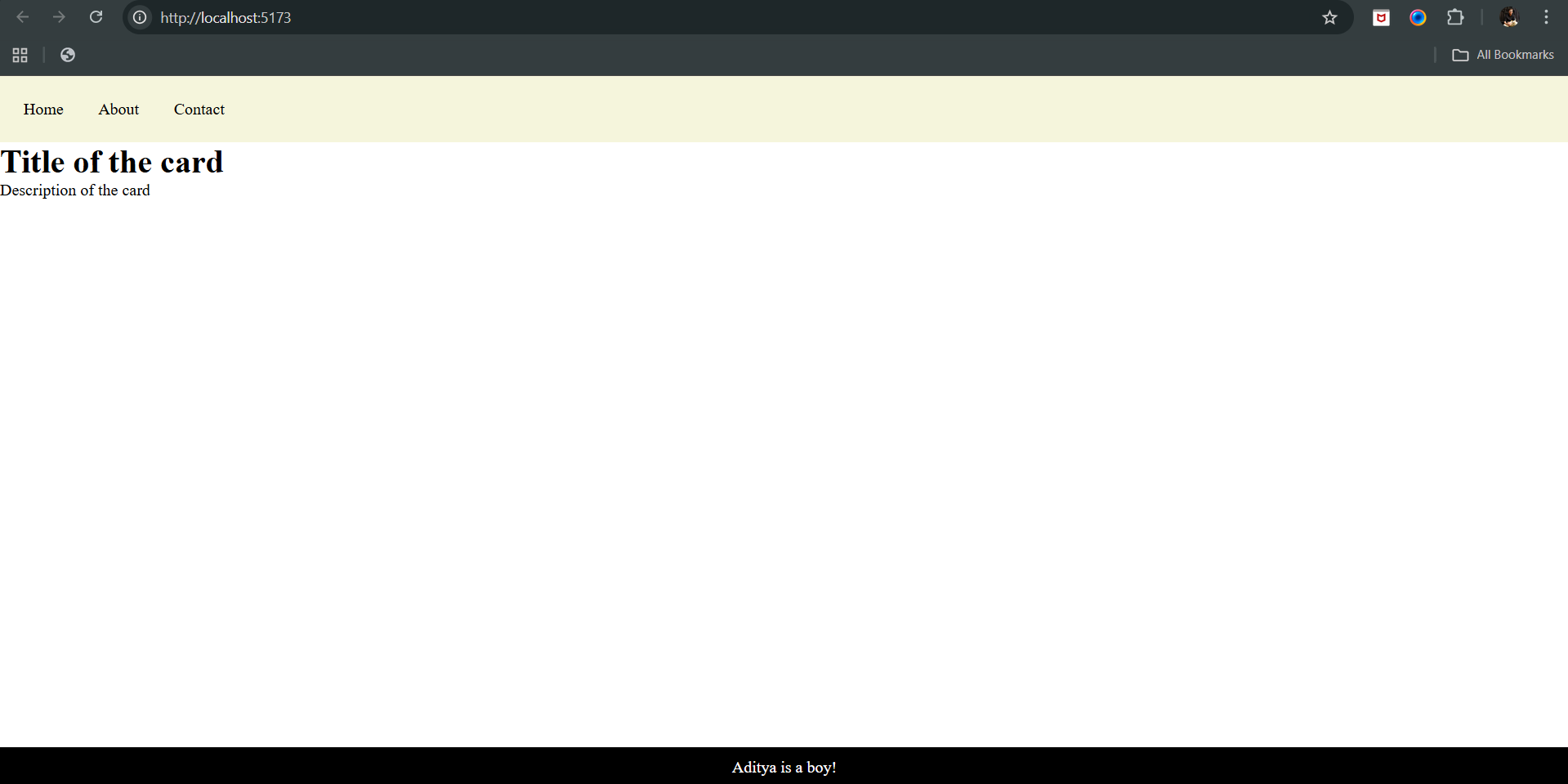


Code for the Card.jsx:  


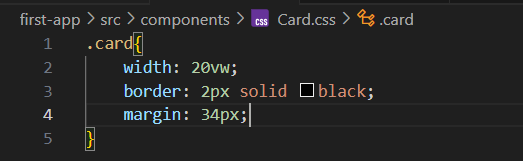
Code for the updated App.jsx:



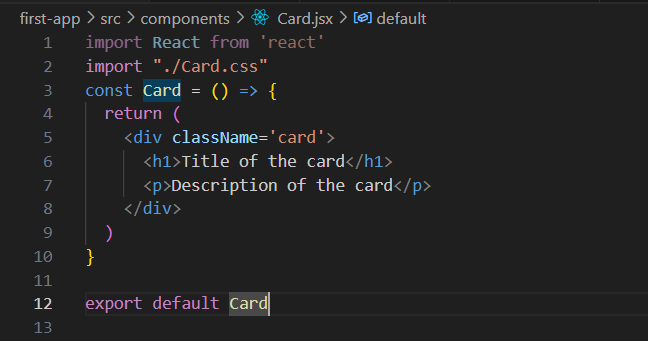
Output:



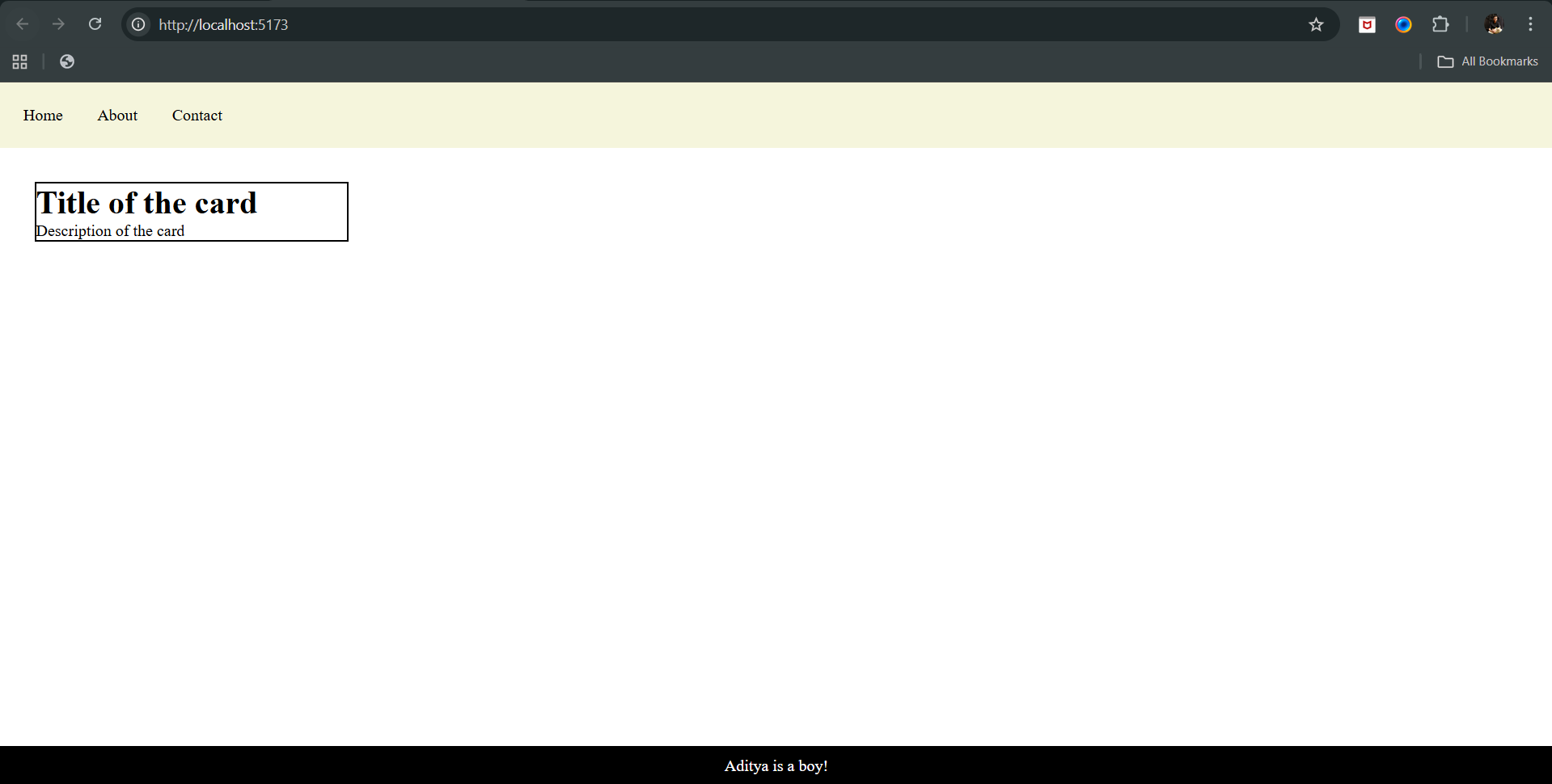
Now, we will style these cards:



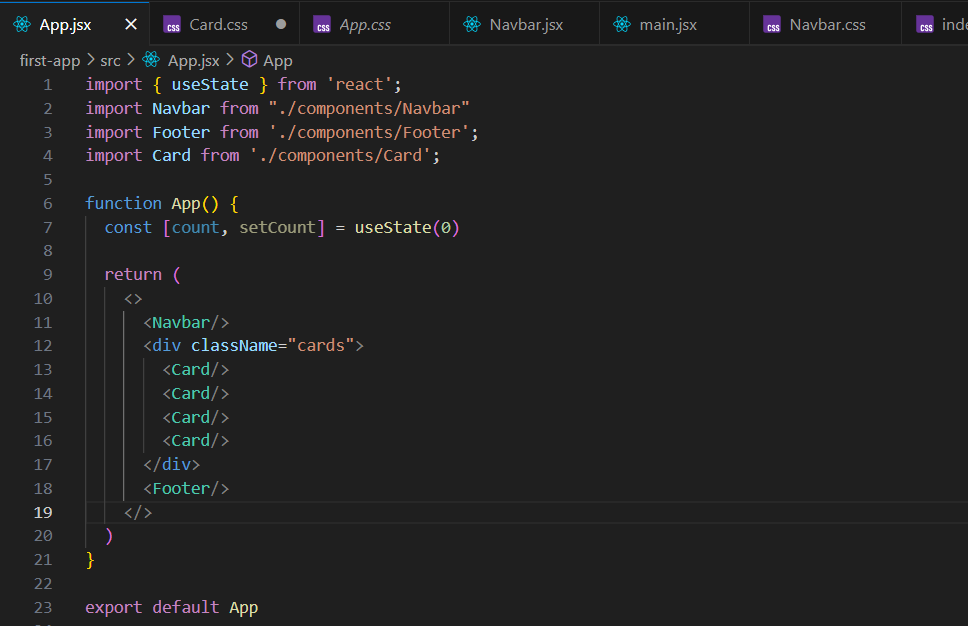
Import it in the Card.jsx too:



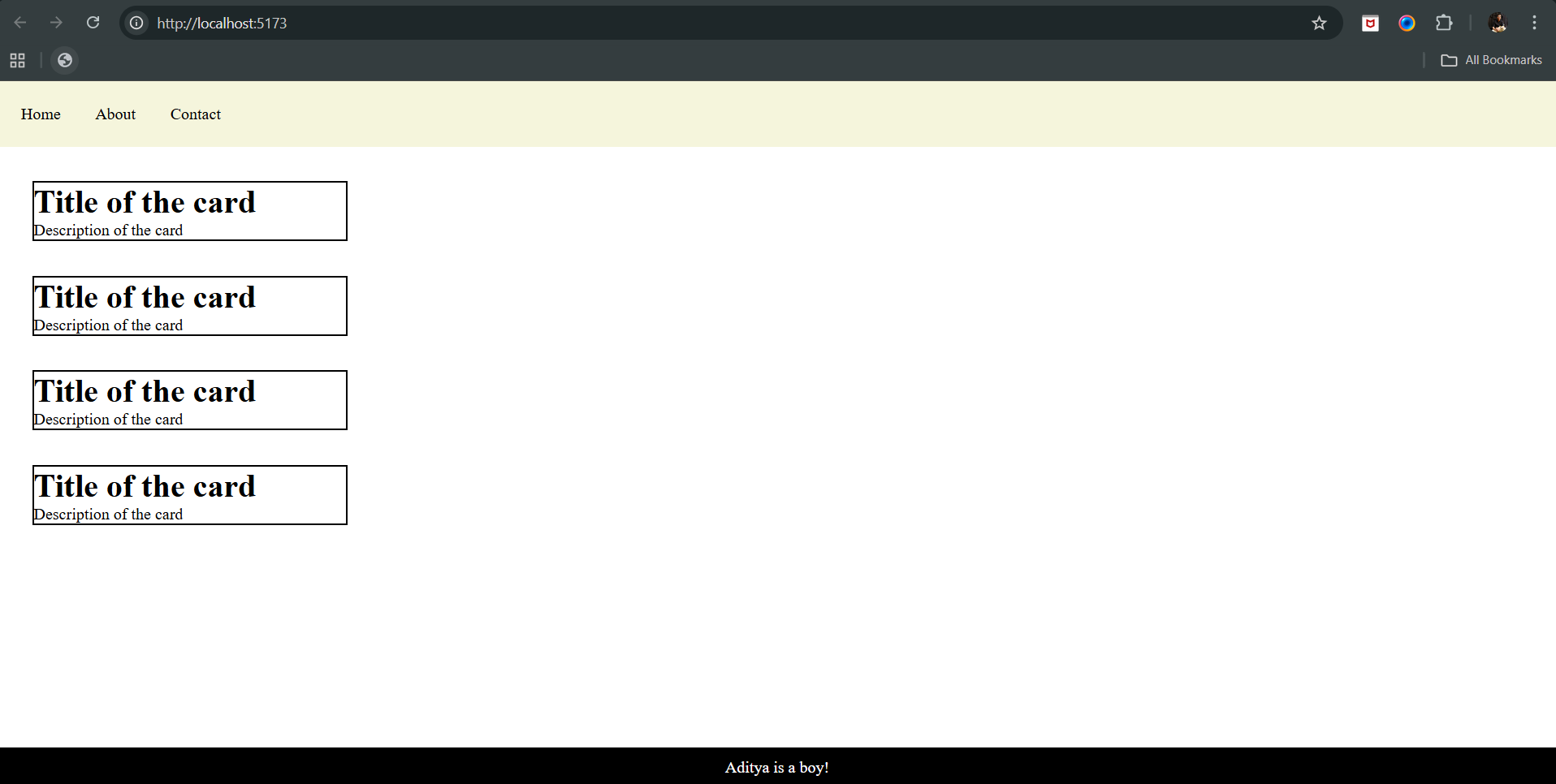
Output:



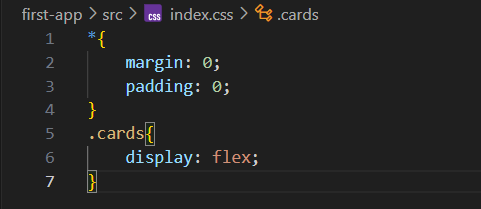
Now, we can duplicate these cards by simply writing the <Card/> in App.jsx as many times we wish:



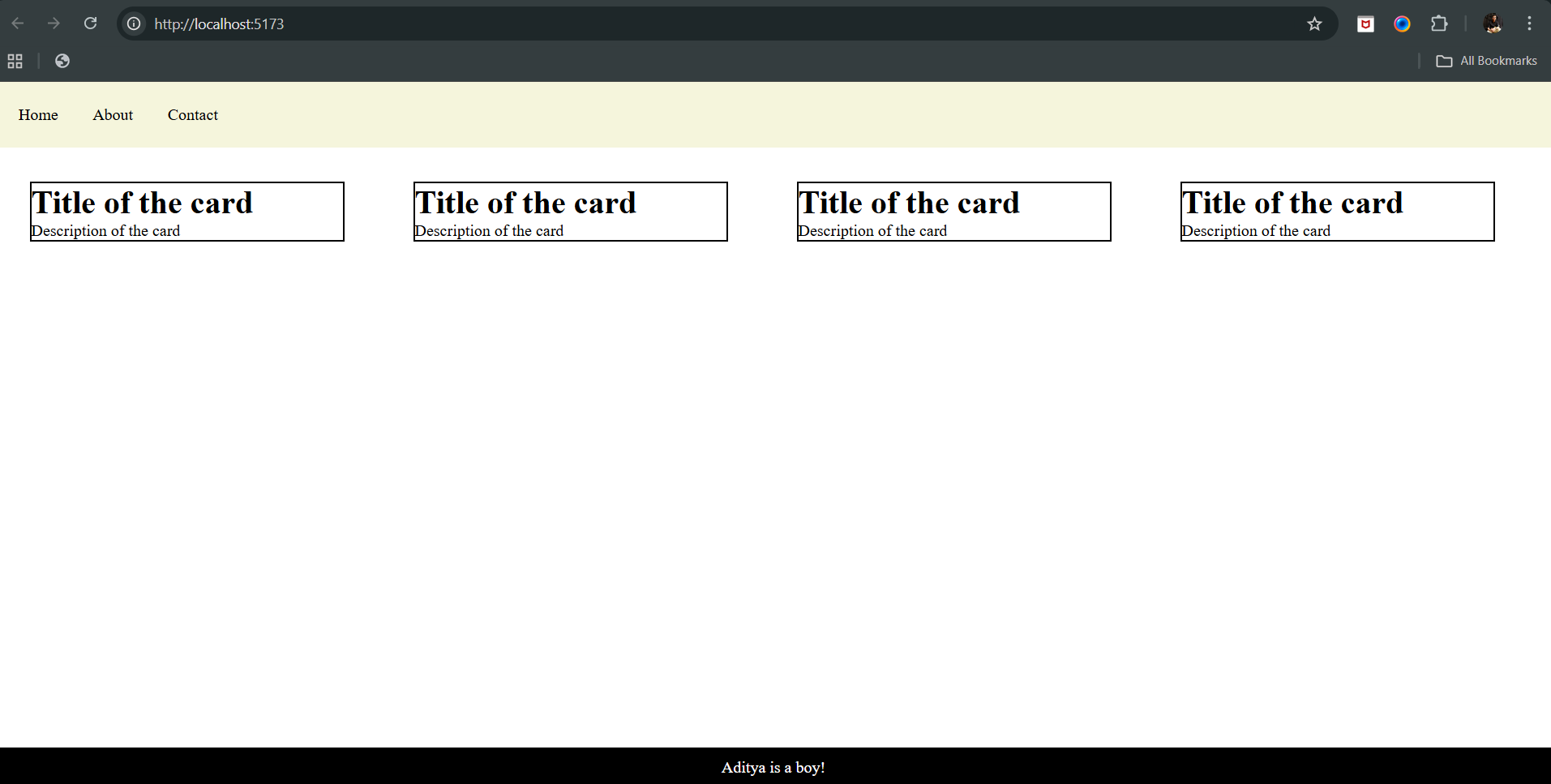
Output:



Now, we can also make these boxes come in one row, if we change the flex of the cards class in the index.css as shown below:

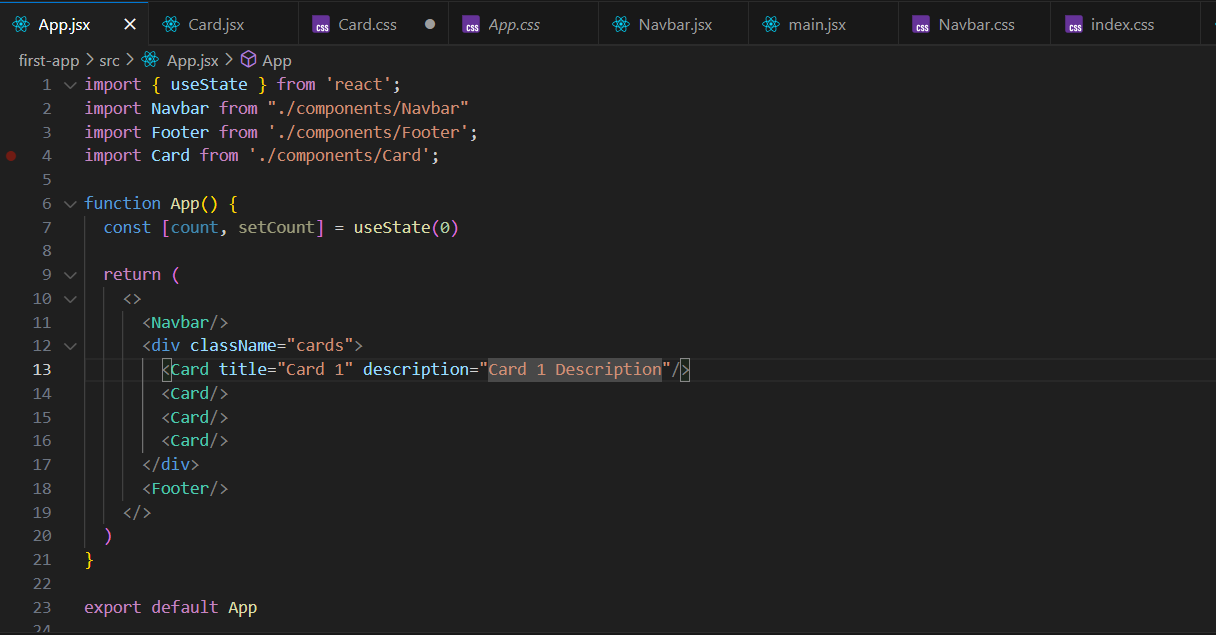


Output:

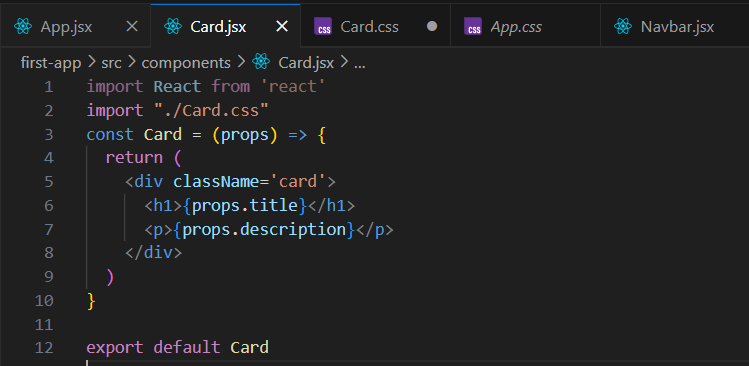


So, what next?

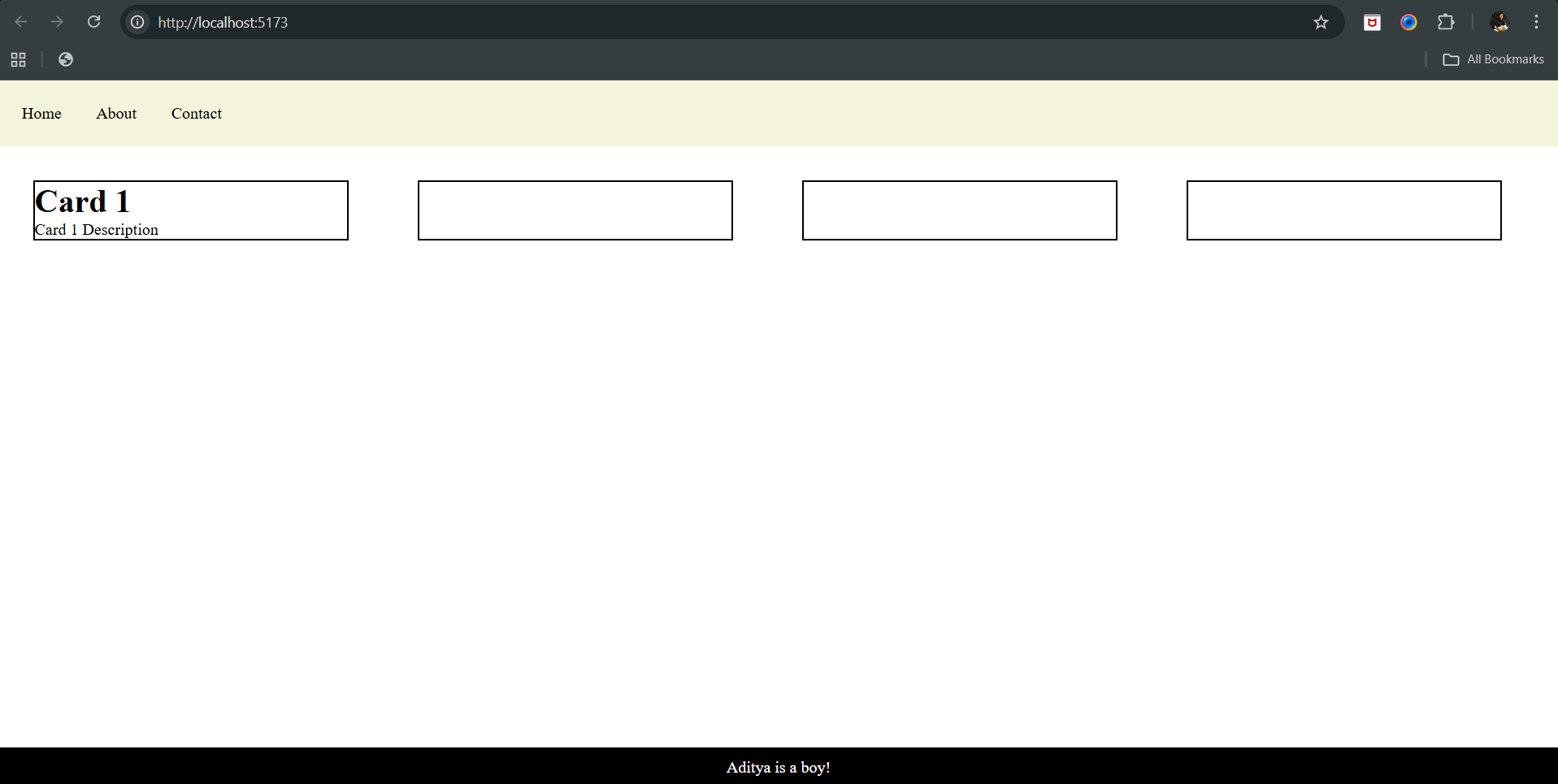
We can also, make the card entry from the App.jsx easily. How? We will use props:

App.jsx: 

Now, we will direct our Card.jsx to accept props, Card.jsx:



Output: clearly, whatever we had passed via the title and description in the <Card/> is printed



Why others are blank? Because we haven’t passed any value in it yet.

So, let’s update and check:

App.jsx:

Output:



--The End--