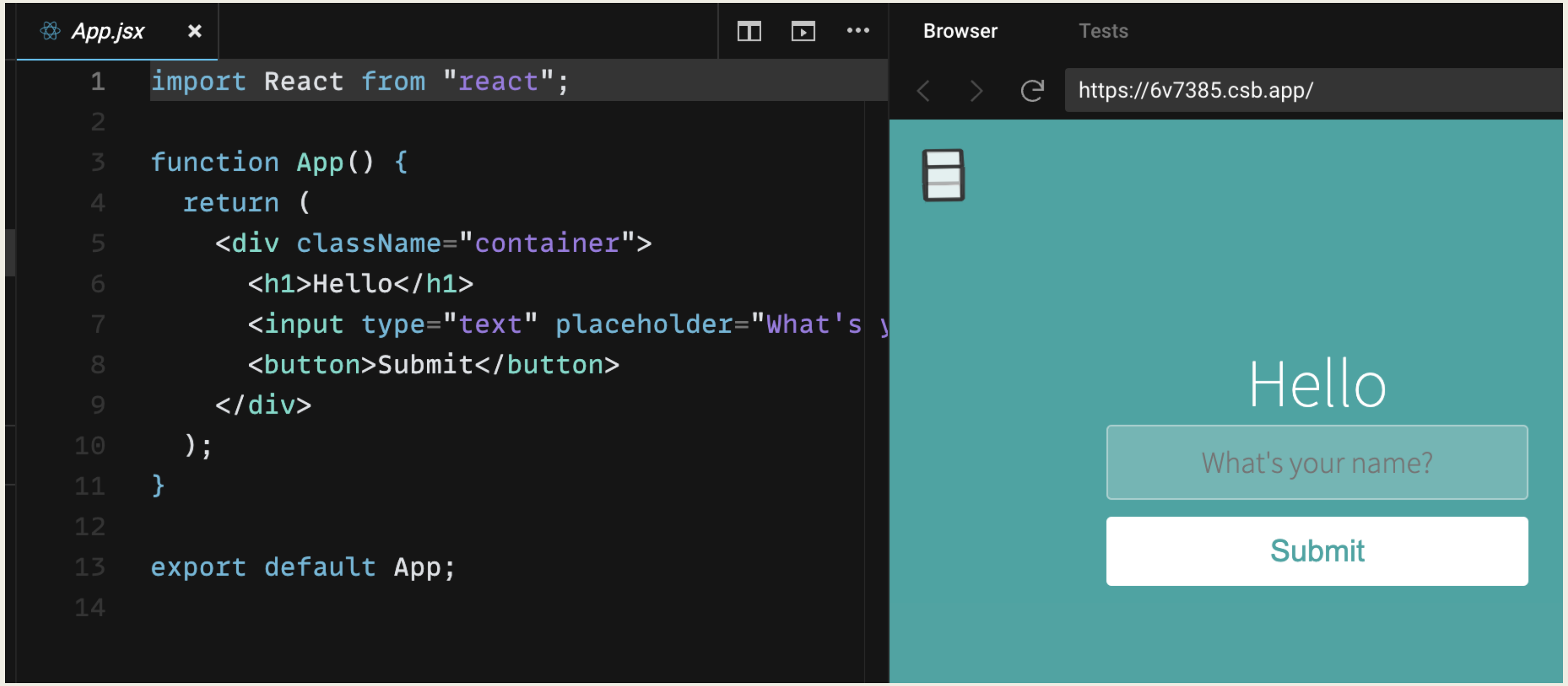
EVENT HANDLING

We will learn -

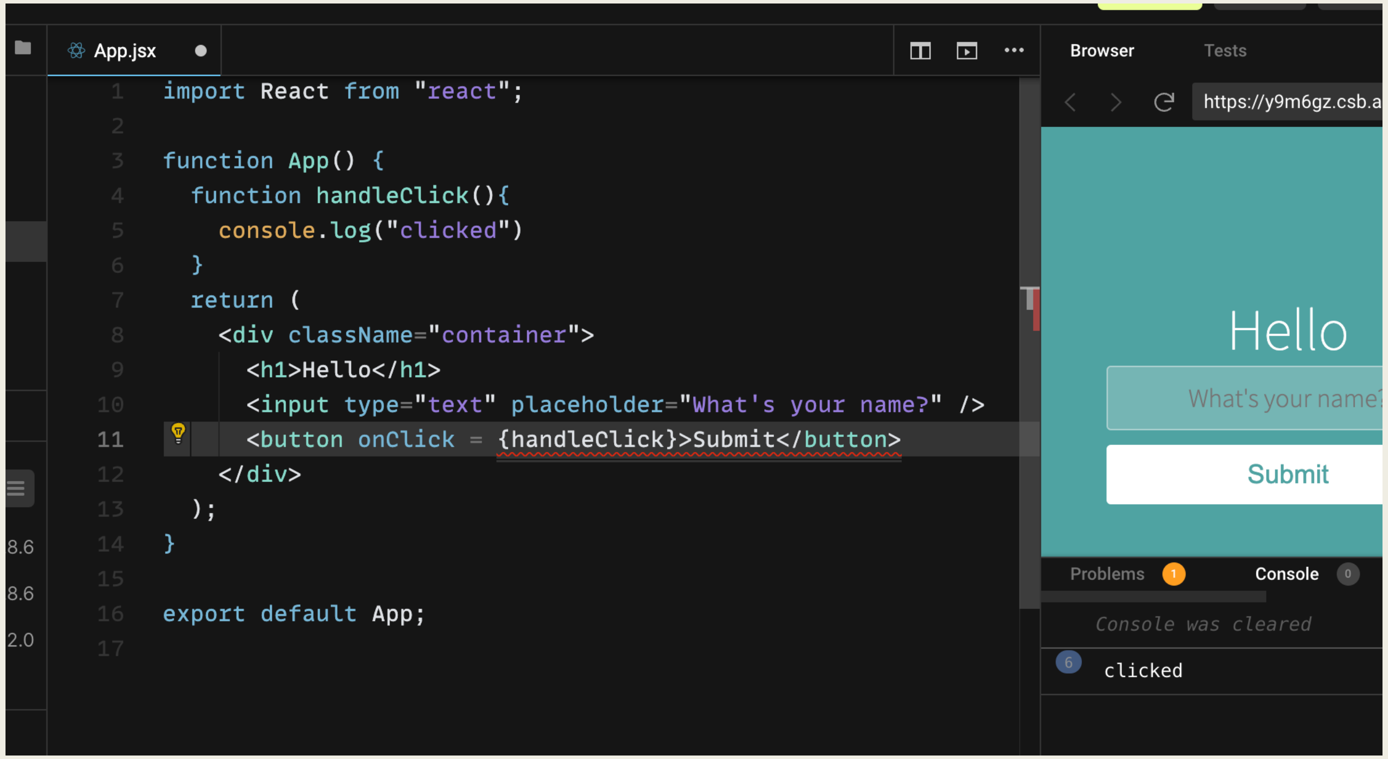
- How to detect when users interact with the app and
- How to render different things based on those events.
- Go here - https://codesandbox.io/s/event-handling-in-react-forked-msr7np

App.jsx – h1, input, button



```jsx
import React from "react";

function App() {
  return (
    <div className="container">
      <h1>Hello</h1>
      <input type="text" placeholder="What's y
      <button>Submit</button>
    </div>
  );
}

export default App;
```

Activating button : onClick, handleClick()



App.jsx

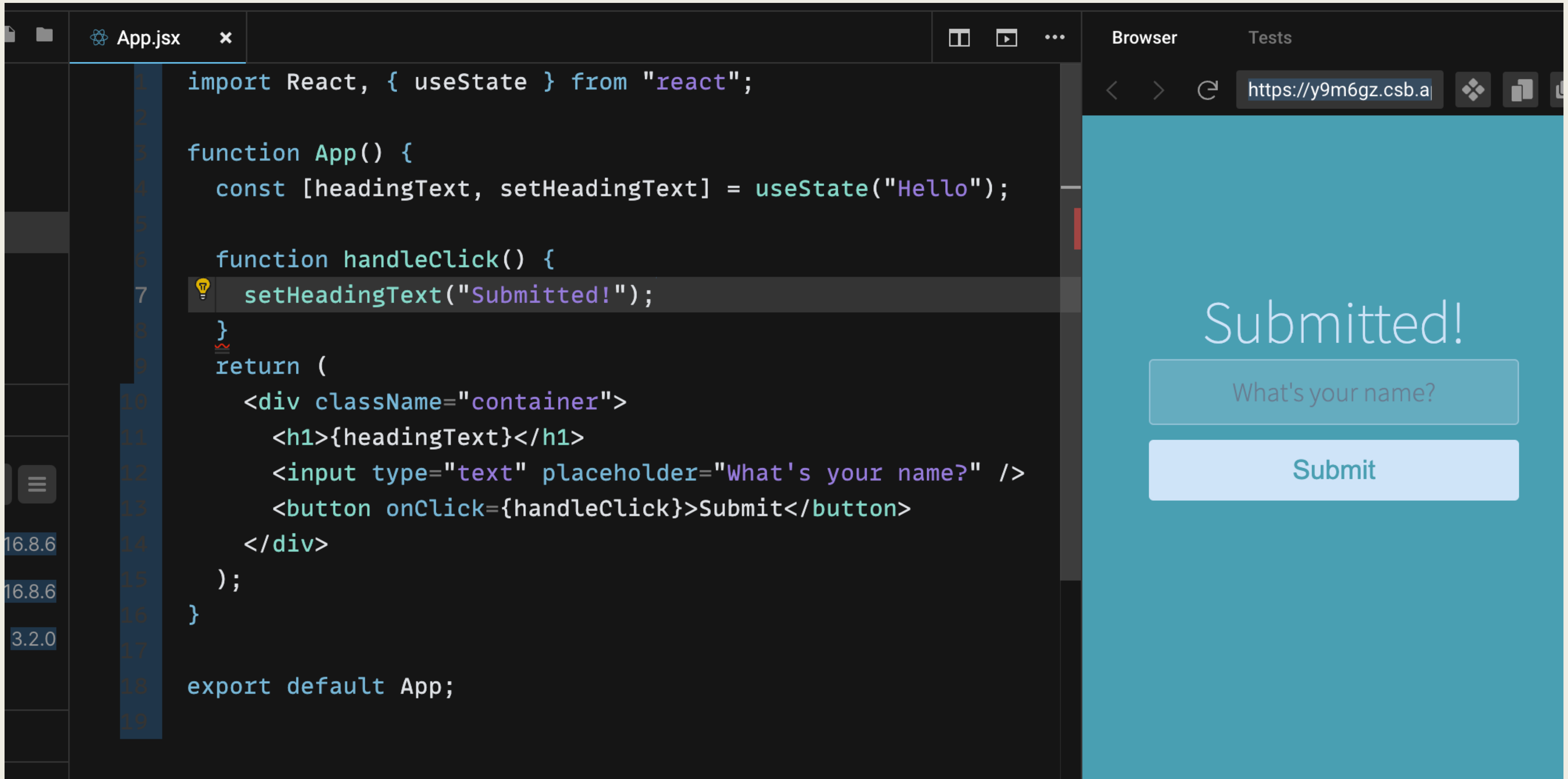
```jsx
import React from "react";

function App() {
  function handleClick(){
    console.log("clicked")
  }
  return (
    <div className="container">
      <h1>Hello</h1>
      <input type="text" placeholder="What's your name?" />
      <button onClick = {handleClick}>Submit</button>
    </div>
  );
}

export default App;
```

```
Console was cleared
6 clicked
```

How can we change the content of h1 when the button is clicked?

const [headingText, setHeadingText] = useState();

- Use useState() hooks, declare the starting state and a function that change that starting state.



```jsx
import React, { useState } from "react";

function App() {
  const [headingText, setHeadingText] = useState("Hello");

  function handleClick() {
    setHeadingText("Submitted!");
  }
  return (
    <div className="container">
      <h1>{headingText}</h1>
      <input type="text" placeholder="What's your name?" />
      <button onClick={handleClick}>Submit</button>
    </div>
  );
}

export default App;
```

What are the different events?

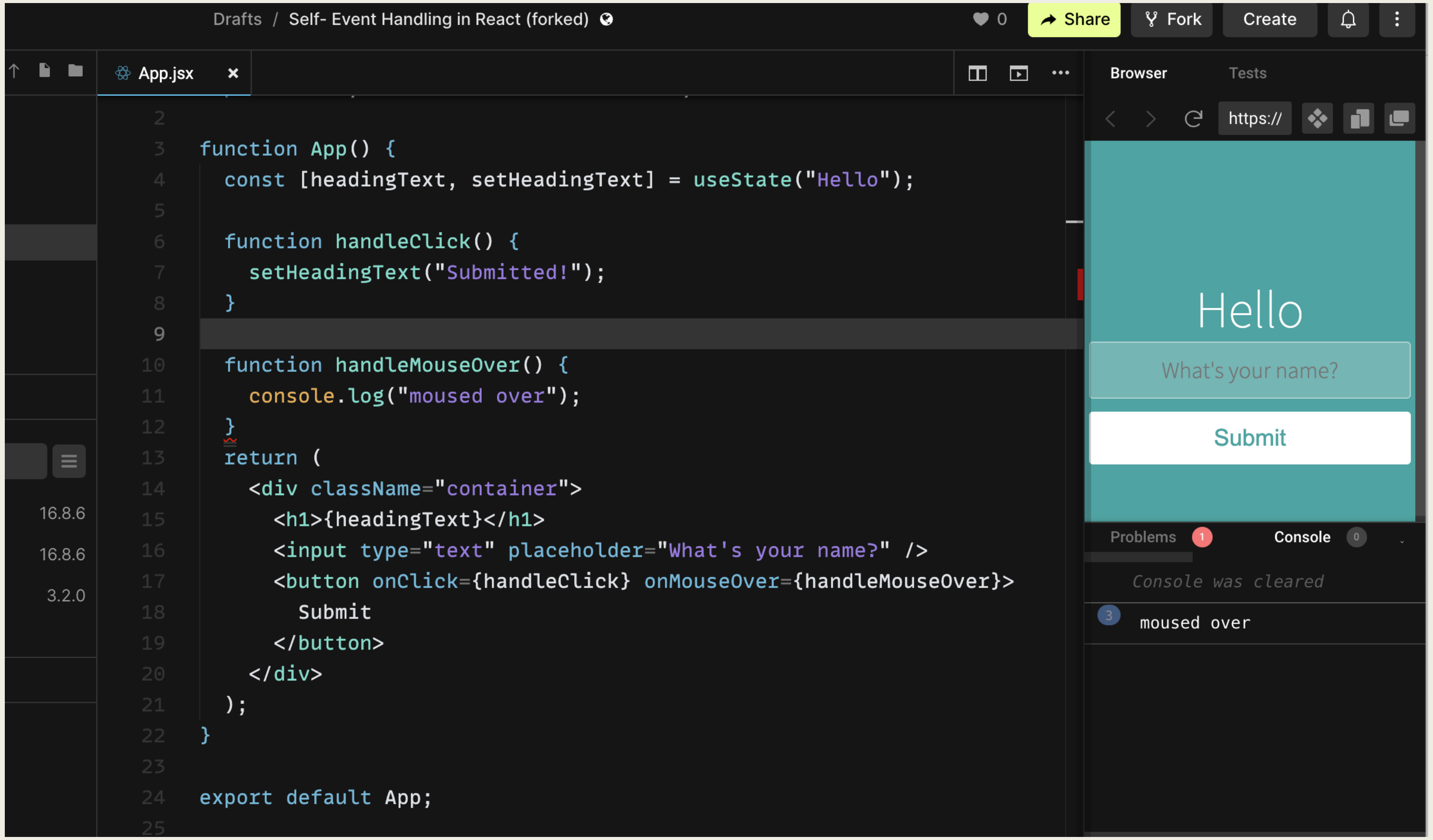
- List - https://www.w3schools.com/tags/ref_eventattributes.asp

Challenge :

- Use onmouseout and onmouseover events to change the background color of submit button when hovered over.

Solution

- Use onMouseOver = handleMouseOver



```jsx

function App() {
  const [headingText, setHeadingText] = useState("Hello");

  function handleClick() {
    setHeadingText("Submitted!");
  }

  function handleMouseOver() {
    console.log("moused over");
  }
  return (
    <div className="container">
      <h1>{headingText}</h1>
      <input type="text" placeholder="What's your name?" />
      <button onClick={handleClick} onMouseOver={handleMouseOver}>
        Submit
      </button>
    </div>
  );
}

export default App;

```

Hello

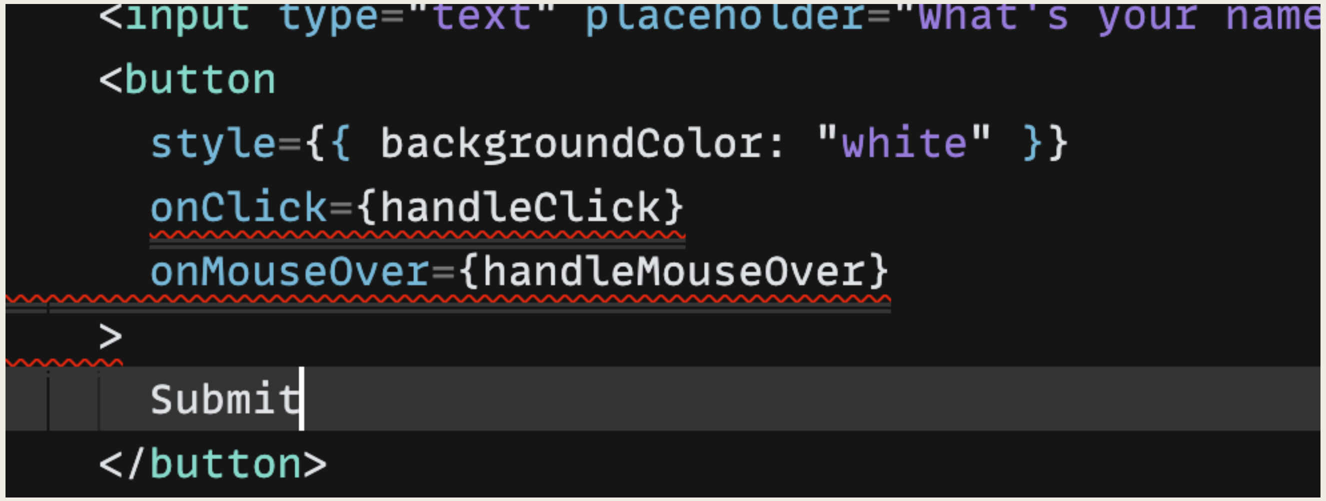
What's your name?

Submit

```
Console was cleared
3 moused over
```

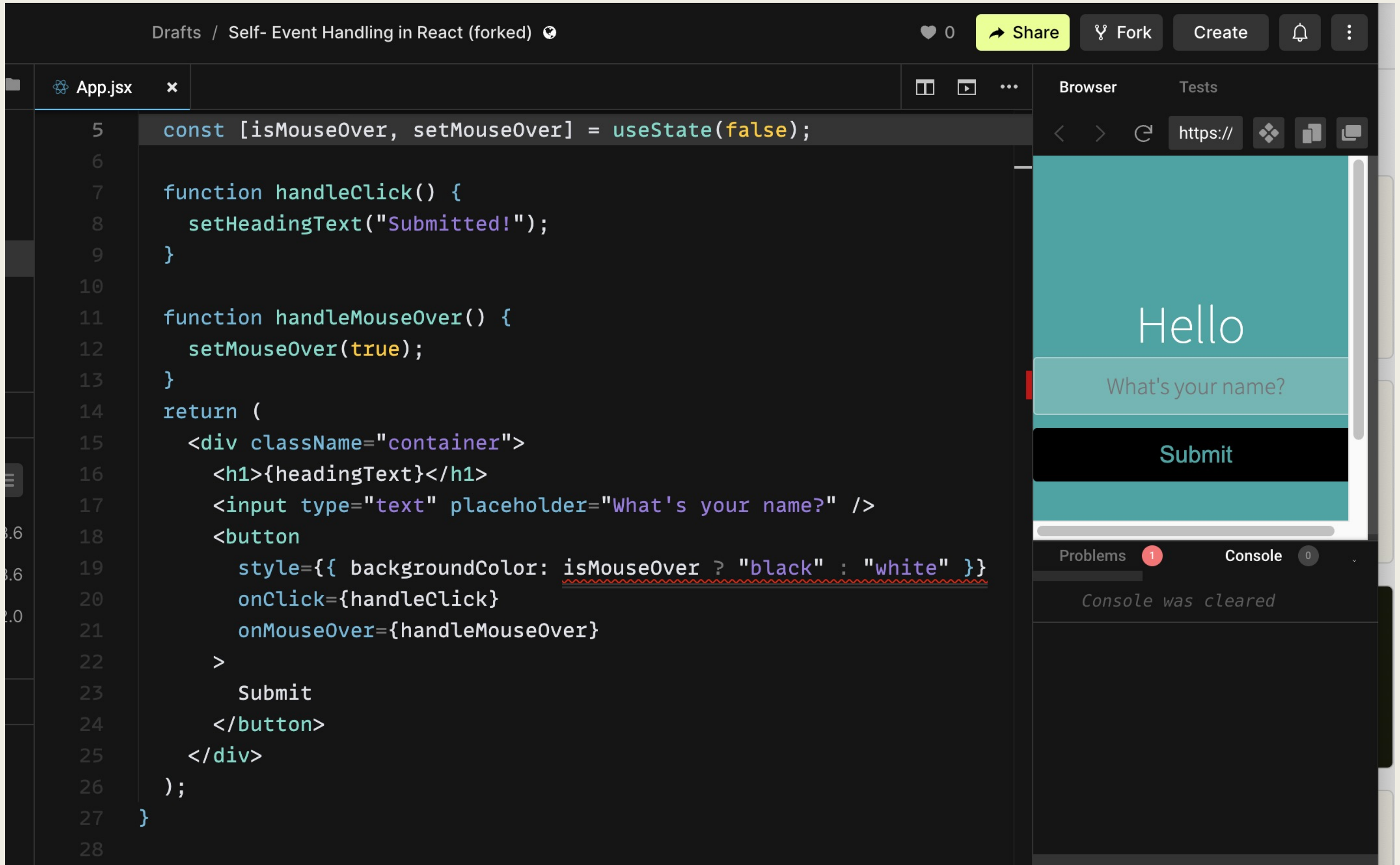
Use State to change the background color

- const [isMouseOver, setMouseOver] = useState(false);
- Starting value is false assuming when the page loads, mouse will not be on the button.
- Once the event is triggered handleMouseOver() sets SetMouseOver(true);
- Also set a backgroundColor :”white”



```
<input type="text" placeholder="What's your name
<button
  style={{ backgroundColor: "white" }}
  onClick={handleClick}
  onMouseOver={handleMouseOver}
>
  Submit
</button>
```

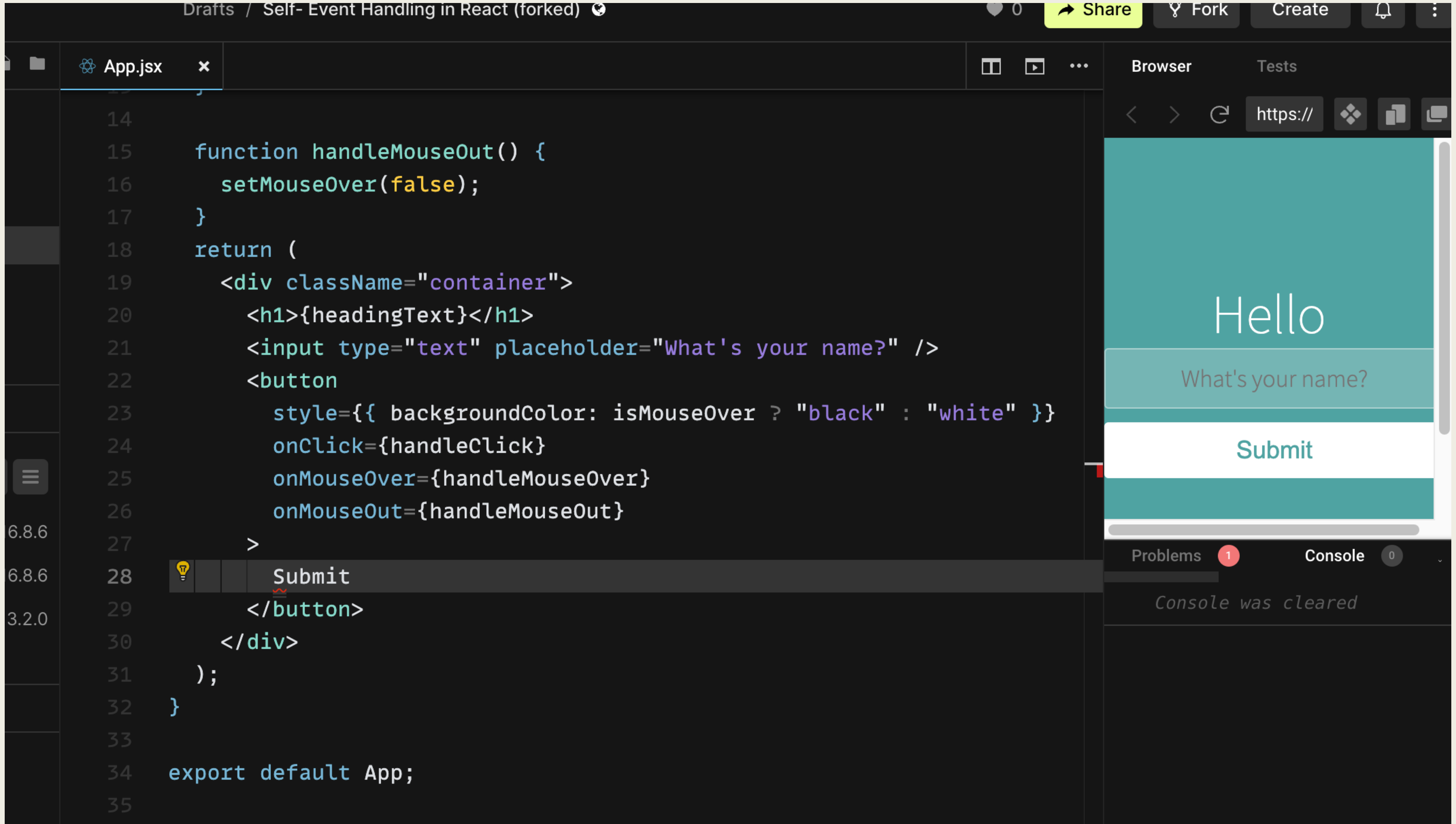
Background color changed via mouseover



```jsx
  const [isMouseOver, setMouseOver] = useState(false);

  function handleClick() {
    setHeadingText("Submitted!");
  }

  function handleMouseOver() {
    setMouseOver(true);
  }
  return (
    <div className="container">
      <h1>{headingText}</h1>
      <input type="text" placeholder="What's your name?" />
      <button
        style={{ backgroundColor: isMouseOver ? "black" : "white" }}
        onClick={handleClick}
        onMouseOver={handleMouseOver}
      >
        Submit
      </button>
    </div>
  );
}
```

How to change back to white color when mouse is out?

Solution : HandleMouseOut is set to false



```
  function handleMouseOut() {
    setMouseOver(false);
  }
  return (
    <div className="container">
      <h1>{headingText}</h1>
      <input type="text" placeholder="What's your name?" />
      <button
        style={{ backgroundColor: isMouseOver ? "black" : "white" }}
        onClick={handleClick}
        onMouseOver={handleMouseOver}
        onMouseOut={handleMouseOut}
      >
        Submit
      </button>
    </div>
  );
}

export default App;
```

```
Console was cleared
```

- Learn more about conditional (ternary operator) - https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Conditional_operator

- The full code can be viewed as - https://codesandbox.io/s/self-event-handling-in-react-forked-y9m6gz?file=/src/components/App.jsx