



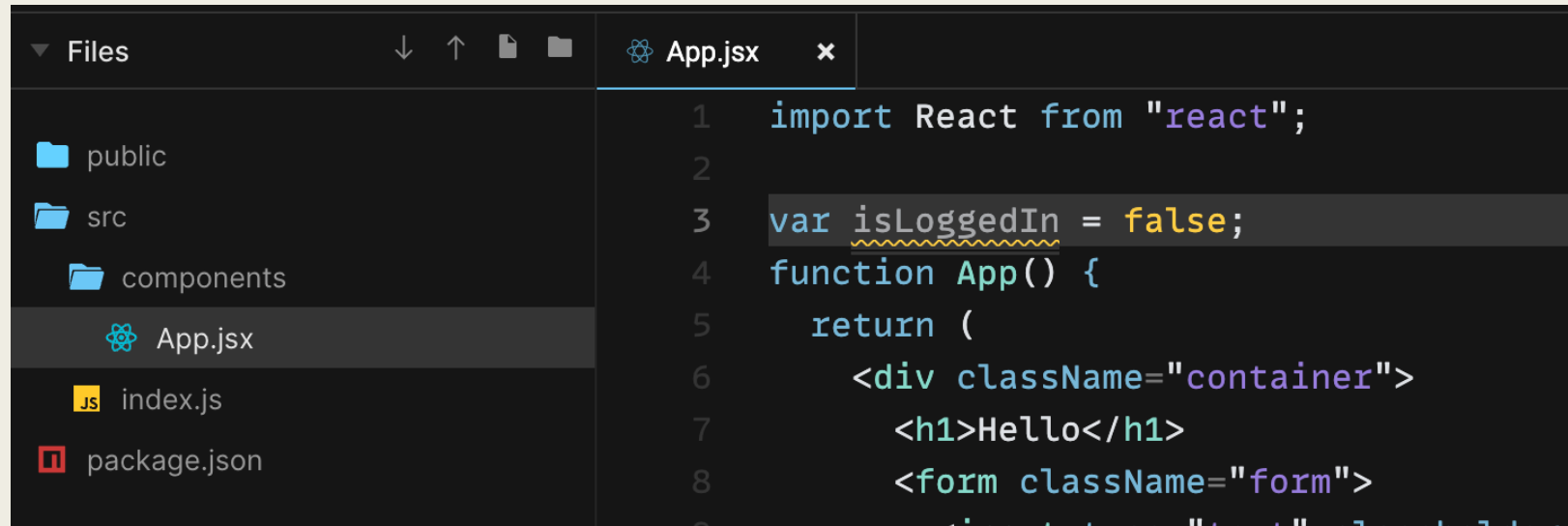
REACT RENDERING

Conditional Rendering, Ternary Operator



- Codesandbox - <https://codesandbox.io/s/student-conditional-rendering-forked-v3pf4h?file=/src/components/App.jsx>

1. Create a variable `isLoggedIn` and assign `true` as it's value.



```
1 import React from "react";
2
3 var isLoggedIn = false;
4 function App() {
5   return (
6     <div className="container">
7       <h1>Hello</h1>
8       <form className="form">
```

2. We want to check the value of variable `isLoggedIn` and based on true render `h1` and based on false render the login form.

How would you do this? Are you thinking
a conditional statement?

App.jsx

```
1 import React from "react";
2
3 var isLoggedIn = false;
4
5 function renderConditionally() {
6   if (isLoggedIn === true) {
7     return <h1>Hello</h1>;
8   } else {
9     return (
10      <form className="form">
11        <input type="text" placeholder="Username" />
12        <input type="password" placeholder="Password" />
13        <button type="submit">Login</button>
14      </form>
15    );
16   }
17 }
18
19 function App() {
20   return <div className="container">{renderConditionally()}</div>;
21 }
22
23 export default App;
24
```

BrowserTests

https://jfskvl.csb.

Username

Password

Login

Question: What change in the code you need to make to appear Hello?

Answer: `var isLoggedIn = true;`
;

Each component should have single responsibility!

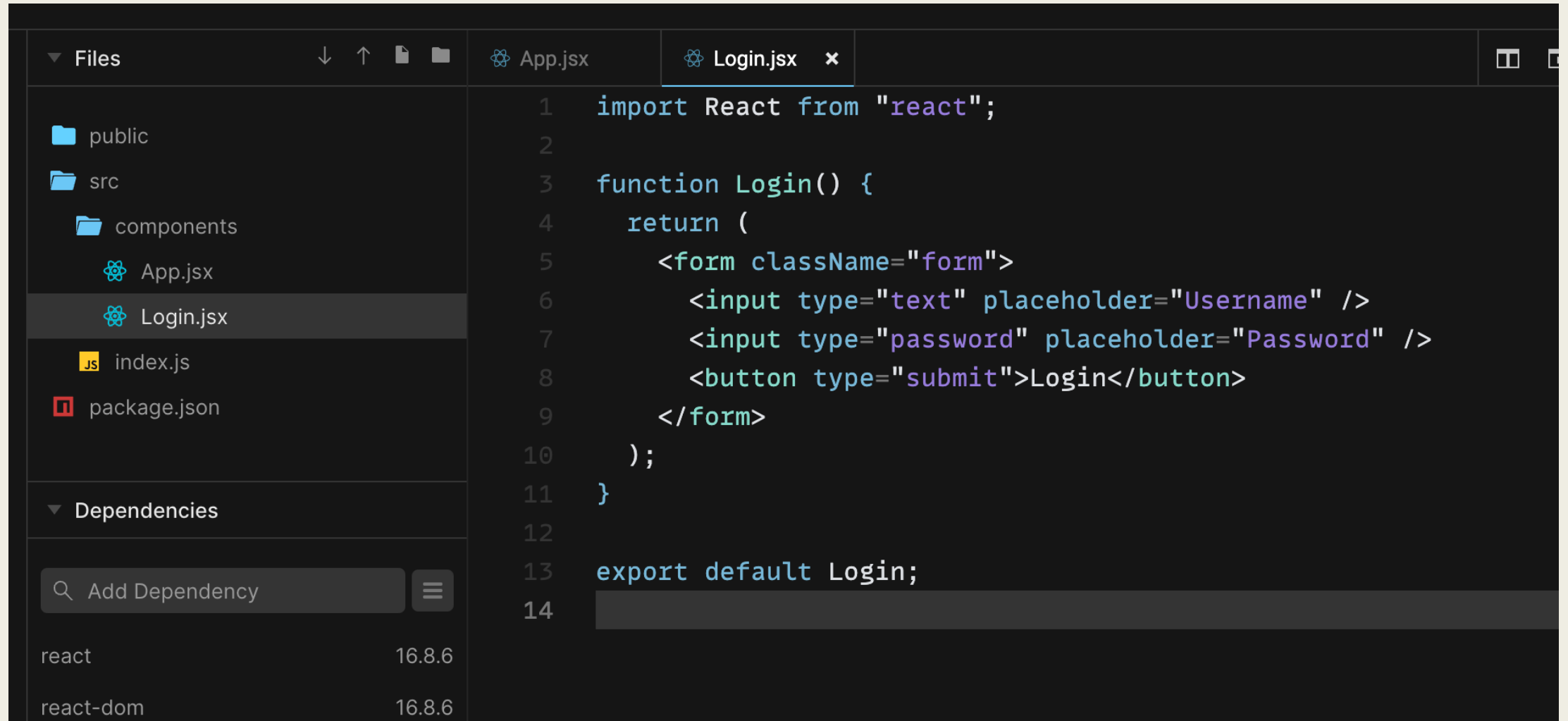
- Single responsibility principle - https://en.wikipedia.org/wiki/Single-responsibility_principle

Our codes are long. How to make it short and component based?

Challenge:

- Create a 'Login' component for the form.
- Create an 'Input' component for the input.

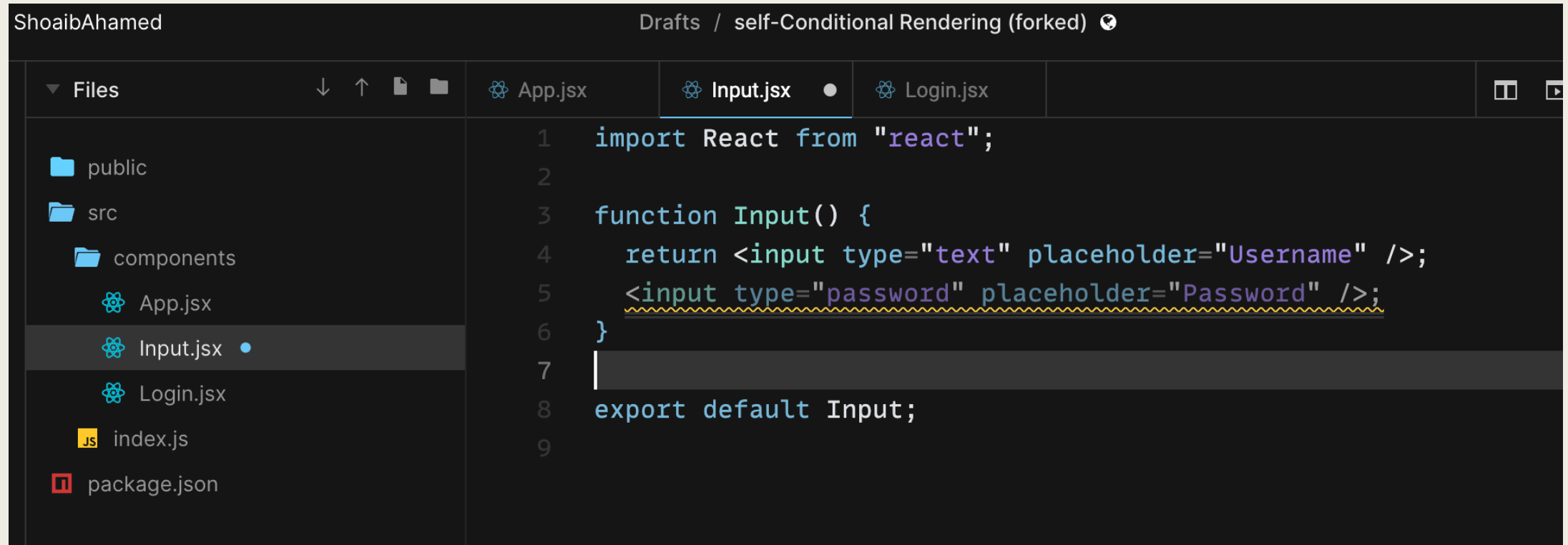
Login.jsx



The screenshot displays a code editor interface with a dark theme. On the left, a sidebar shows the project structure under 'Files'. The 'src' directory contains 'components', which includes 'App.jsx' and 'Login.jsx' (the active file). Below the files, the 'Dependencies' section lists 'react' and 'react-dom', both at version 16.8.6. The main editor area shows the code for 'Login.jsx' with line numbers 1 through 14. The code imports React from 'react', defines a 'Login' function that returns a JSX element, and exports the function as the default export. The JSX element is a form with a className of 'form', containing two input fields (one for 'Username' and one for 'Password') and a 'submit' button labeled 'Login'.

```
1 import React from "react";
2
3 function Login() {
4   return (
5     <form className="form">
6       <input type="text" placeholder="Username" />
7       <input type="password" placeholder="Password" />
8       <button type="submit">Login</button>
9     </form>
10  );
11 }
12
13 export default Login;
14
```

Input.jsx



The screenshot shows a code editor interface with a dark theme. The top bar displays the username 'ShoaibAhamed' and the project path 'Drafts / self-Conditional Rendering (forked)'. The left sidebar shows a file explorer with a tree structure: 'public' folder, 'src' folder, 'components' folder, and files 'App.jsx', 'Input.jsx' (selected), 'Login.jsx', 'index.js', and 'package.json'. The main editor area shows the code for 'Input.jsx' with line numbers 1 through 9. The code defines an 'Input' function that returns two JSX elements: a text input and a password input. The password input is underlined with a yellow wavy line, indicating a warning or error. The code is as follows:

```
1 import React from "react";
2
3 function Input() {
4   return <input type="text" placeholder="Username" />;
5   <input type="password" placeholder="Password" />;
6 }
7
8 export default Input;
9
```

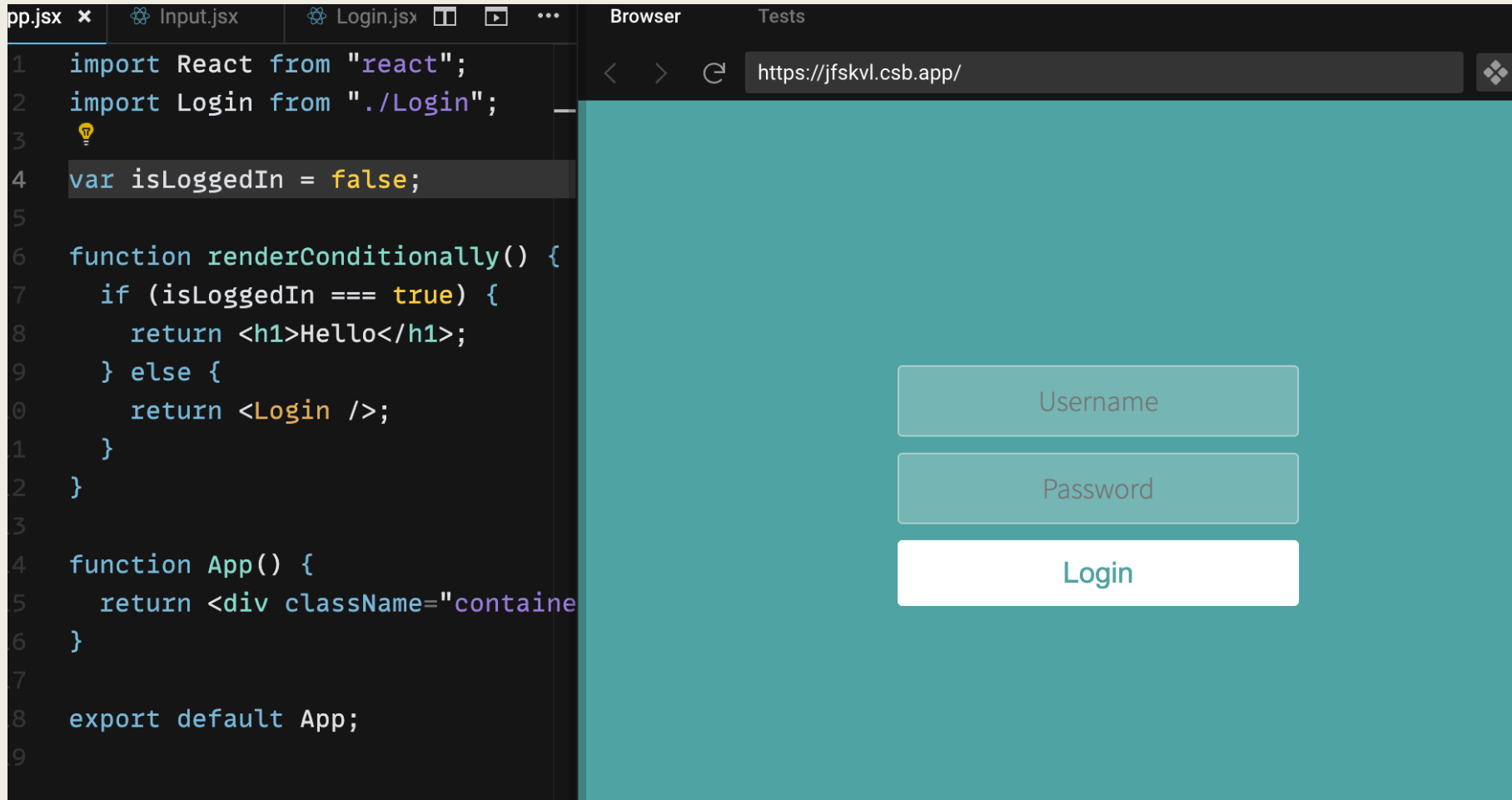
Final Input.jsx

```
App.jsx  Input.jsx  Login.jsx
1  import React from "react";
2
3  function Input(props) {
4    return <input type={props.type} placeholder={props.placeholder} />;
5  }
6
7  export default Input;
8
```

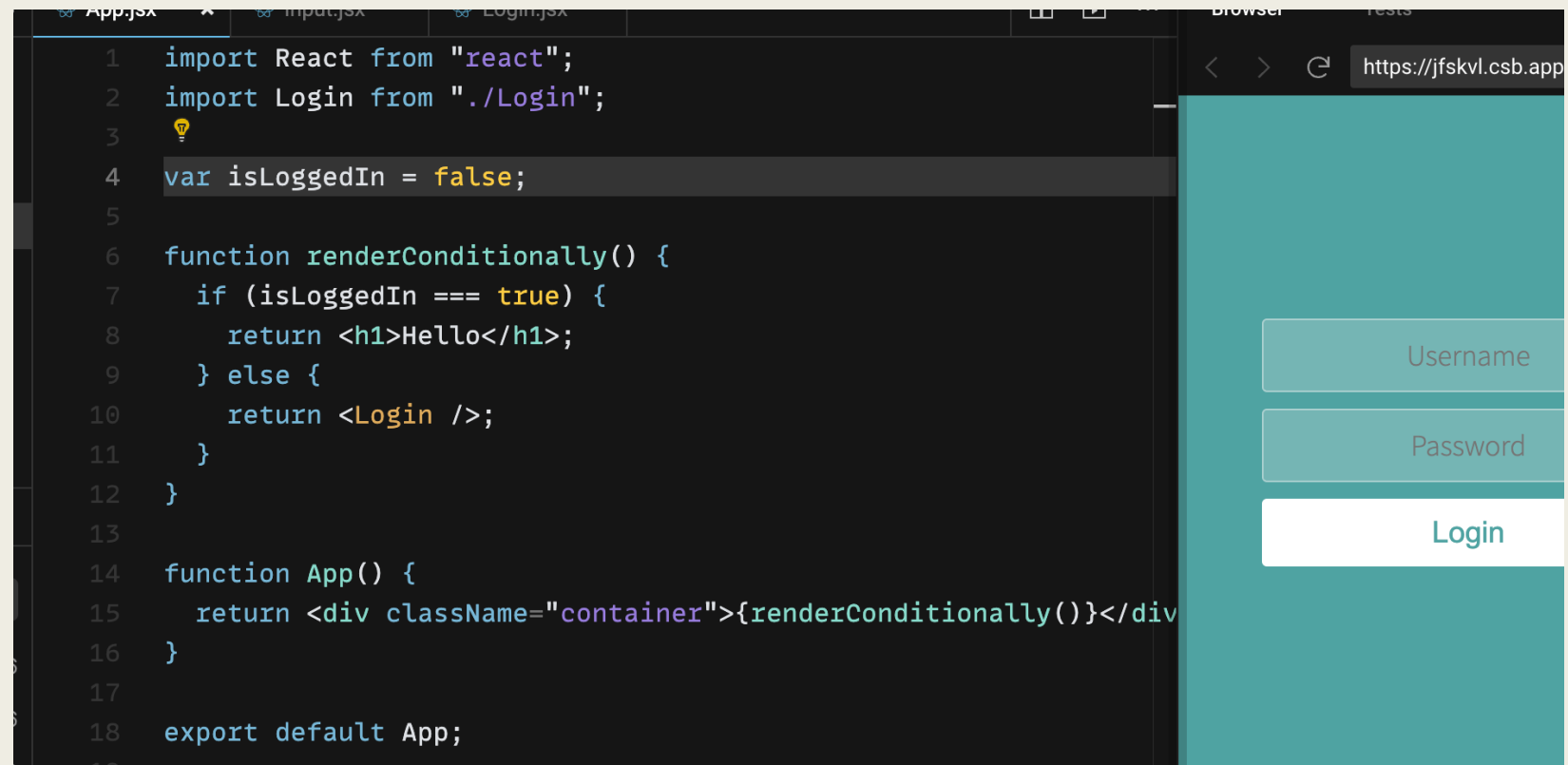
Final Login.jsx

```
App.jsx  Input.jsx  Login.jsx  x
1  import React from "react";
2  import Input from "../Input";
3
4  function Login() {
5    return (
6      <form className="form">
7        <Input type="text" placeholder="Username" />
8        <Input type="password" placeholder="Password" />
9        <button type="submit">Login</button>
10     </form>
11   );
12 }
13
14 export default Login;
15
```

App looks same in browser!



Now our code are shorter but how can we optimize it even more?



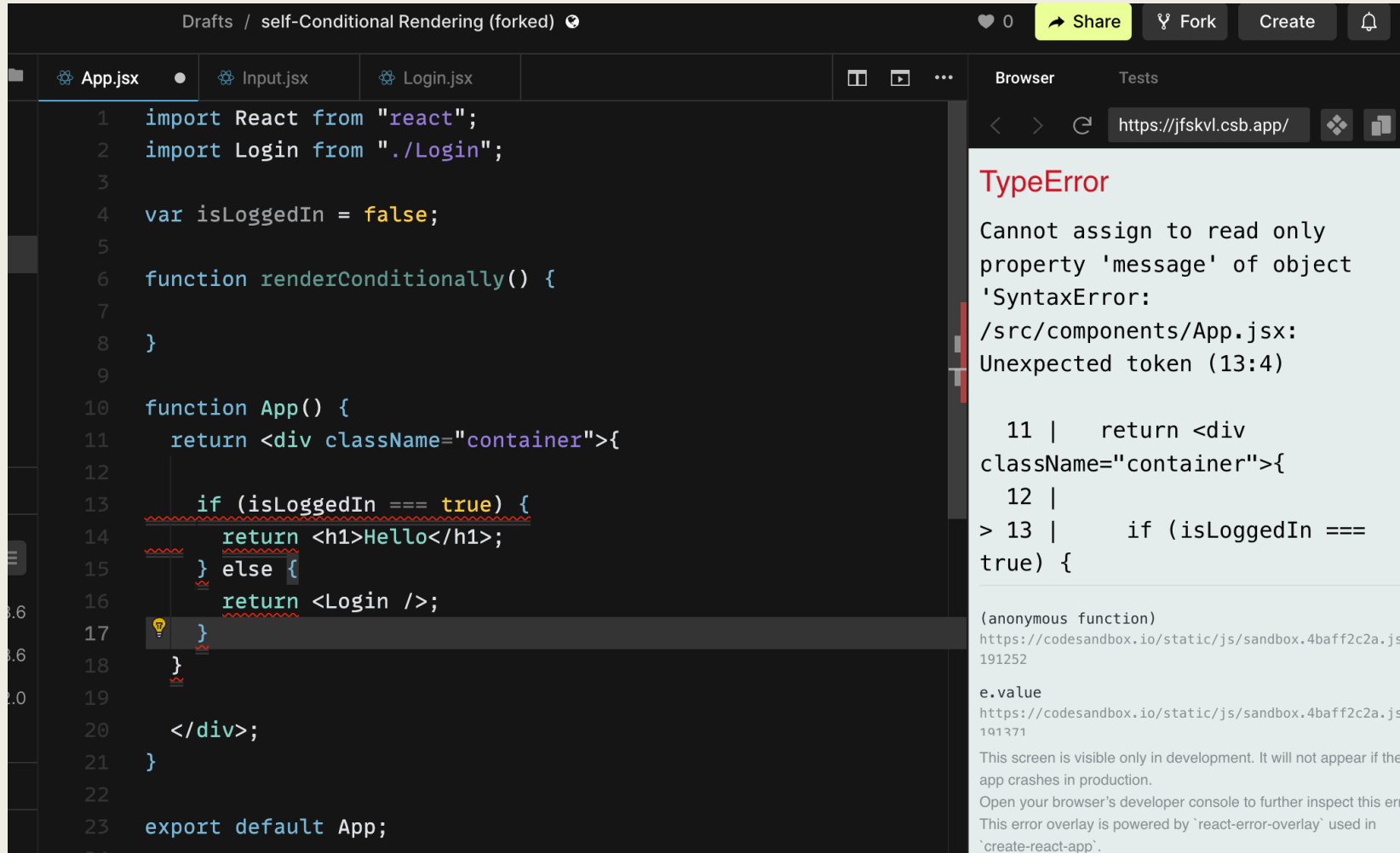
The image shows a code editor on the left and a browser window on the right. The code editor displays the following JavaScript code for an application:

```
1 import React from "react";
2 import Login from "./Login";
3
4 var isLoggedIn = false;
5
6 function renderConditionally() {
7   if (isLoggedIn === true) {
8     return <h1>Hello</h1>;
9   } else {
10    return <Login />;
11  }
12 }
13
14 function App() {
15   return <div className="container">{renderConditionally()}</div>;
16 }
17
18 export default App;
```

The browser window on the right shows the URL `https://jfskvl.csb.app` and displays a login form with the following elements:

- A teal header bar.
- A light blue input field labeled "Username".
- A light blue input field labeled "Password".
- A white button labeled "Login".

If we replace our function calling with if-else statement it results an error! Reason?



```
Drafts / self-Conditional Rendering (forked)
App.jsx Input.jsx Login.jsx
1 import React from "react";
2 import Login from "../Login";
3
4 var isLoggedIn = false;
5
6 function renderConditionally() {
7
8 }
9
10 function App() {
11   return <div className="container">{
12
13     if (isLoggedIn === true) {
14       return <h1>Hello</h1>;
15     } else {
16       return <Login />;
17     }
18   }
19
20 </div>;
21 }
22
23 export default App;
```

TypeError
Cannot assign to read only property 'message' of object 'SyntaxError:
'SyntaxError:
'/src/components/App.jsx:
Unexpected token (13:4)

```
11 |   return <div
12 |     className="container">{
> 13 |       if (isLoggedIn ===
    |       true) {
```

(anonymous function)
https://codesandbox.io/static/js/sandbox.4baff2c2a.js
191252
e.value
https://codesandbox.io/static/js/sandbox.4baff2c2a.js
101371
This screen is visible only in development. It will not appear if the app crashes in production.
Open your browser's developer console to further inspect this error.
This error overlay is powered by 'react-error-overlay' used in 'create-react-app'.

In JSX {} only takes expression not statement.

- Difference between JS expressions and Statements - <https://www.youtube.com/watch?v=WVyCrl1cHi8>
- Expressions yields value – `const y = getAnswer();`
- Statements are instructions that perform action but don't yield values – `if/else`, `while loop`, `for loop`.

Then how can we place our if/else statement in jsx {} ?

Ternary Operator

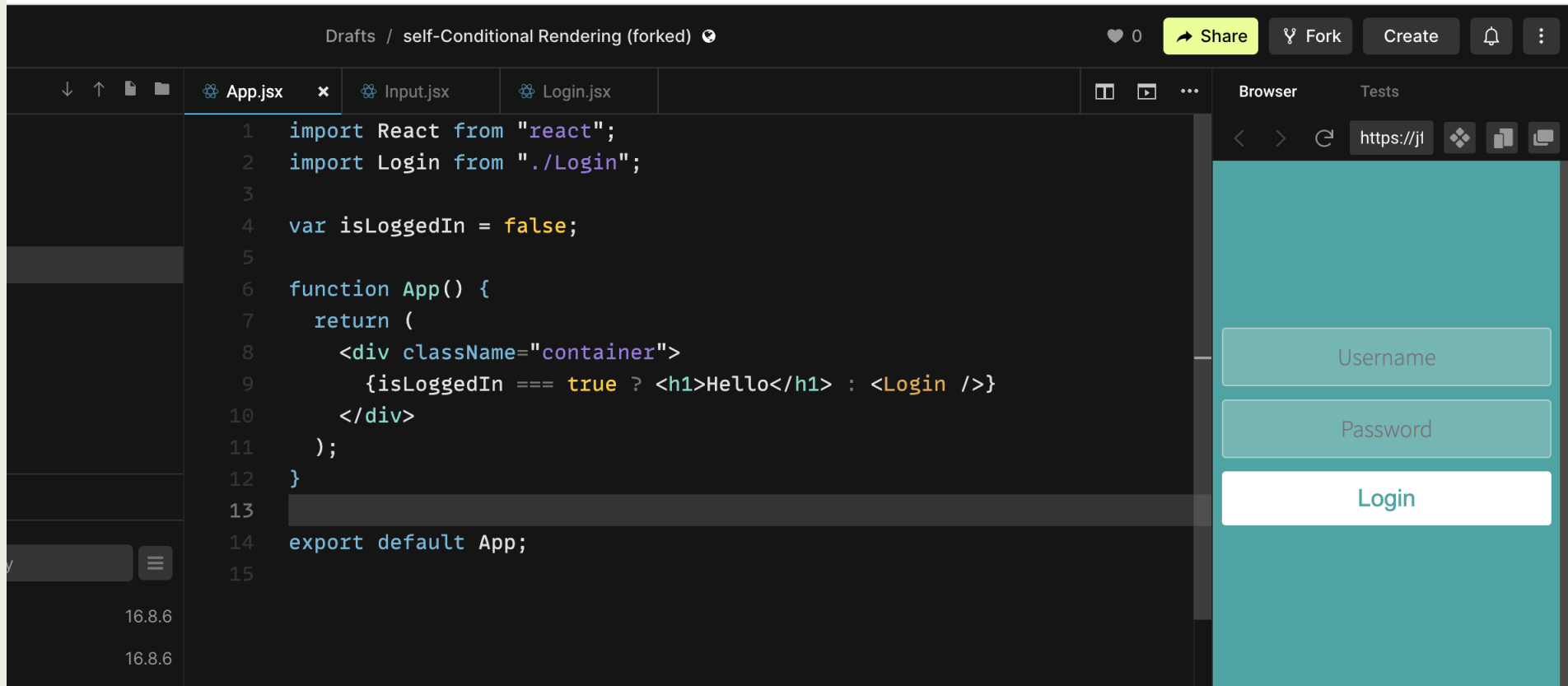
CONDITION ? DO IF TRUE : DO IF FALSE

isCloudy === true ? bringUmbrella() : bringSunscreen()

Each of them has three expressions.

MORE ON THIS - https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Conditional_operator

Implementing in react -



The screenshot shows a code editor with a dark theme. The top bar indicates the project is 'Drafts / self-Conditional Rendering (forked)'. The editor has three tabs: 'App.jsx', 'Input.jsx', and 'Login.jsx'. The 'App.jsx' tab is active, showing the following code:

```
1 import React from "react";
2 import Login from "../Login";
3
4 var isLoggedIn = false;
5
6 function App() {
7   return (
8     <div className="container">
9       {isLoggedIn === true ? <h1>Hello</h1> : <Login />}
10    </div>
11  );
12 }
13
14 export default App;
```

On the right side of the editor, there is a 'Browser' window showing a preview of the application. The preview has a teal background and contains three input fields: 'Username', 'Password', and a 'Login' button.

Assignment

- *Go here - <https://codesandbox.io/s/students-conditional-rendering-practice-forked-yz32rp?file=/src/index.js:158-462>*
- *//Without moving the `userIsRegistered` variable,*
- *//1. Show Login as the button text if `userIsRegistered` is true.*
- *//Show Register as the button text if `userIsRegistered` is false.*
- *//2. Only show the Confirm Password input if `userIsRegistered` is false.*
- *//Don't show it if `userIsRegistered` is true.*

