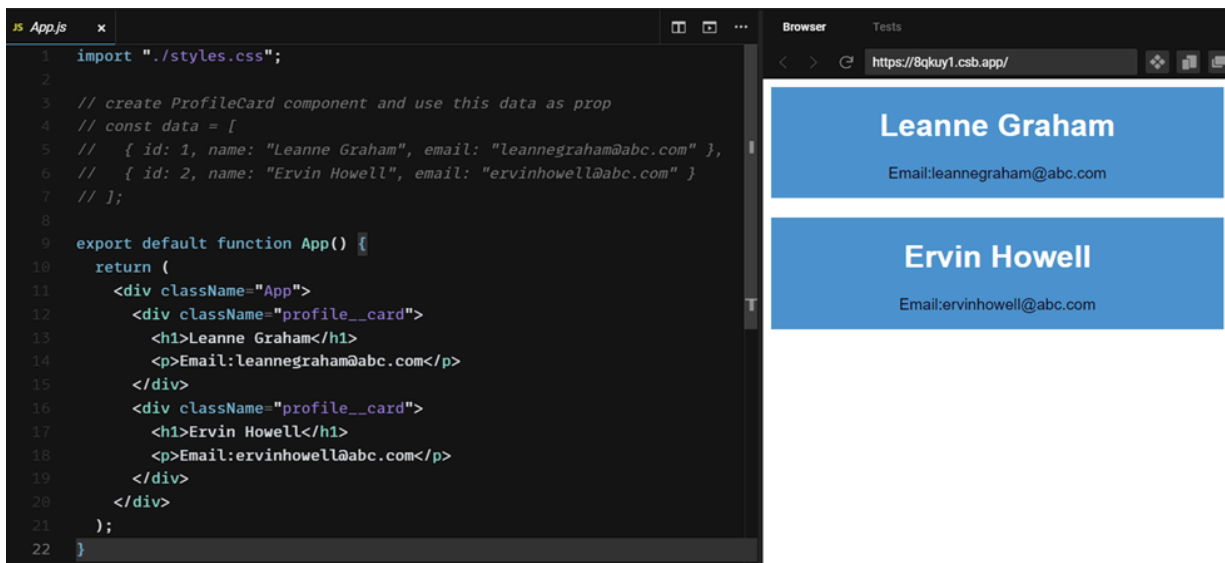


Assignments

1. Create a reusable component for this UI.

- Props
- Array.map()
- Key prop
- Template literal / String interpolation



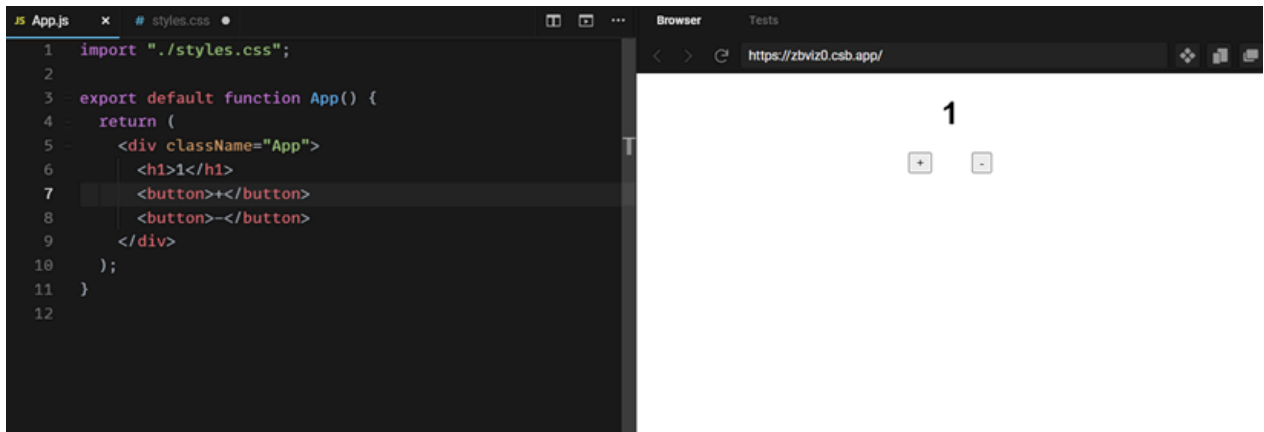
The screenshot shows a code editor on the left and a browser window on the right. The code editor displays the following JavaScript code:

```
1 import './styles.css';
2
3 // create ProfileCard component and use this data as prop
4 // const data = [
5 //   { id: 1, name: "Leanne Graham", email: "leannegraham@abc.com" },
6 //   { id: 2, name: "Ervin Howell", email: "ervinhowell@abc.com" }
7 // ];
8
9 export default function App() {
10   return (
11     <div className="App">
12       <div className="profile__card">
13         <h1>Leanne Graham</h1>
14         <p>Email:leannegraham@abc.com</p>
15       </div>
16       <div className="profile__card">
17         <h1>Ervin Howell</h1>
18         <p>Email:ervinhowell@abc.com</p>
19       </div>
20     </div>
21   );
22 }
```

The browser window shows the rendered output of the code, displaying two blue profile cards. The first card is for Leanne Graham with email Email:leannegraham@abc.com. The second card is for Ervin Howell with email Email:ervinhowell@abc.com.

2. Build a simple React App which increment/decrements count on-screen when buttons are clicked.

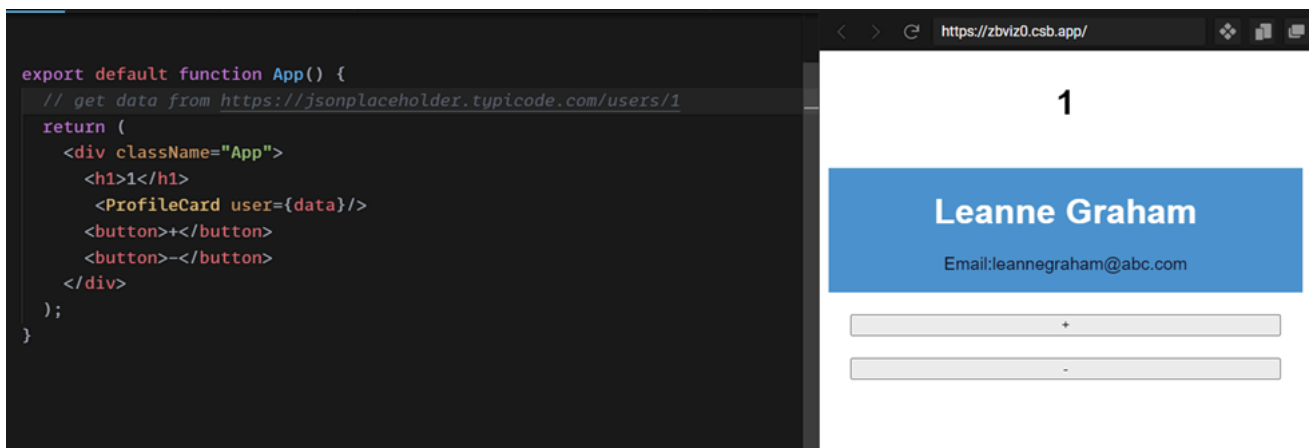
- State and useState
- onClick



3. Enhance the counter application: -

It should display user information related to count value.

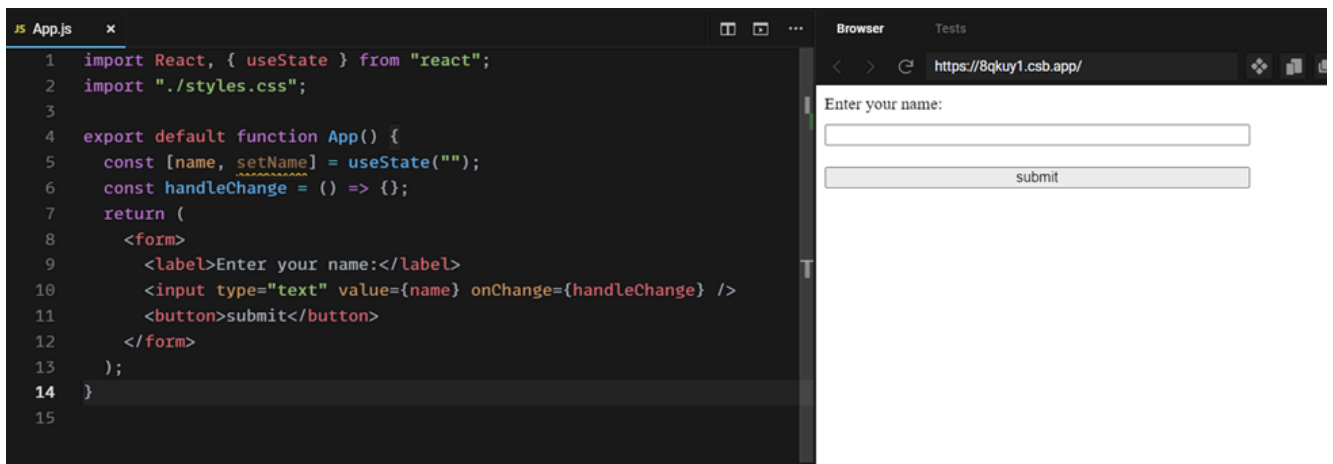
- Use public API eg : <https://jsonplaceholder.typicode.com/users/1>
- `useEffect`
- `axios/fetch`
- loading/success/error states



4. Forms

clicking the submit button should print the input value in console.

- onSubmit / onClick
- Form validation
- Focus the input element using useRef hook
- controlled and uncontrolled components



The screenshot shows a development environment with a code editor on the left and a browser on the right. The code editor displays the following JavaScript code for an application named App.js:

```
1 import React, { useState } from "react";
2 import "./styles.css";
3
4 export default function App() {
5   const [name, setName] = useState("");
6   const handleChange = () => {};
7   return (
8     <form>
9       <label>Enter your name:</label>
10      <input type="text" value={name} onChange={handleChange} />
11      <button>submit</button>
12    </form>
13  );
14 }
15
```

The browser window on the right shows the rendered application at the URL <https://8qkuy1.csb.app/>. It displays a form with the label "Enter your name:" followed by a text input field and a "submit" button.