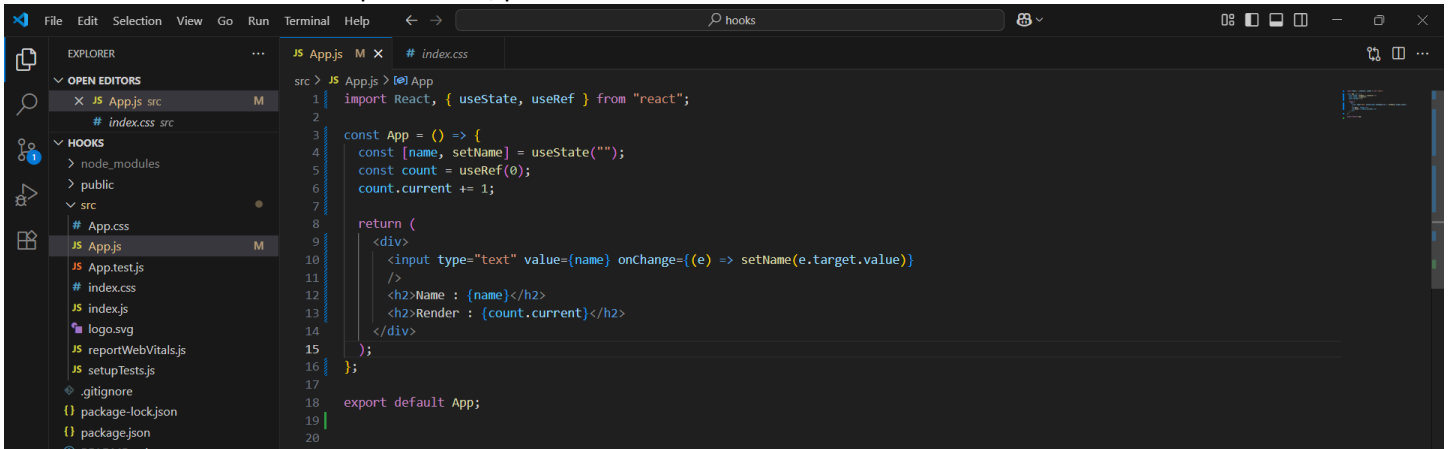


4. useRef — Re-render ছাড়াই value ধরে রাখা(allows to access DOM element, creating mutable variables which will not re-render the component)

কাজে: এমন data রাখা যা পরিবর্তিত হলেও re-render হয় না, বা DOM element access করা।

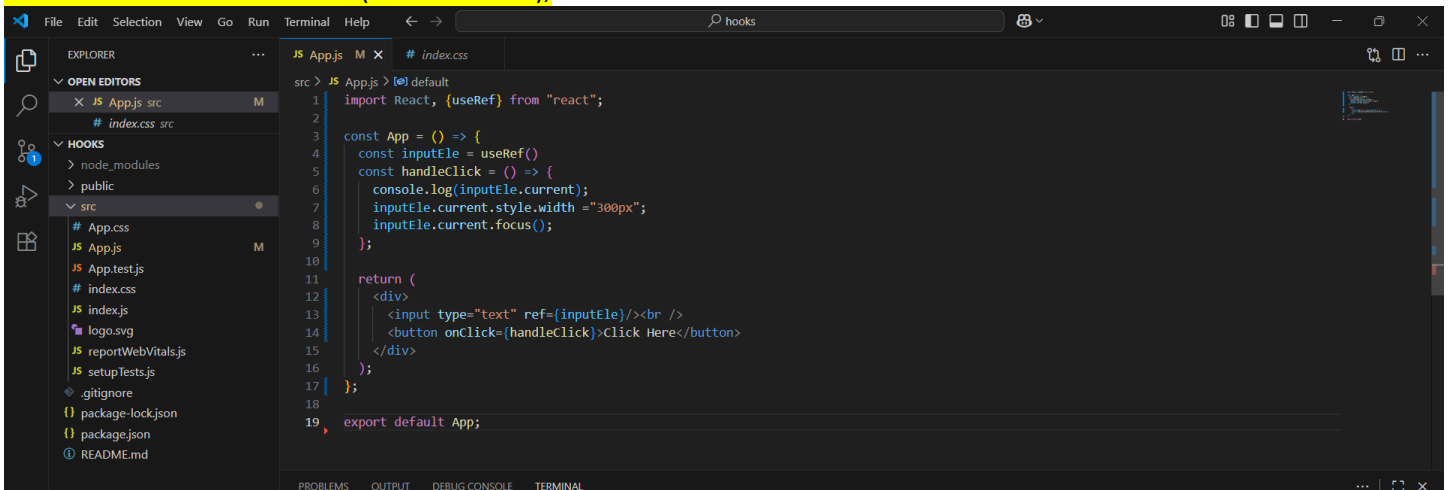
উদাহরণ: Focus করার জন্য input-এর ref, previous value track করা।



```
File Edit Selection View Go Run Terminal Help hooks
src > JS App.js M X # index.css
1 import React, { useState, useRef } from "react";
2
3 const App = () => {
4   const [name, setName] = useState("");
5   const count = useRef(0);
6   count.current += 1;
7
8   return (
9     <div>
10      <input type="text" value={name} onChange={(e) => setName(e.target.value)} />
11      </div>
12      <h2>Name : {name}</h2>
13      <h2>Render : {count.current}</h2>
14    </div>
15  );
16 };
17
18 export default App;
19
20
```

Syntax of useRef:

Const variableName = useRef(variableName);



```
File Edit Selection View Go Run Terminal Help hooks
src > JS App.js M X # index.css
1 import React, { useRef } from "react";
2
3 const App = () => {
4   const inputEle = useRef()
5   const handleClick = () => {
6     console.log(inputEle.current);
7     inputEle.current.style.width = "300px";
8     inputEle.current.focus();
9   };
10
11   return (
12     <div>
13       <input type="text" ref={inputEle}/><br />
14       <button onClick={handleClick}>Click Here</button>
15     </div>
16   );
17 };
18
19 export default App;
20
```

To update/access it:

variableName.current