

5. useReducer — Complex state manage করা

কাজ: State update-এর জন্য reducer function ও action ব্যবহার করা (Redux-এর মতো কিন্তু local)।

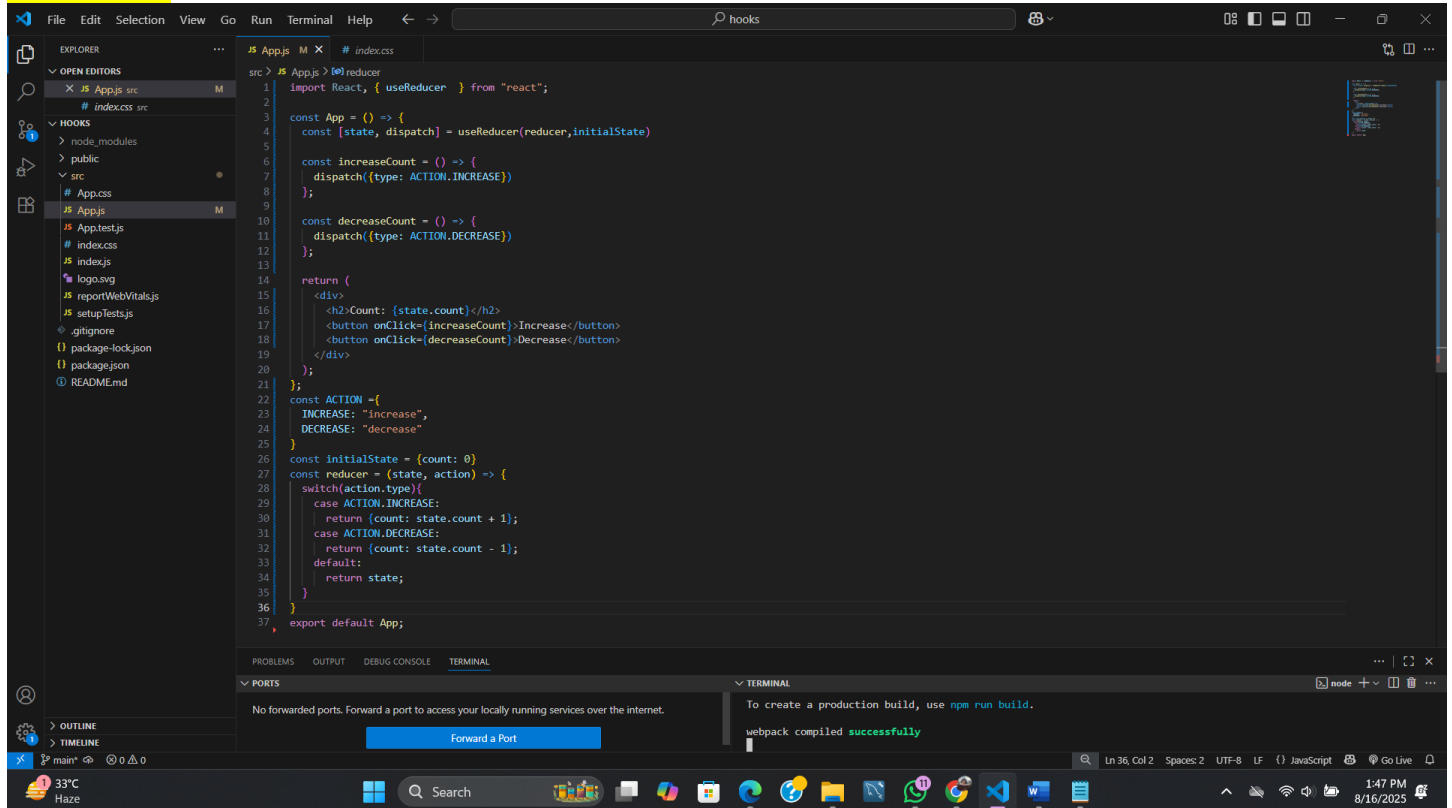
উদাহরণ: Shopping cart, multi-step form state

Syntax: `const [state, dispatch] = useReducer(reducer, initialState)`

Dispatch=> update per state

State => access current state

initialState=> object of all variable



The screenshot shows a VS Code editor with a project named 'Appjs'. The Explorer sidebar on the left shows the file structure, including 'Appjs.src', 'index.css', and 'App.css'. The main editor window displays the 'index.css' file, which contains the following code:

```
src > JS Appjs > index.css
1  import React, { useReducer } from "react";
2
3
4  const App = () => {
5    const [state, dispatch] = useReducer(reducer, initialState)
6
7    const increaseCount = () => {
8      dispatch({ type: ACTION.INCREASE })
9    };
10
11   const decreaseCount = () => {
12     dispatch({ type: ACTION.DECREASE })
13   };
14
15   return (
16     <div>
17       <h2>Count: {state.count}</h2>
18       <button onClick={increaseCount}>Increase</button>
19       <button onClick={decreaseCount}>Decrease</button>
20     </div>
21   );
22 };
23
24 const ACTION = {
25   INCREASE: "increase",
26   DECREASE: "decrease"
27 };
28
29 const initialState = { count: 0 };
30 const reducer = (state, action) => {
31   switch (action.type) {
32     case ACTION.INCREASE:
33       return { count: state.count + 1 };
34     case ACTION.DECREASE:
35       return { count: state.count - 1 };
36     default:
37       return state;
38   }
39 }
40
41 export default App;
```

The bottom of the editor shows the 'TERMINAL' panel with the following output:

```
node
To create a production build, use npm run build.
webpack compiled successfully
```