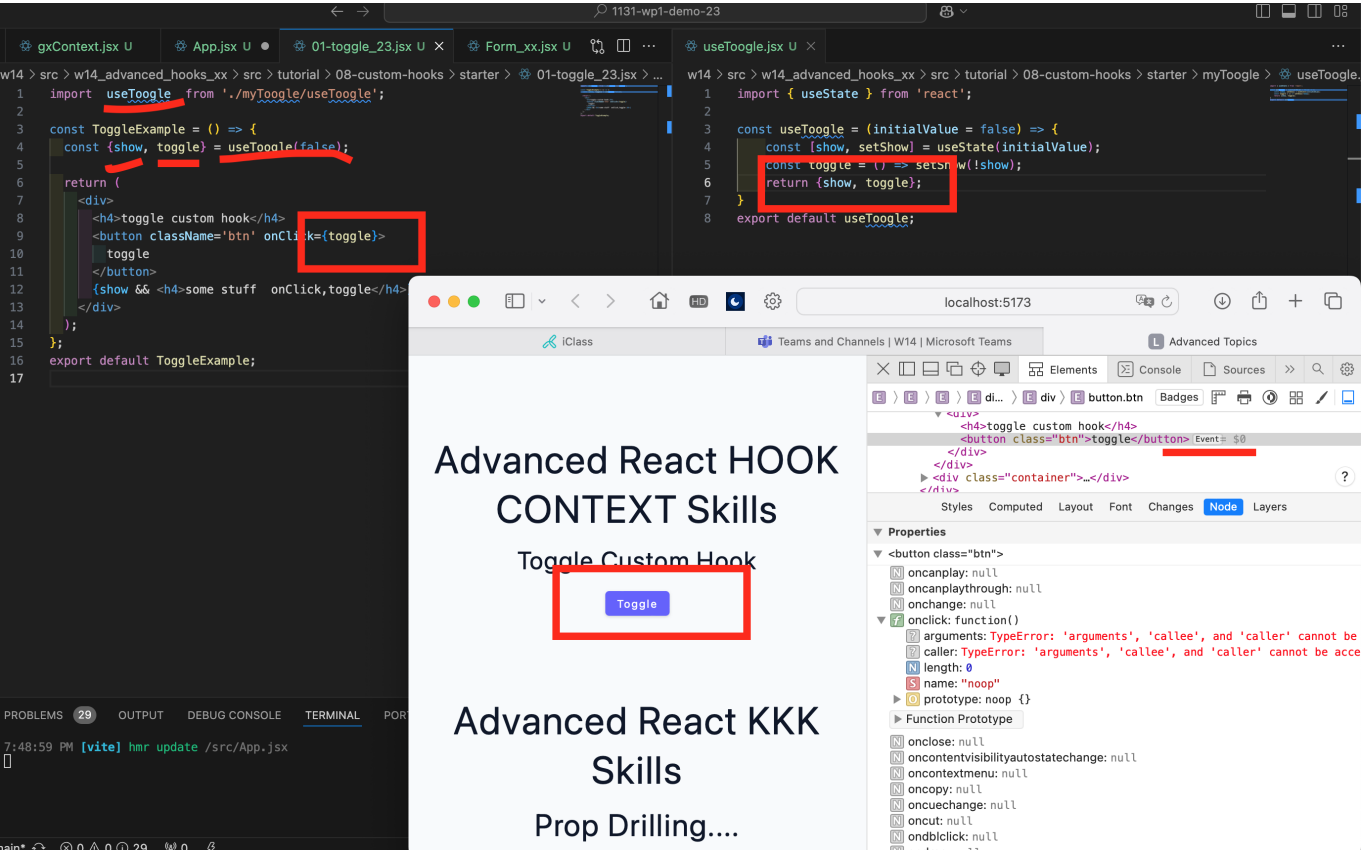


W14-P1: create a custom hook useToggle to toggle a button



W14-P2: Convert drop drilling code into context api code



Toggle Custom Hook

2. Ctx Api Skills

Prop Drilling....

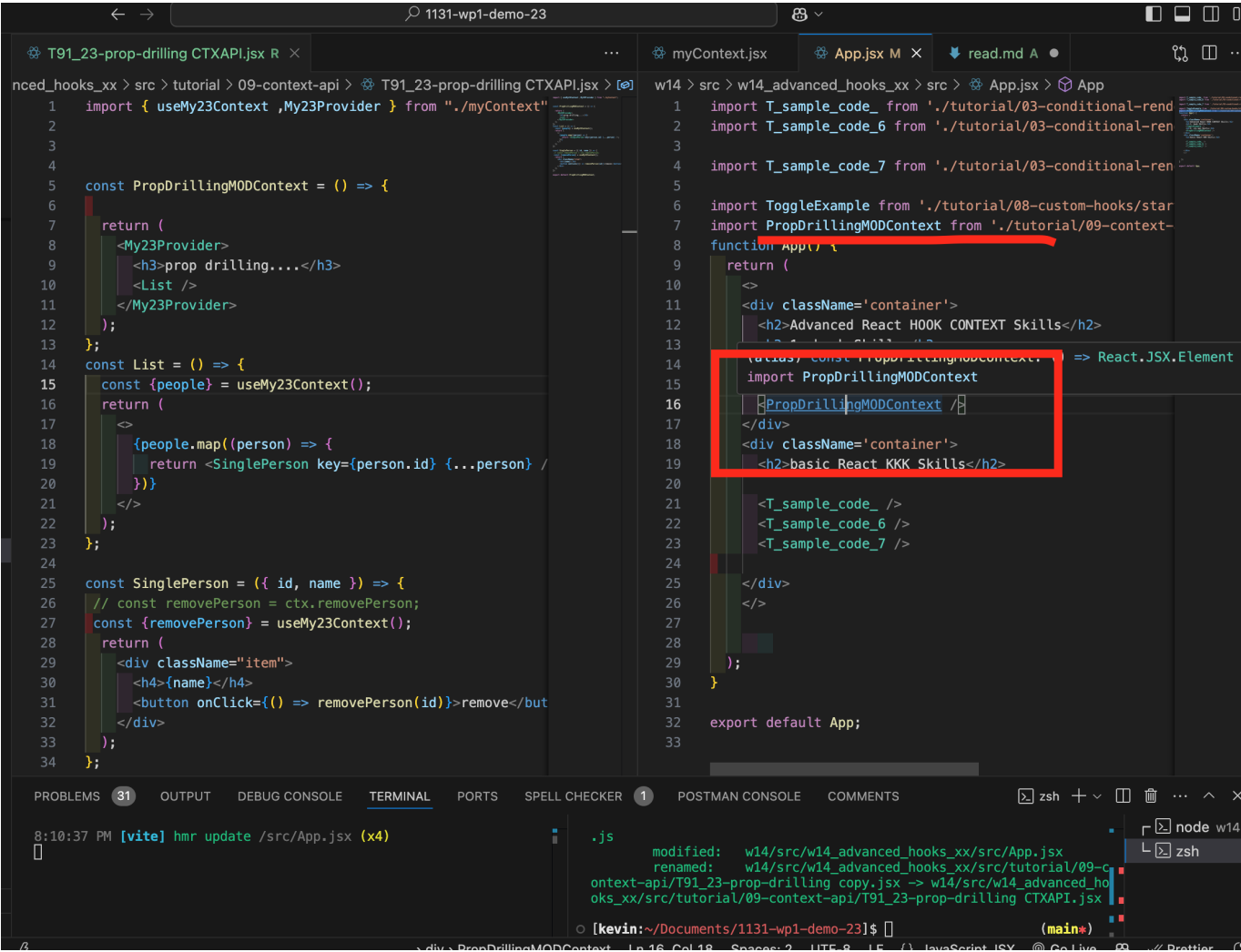
remove

Susan

remove

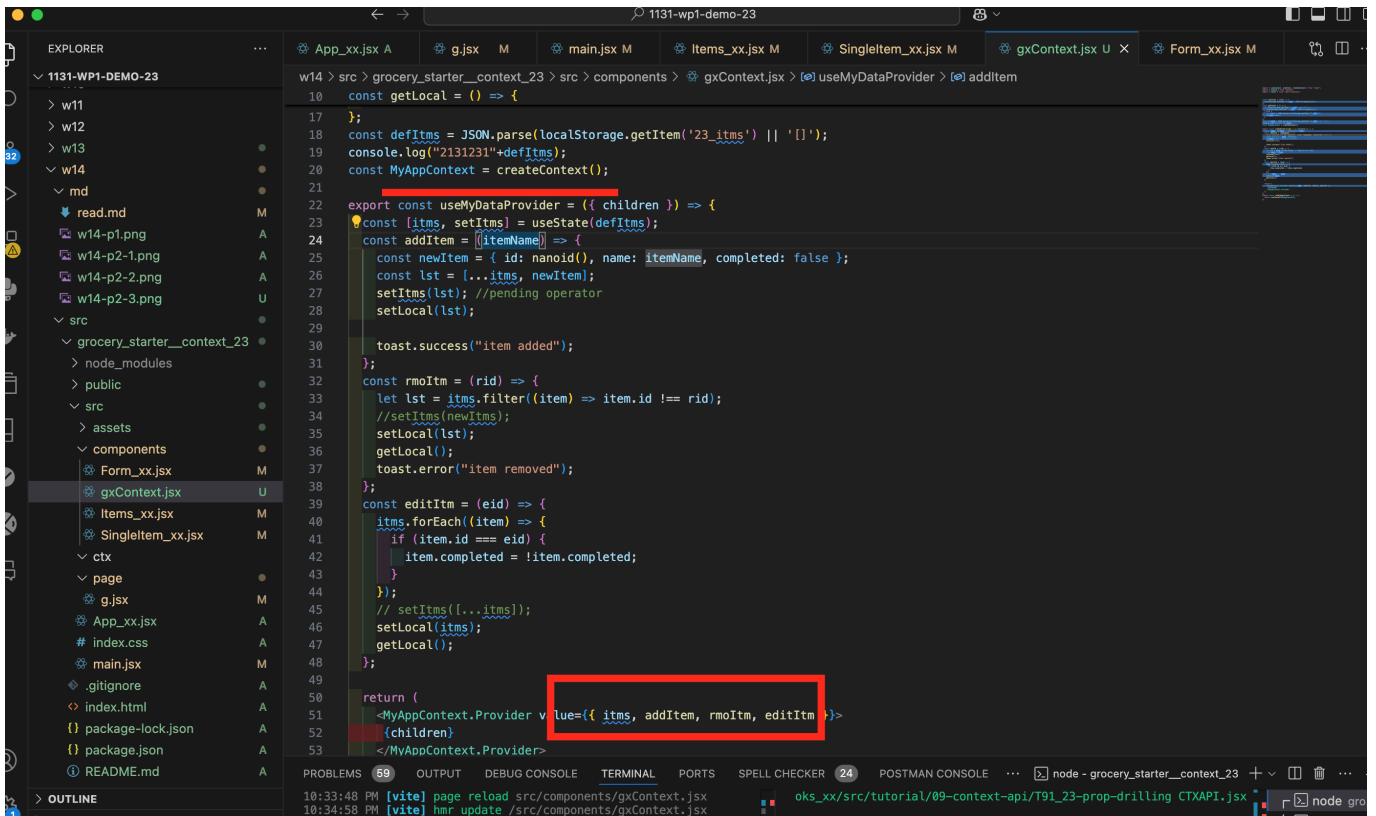
Basic React KKK Skills





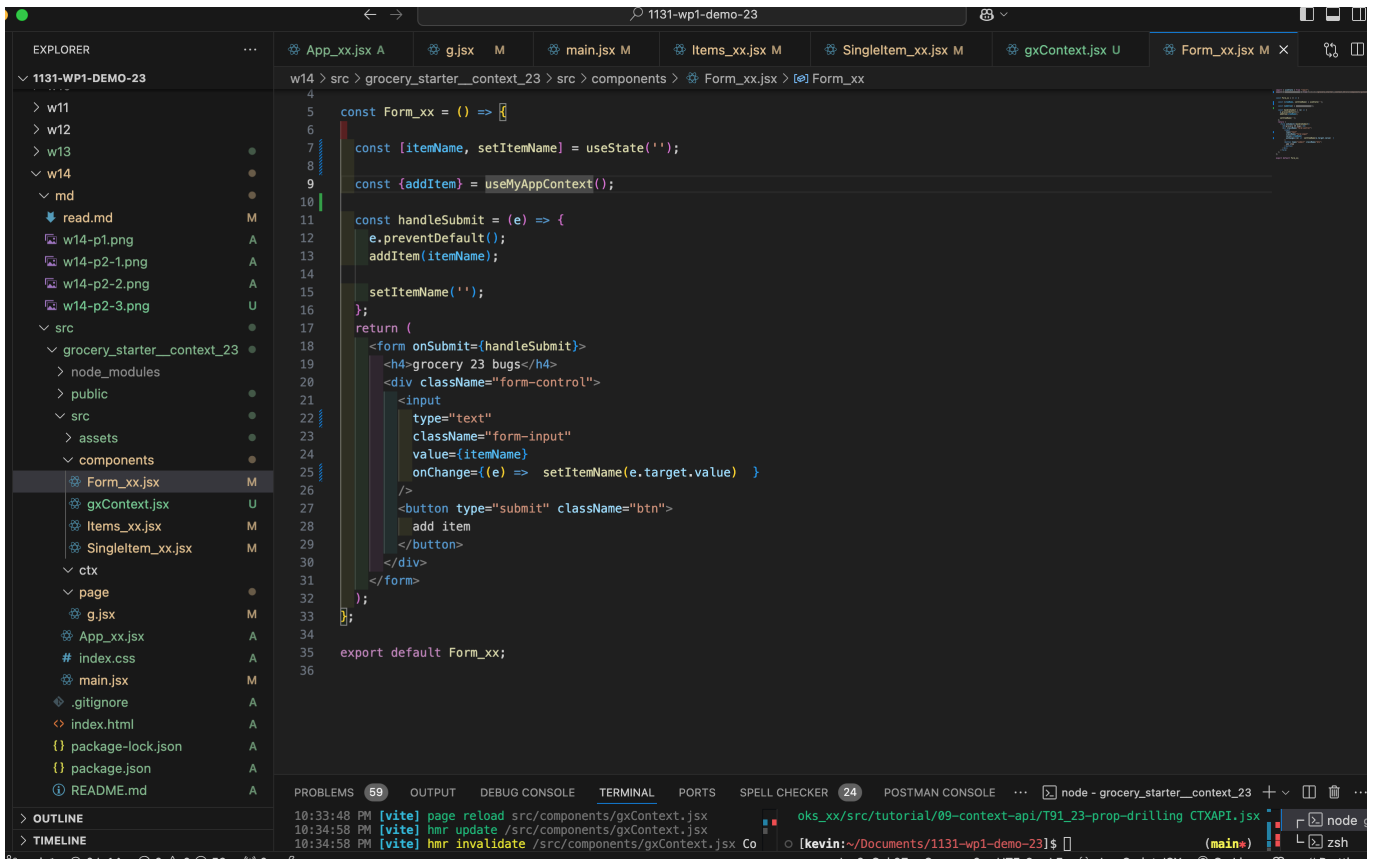
W14-P3: Convert w13 code into context api code

=> show from Chrome components and SingleItem_xx.jsx



```
10  const getLocal = () => {  
11  
12  };  
13  
14  const defItems = JSON.parse(localStorage.getItem('23_items') || '[]');  
15  console.log("2131231"+defItems);  
16  const MyAppContext = createContext();  
17  
18  export const useMyDataProvider = ({ children }) => {  
19    const [items, setItems] = useState(defItems);  
20    const addItem = (itemName) => {  
21      const newItem = { id: nanoid(), name: itemName, completed: false };  
22      const lst = [...items, newItem];  
23      setItems(lst); //pending operator  
24      setLocal(lst);  
25      toast.success("item added");  
26    };  
27    const rmoItem = (rid) => {  
28      let lst = items.filter((item) => item.id !== rid);  
29      //setItems(newItems);  
30      setLocal(lst);  
31      getLocal();  
32      toast.error("item removed");  
33    };  
34    const editItem = (eid) => {  
35      items.forEach((item) => {  
36        if (item.id === eid) {  
37          item.completed = !item.completed;  
38        }  
39      });  
40      // setItems([...items]);  
41      setLocal(items);  
42      getLocal();  
43    };  
44    return (  
45      <MyAppContext.Provider value={{ items, addItem, rmoItem, editItem }}>  
46        {children}  
47      </MyAppContext.Provider>  
48    );  
49  };  
50  
51  
52  
53
```

=> Show code (main.jsx, App_xx.jsx, Form_xx.jsx)



```
5  const Form_xx = () => {  
6  
7    const [itemName, setItemName] = useState('');  
8  
9    const {addItem} = useMyAppContext();  
10  
11    const handleSubmit = (e) => {  
12      e.preventDefault();  
13      addItem(itemName);  
14  
15      setItemName('');  
16    };  
17    return (  
18      <form onSubmit={handleSubmit}>  
19        <h4>grocery 23 bugs</h4>  
20        <div className="form-control">  
21          <input  
22            type="text"  
23            className="form-input"  
24            value={itemName}  
25            onChange={(e) => setItemName(e.target.value)} />  
26          </div>  
27          <button type="submit" className="btn">  
28            add item  
29          </button>  
30        </div>  
31      </form>  
32    );  
33  };  
34  
35  export default Form_xx;  
36
```

=> Show code (Items_xx.jsx, SingleItems_xx.jsx)

