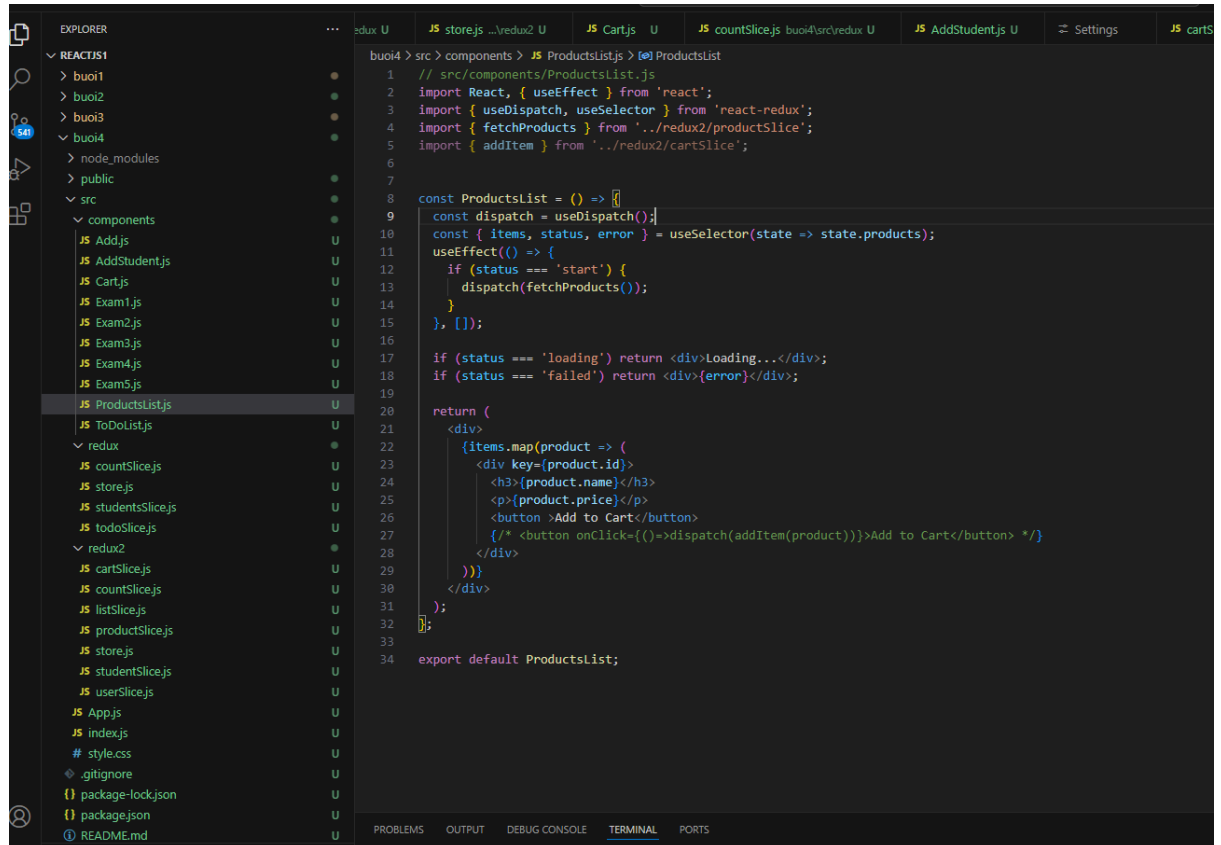


# Thực hiện chức năng giỏ hàng bằng redux toolkit

## B1: hiển thị danh sách sản phẩm



The screenshot shows a VS Code editor with a project structure on the left and a code editor on the right. The project structure includes a 'src' directory with 'components' and 'redux' subdirectories. The 'ProductsList.js' file is selected in the Explorer. The code in the editor implements a functional component that uses Redux Toolkit for state management and React for rendering.

```
1 // src/components/ProductsList.js
2 import React, { useEffect } from 'react';
3 import { useDispatch, useSelector } from 'react-redux';
4 import { fetchProducts } from '../redux2/productSlice';
5 import { addItem } from '../redux2/cartSlice';
6
7
8 const ProductsList = () => {
9   const dispatch = useDispatch();
10   const { items, status, error } = useSelector(state => state.products);
11   useEffect(() => {
12     if (status === 'start') {
13       dispatch(fetchProducts());
14     }
15   }, []);
16
17   if (status === 'loading') return <div>Loading...</div>;
18   if (status === 'failed') return <div>{error}</div>;
19
20   return (
21     <div>
22       {items.map(product => (
23         <div key={product.id}>
24           <h3>{product.name}</h3>
25           <p>{product.price}</p>
26           <button>Add to Cart</button>
27           { /* <button onClick={()=>dispatch(addItem(product))}>Add to Cart</button> */ }
28         </div>
29       ))}
30     </div>
31   );
32 };
33
34 export default ProductsList;
```

File Edit Selection View Go Run Terminal Help

reactjs1

EXPLORER

REACTJS1

> buoi1

> buoi2

> buoi3

> buoi4

> node\_modules

> public

> src

> components

JS Add.js

JS AddStudent.js

JS Cart.js

JS Exam1.js

JS Exam2.js

JS Exam3.js

JS Exam4.js

JS Exam5.js

JS ProductsList.js

JS ToDoList.js

> redux

JS countSlice.js

JS store.js

JS studentsSlice.js

JS todoSlice.js

> redux2

JS cartSlice.js

JS countSlice.js

JS listSlice.js

JS productSlice.js

JS store.js

JS studentSlice.js

JS userSlice.js

JS App.js

JS index.js

# style.css

.gitignore

( ) package-lock.json

( ) package.json

① README.md

> node\_modules

redux U

JS store.js ...redux2 U

JS Cart.js U

JS countSlice.js buoi4/src/redux U

JS AddStudent.js U

Settings

JS cartSlice.js U

buoi4 > src > redux2 > JS productSlice.js > ...

1 import { createSlice, createAsyncThunk } from '@reduxjs/toolkit';

2 import axios from 'axios';

3

4 // Define the initial state for products

5 const initialState = {

6 items: [],

7 status: 'start',

8 error: null,

9 };

10 // axios.get(url)

11 // .then()

12 // .catch

13 // const meo = async()=>{

14 // await/các phép toán xử lý bất đồng bộ: Promise

15 // }

16 // Define an async thunk for fetching products from an API

17 //name/action\_creator

18 export const fetchProducts = createAsyncThunk('products/fetchProducts', async () => {

19 const response = await axios.get('https://66a077917053166bcabb8106.mockapi.io/student');

20 return response.data;

21 });

22

23 // Create the products slice

24 const productsSlice = createSlice({

25 name: 'products',

26 initialState,

27 reducers: {},

28 extraReducers: (builder) => {

29 builder

30 .addCase(fetchProducts.pending, (state) => {

31 state.status = 'loading';

32 })

33 .addCase(fetchProducts.fulfilled, (state, action) => {

34 state.status = 'succeeded';

35 state.items = action.payload;

36 })

37 .addCase(fetchProducts.rejected, (state, action) => {

38 state.status = 'failed';

39 state.error = action.error.message;

40 });

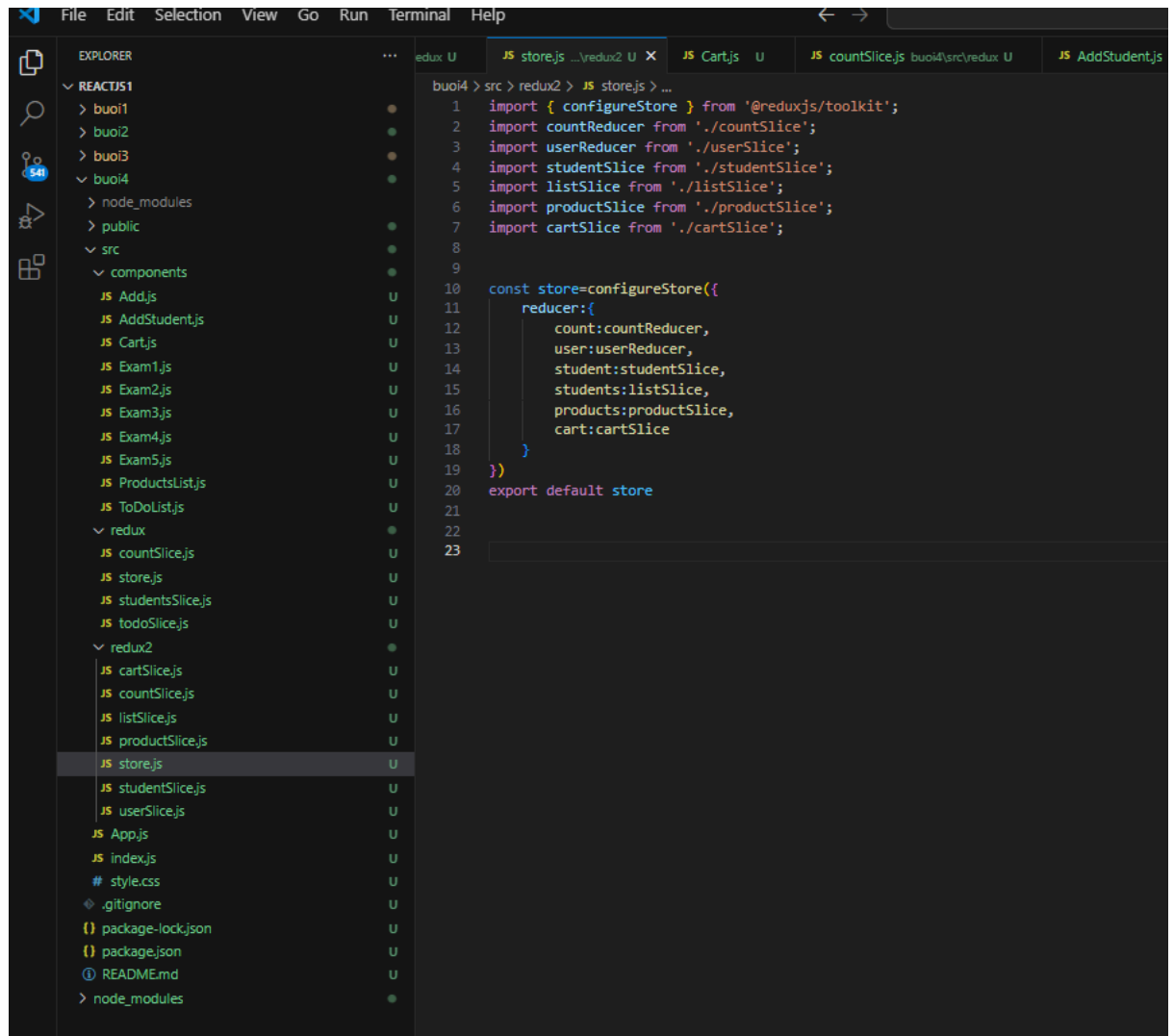
41 },

42 });

43

44 // Export the async thunk and the slice reducer

45 export default productsSlice.reducer;



-Trong Redux Toolkit, `extraReducers` là một phần quan trọng trong việc xử lý các action bên ngoài slice , đặc biệt là khi bạn đang sử dụng các thunks (như `createAsyncThunk`) để thực hiện các hoạt động không đồng bộ (asynchronous operations).

**`builder.addCase(fetchProducts.pending, ...):`**

- Xử lý trạng thái khi thunk `fetchProducts` bắt đầu thực hiện (tức là khi request đang chờ). Ở đây, bạn có thể đặt `status` thành `'loading'` để biểu thị rằng dữ liệu đang được tải.

**`builder.addCase(fetchProducts.fulfilled, ...):`**

- Xử lý khi thunk `fetchProducts` hoàn thành thành công. `action.payload` chứa dữ liệu trả về từ API. Bạn có thể cập nhật trạng thái và lưu trữ dữ liệu sản phẩm vào `items`.

**`builder.addCase(fetchProducts.rejected, ...):`**

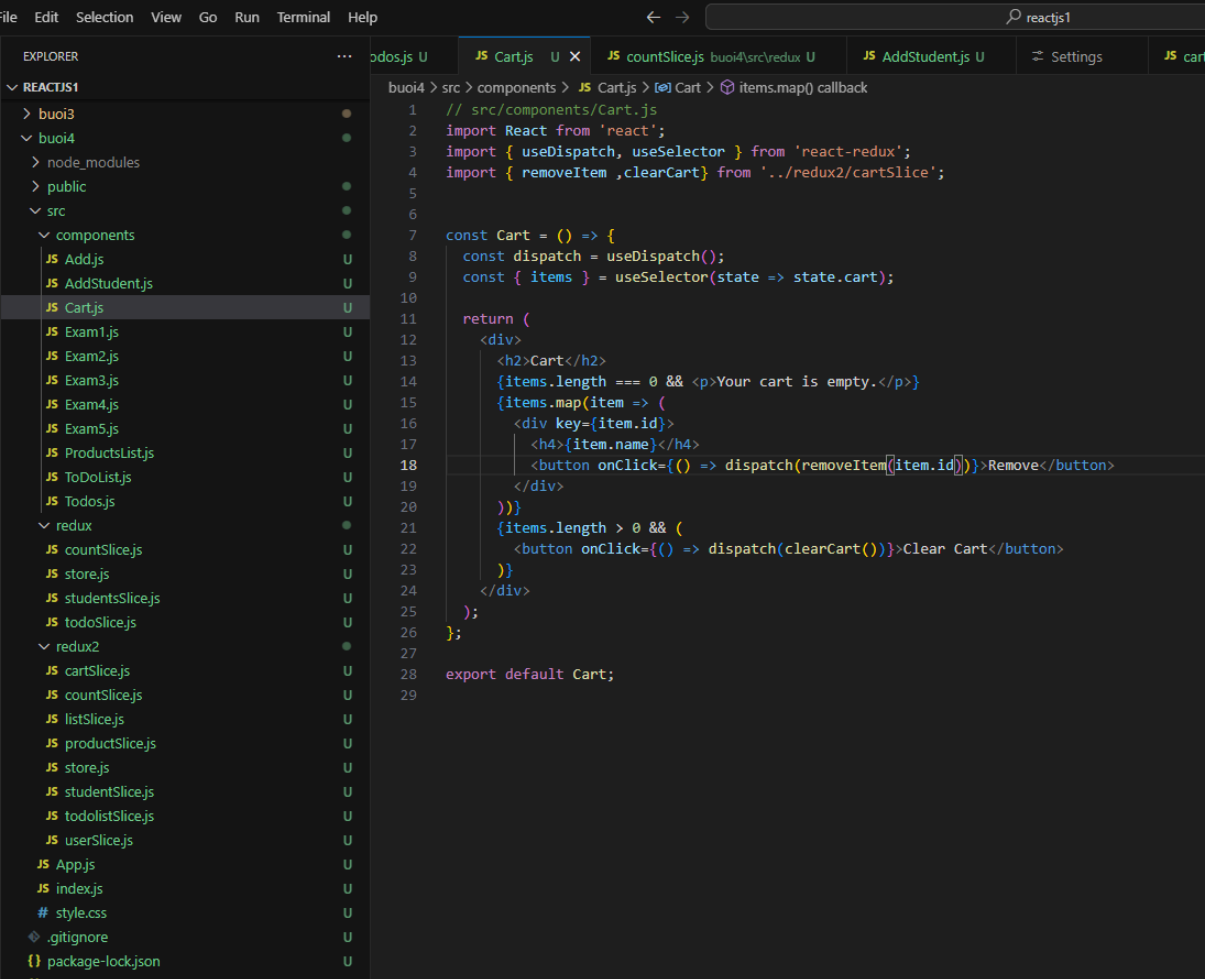
- Xử lý khi thunk `fetchProducts` bị từ chối hoặc thất bại. `action.error.message` chứa thông tin lỗi. Bạn có thể cập nhật trạng thái và lưu thông tin lỗi vào `error`.

Khi bạn sử dụng `createAsyncThunk`, Redux Toolkit tự động tạo ra ba loại action:

- **Pending:** `sliceName/asyncActionName/pending`
- **Fulfilled:** `sliceName/asyncActionName/fulfilled`
- **Rejected:** `sliceName/asyncActionName/rejected`

- `async` và `await` là các từ khóa trong JavaScript dùng để làm việc với các phép toán bất đồng bộ (asynchronous) theo cách đồng bộ hơn. Chúng giúp làm

cho mã bất đồng bộ dễ đọc hơn và dễ hiểu hơn bằng cách ẩn đi các chi tiết của Promise và `then/catch`.



The screenshot shows a VS Code editor with a project named 'reactjs1'. The Explorer panel on the left shows the file structure, with the 'src' directory expanded to show 'components'. The 'Cart.js' file is selected. The main editor displays the code for 'Cart.js', which is a functional component using React hooks and Redux. The code includes imports for React, useReducer, useSelector, and useDispatch. It defines a stateful component 'Cart' that uses 'useReducer' to manage a list of items and 'useSelector' to access the state. The component renders a list of items and a 'Remove' button. The code is as follows:

```
1 // src/components/Cart.js
2 import React from 'react';
3 import { useReducer, useSelector, useDispatch } from 'react-redux';
4 import { removeItem, clearCart } from '../redux2/cartSlice';
5
6
7 const Cart = () => {
8   const dispatch = useDispatch();
9   const { items } = useSelector(state => state.cart);
10
11   return (
12     <div>
13       <h2>Cart</h2>
14       {items.length === 0 && <p>Your cart is empty.</p>}
15       {items.map(item => (
16         <div key={item.id}>
17           <h4>{item.name}</h4>
18           <button onClick={() => dispatch(removeItem(item.id))}>Remove</button>
19         </div>
20       ))}
21       {items.length > 0 && (
22         <button onClick={() => dispatch(clearCart())}>Clear Cart</button>
23       )}
24     </div>
25   );
26 };
27
28 export default Cart;
```

```
EXPLORER
  REACTJS1
    > buoi3
    > buoi4
    > node_modules
    > public
    > src
      > components
        JS Add.js
        JS AddStudent.js
        JS Cart.js
        JS Exam1.js
        JS Exam2.js
        JS Exam3.js
        JS Exam4.js
        JS Exam5.js
        JS ProductList.js
        JS ToDoList.js
        JS Todos.js
      > redux
        JS countSlice.js
        JS store.js
        JS studentSlice.js
        JS todoSlice.js
      > redux2
        JS cartSlice.js
        JS countSlice.js
        JS listSlice.js
        JS productSlice.js
        JS store.js
        JS studentSlice.js
        JS todoListSlice.js
        JS userSlice.js
      JS App.js
      JS index.js
      # style.css
      .gitignore
      package-lock.json
      package.json
      README.md
      > node_modules

buoi4 > src > redux2 > JS cartSlice.js > [0] cartSlice > reducers > removeItem > state.items.filter() callback
1 import { createSlice } from '@reduxjs/toolkit';
2
3 const cartSlice = createSlice({
4   name: 'cart',
5   initialState: {
6     items: [],
7     status: 'idle',
8     error: null,
9   },
10  reducers: {
11    addItem: (state, action) => {
12      state.items.push(action.payload);
13    },
14    removeItem: (state, action) => {
15      state.items = state.items.filter(item => item.id !== action.payload);
16    },
17    clearCart: (state) => {
18      state.items = [];
19    },
20    setloading: (state) => {
21      state.status = 'loading';
22    },
23    setError: (state, action) => {
24      state.status = 'failed';
25      state.error = action.payload;
26    },
27    setloaded: (state) => {
28      state.status = 'succeeded';
29    },
30  },
31 });
32
33 export const { addItem, removeItem, clearCart, setloading, setError, setloaded } = cartSlice.actions;
34 export default cartSlice.reducer;
```

```
File Edit Selection View Go Run Terminal Help
reactjs1

EXPLORER
  REACTJS1
    > buoi3
    > buoi4
    > node_modules
    > public
    > src
      > components
        JS Add.js
        JS AddStudent.js
        JS Cart.js
        JS Exam1.js
        JS Exam2.js
        JS Exam3.js
        JS Exam4.js
        JS Exam5.js
        JS ProductList.js
        JS ToDoList.js
        JS Todos.js
      > redux
        JS countSlice.js
        JS store.js
        JS studentSlice.js
        JS todoSlice.js
      > redux2
        JS cartSlice.js
        JS countSlice.js
        JS listSlice.js
        JS productSlice.js
        JS store.js
        JS studentSlice.js
        JS todoListSlice.js
        JS userSlice.js

buoi4 > src > components > JS ProductList.js > [0] ProductList
1 // src/components/ProductList.js
2 import React, { useEffect } from 'react';
3 import { useDispatch, useSelector } from 'react-redux';
4 import { fetchProducts } from '../redux2/productSlice';
5 import { addItem } from '../redux2/cartSlice';
6
7
8 const ProductList = () => {
9   const dispatch = useDispatch();
10  const { items, status, error } = useSelector(state => state.products);
11  useEffect(() => {
12    if (status === 'start') {
13      dispatch(fetchProducts());
14    }
15  }, []);
16
17  if (status === 'loading') return <div>loading...</div>;
18  if (status === 'failed') return <div>{error}</div>;
19
20  return (
21    <div>
22      {items.map(product => (
23        <div key={product.id}>
24          <h3>{product.name}</h3>
25          <p>{product.price}</p>
26          <button onClick={() => dispatch(addItem(product))}>Add to Cart</button>
27        </div>
28      ))}
29    </div>
30  );
31 };
32
33 export default ProductList;
```

Ví dụ có 31 bản ghi

limit=5,

=> totalpage= $31/5 = 7$  page

page 1: [1,5]

page 2:[6,10]

...

page 6:[26,30]

page 7 :[31]

```
File Edit Selection View Go Run Terminal Help
reactjs1

EXPLORER
  REACTJS1
    > buoi1
    > buoi2
    > buoi3
    > buoi4
    > node_modules
    > public
    > src
      components
        Add.js
        AddStudent.js
        Cart.js
        Cats.js
        Exam1.js
        Exam2.js
        Exam3.js
        Exam4.js
        Exam5.js
        ProductsList.js
        ToDoList.js
        Todos.js
      redux
        countSlice.js
        store.js
        studentsSlice.js
        todoSlice.js
      redux2
        cartSlice.js
        catsSlice.js
        countSlice.js
        listSlice.js
        productSlice.js
        store.js
        studentSlice.js
        todolistSlice.js
        userSlice.js
      App.js
      index.js
      style.css
    .gitignore
    package-lock.json
    package.json
    README.md
    node_modules

Cats.js
1 import React, { useEffect } from 'react';
2 import { useDispatch, useSelector } from 'react-redux';
3 import { Button, Container, Input, Table } from 'reactstrap';
4 import { fetchCats, deleteCat, addNewCat, reCheckCat } from '../redux2/catsSlice';
5
6
7 export default function Cats() {
8   const dispatch = useDispatch();
9   const { cats } = useSelector((state) => state.cats)
10   useEffect(() => {
11     dispatch(fetchCats())
12   }, [ ])
13   const handle_delete=(id)=>{
14     dispatch(deleteCat(id))
15   }
16   const handle_add=(cat)=>{
17     dispatch(addNewCat(cat))
18   }
19   const handle_rechecked =(cat)=>{
20     dispatch(reCheckCat(cat))
21   }
22   return (
23     <div>
24       <Container>
25         <Input placeholder='Name' onKeyDown=(e)=>{
26           if(e.key==="Enter"){
27             handle_add({name:"lê Văn mèo",checked:false})
28           }
29         } />
30       <Table hover>
31         <thead>
32           <tr>...
33         </thead>
34         <tbody>
35           {
36             cats.map((item, index) =>
37               <tr key={index}>
38                 <td scope="row">
39                   {item.id}
40                 </td>
41                 <td className={item.checked?"cat-name active":"cat-name"}
42                   onClick={()=>handle_rechecked(item)}
43                 </td>
44                 <td>
45                   {item.name}
46                 </td>
47                 <td>
48                   {item.checked?"true":"false"}
49                 </td>
50                 <td>
51                   <Button onClick={()=>handle_delete(item.id)}>Delete</Button>
52                 </td>
53               </tr>
54             )
55           }
56         </tbody>
57       </Table>
58     </div>
59   )
60 }
```





- Redux + reactjs( Không API)
- html,css,es6
- Nội dung bài thi: 150 phút ( sau 150 phút, học viên được yêu cầu dừng tất cả các hoạt động về làm bài thi)
- Upload source code và deploy website: 30 phút
- Gửi 2 link: link source code và link web domain