

p1

The screenshot shows a development environment with a code editor on the left and a browser window on the right. The code editor displays the `context_78.js` file, which defines a context provider and a reducer. The browser window shows the rendered application, titled "UseReducer", displaying a shopping cart with three items: Samsung Galaxy S7, Google Pixel, and Xiaomi Redmi Note 2. The total amount is \$0.00.

Code Editor (context_78.js):

```
import React, { useState, useContext, useReducer, useEffect } from 'react';
import cartItems_78 from './data';
import reducer_78 from './reducer_78';

const AppContext_78 = React.createContext();

const AppProvider_78 = ({ children }) => {
  const [cart, setCart] = useState(cartItems_78);

  return (
    <AppContext_78.Provider
      value={{
        cart,
      }}
      >
      {children}
    </AppContext_78.Provider>
  );
};

// make sure use
export const useGlobalContext = () => {
  return useContext(AppContext_78);
};




export { AppContext_78, AppProvider_78 };

```

Browser Window (localhost:3000):

UseReducer

YOUR BAG

	Samsung Galaxy S7 \$599.99 remove	1
	Google Pixel \$499.99 remove	1
	Xiaomi Redmi Note 2 \$699.99 remove	1
Total		\$0.00

p2

The screenshot shows a development environment with a code editor on the left and a browser window on the right. The code editor displays the `CartContainer_78.js` file, which defines a component that renders the shopping cart. The browser window shows the rendered application, titled "UseReducer", displaying a shopping cart with three items: Samsung Galaxy S7, Google Pixel, and Xiaomi Redmi Note 2. The total amount is 0. A "CLEAR CART" button is visible at the bottom.

Code Editor (CartContainer_78.js):




```
return (
  <section className='cart'>
    <div>
      <h2>your bag</h2>
    </div>
    <div>
      {cart.map((item) => {
        return <CartItem_78 key={item.id} {...item} />
      })}
    </div>
    <div>
      <div className='cart-total'>
        <h4>
          total <span>{total}</span>
        </h4>
      </div>
      <button
        className='btn clear-btn'
        onClick={() => console.log('clear cart')}
      >
        clear cart
      </button>
    </div>
  </section>
);

```

Browser Window (localhost:3000):

UseReducer

YOUR BAG

	Samsung Galaxy S7 \$599.99 remove	1
	Google Pixel \$499.99 remove	1
	Xiaomi Redmi Note 2 \$699.99 remove	1
Total		0

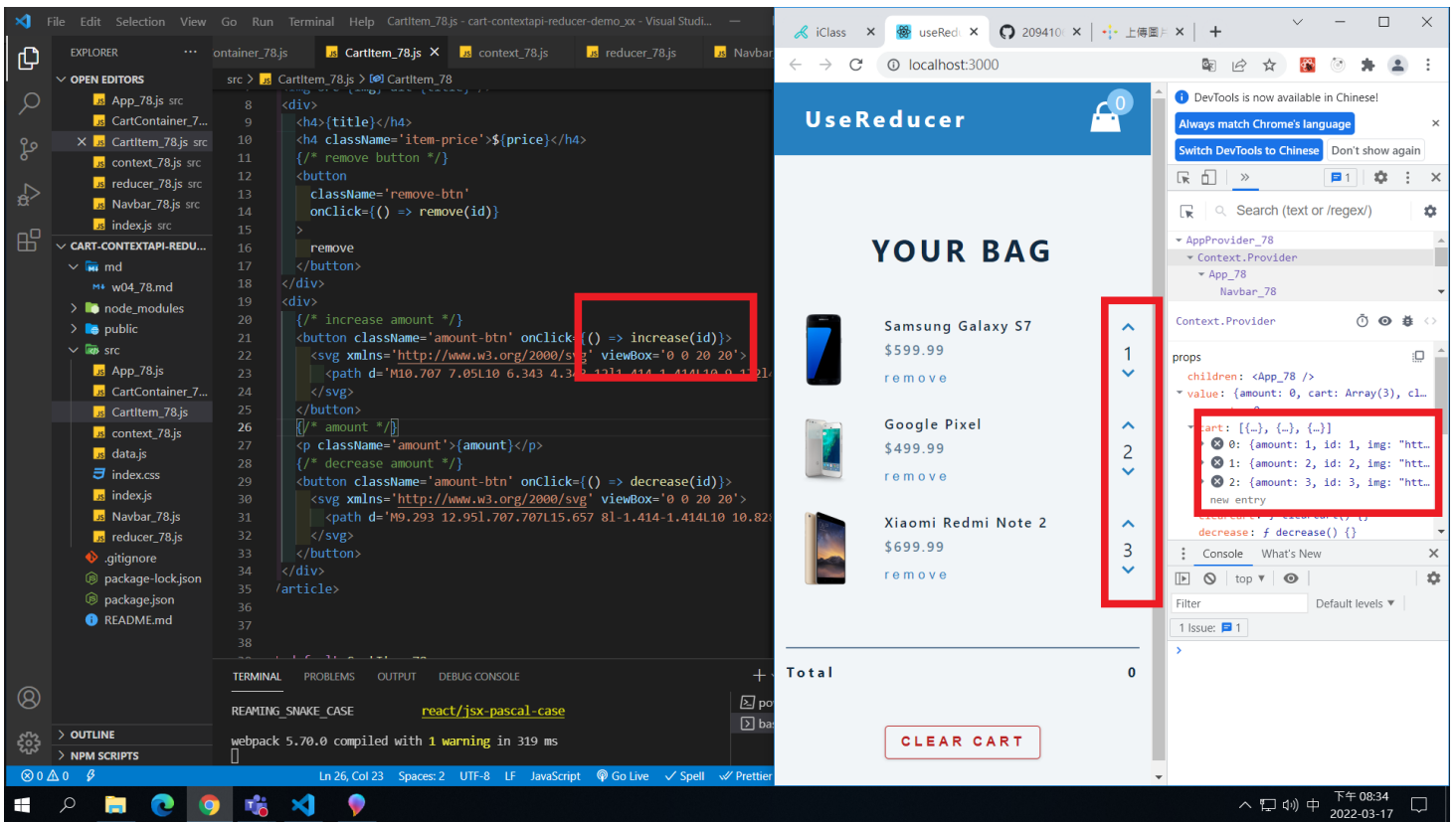
[CLEAR CART](#)

p3

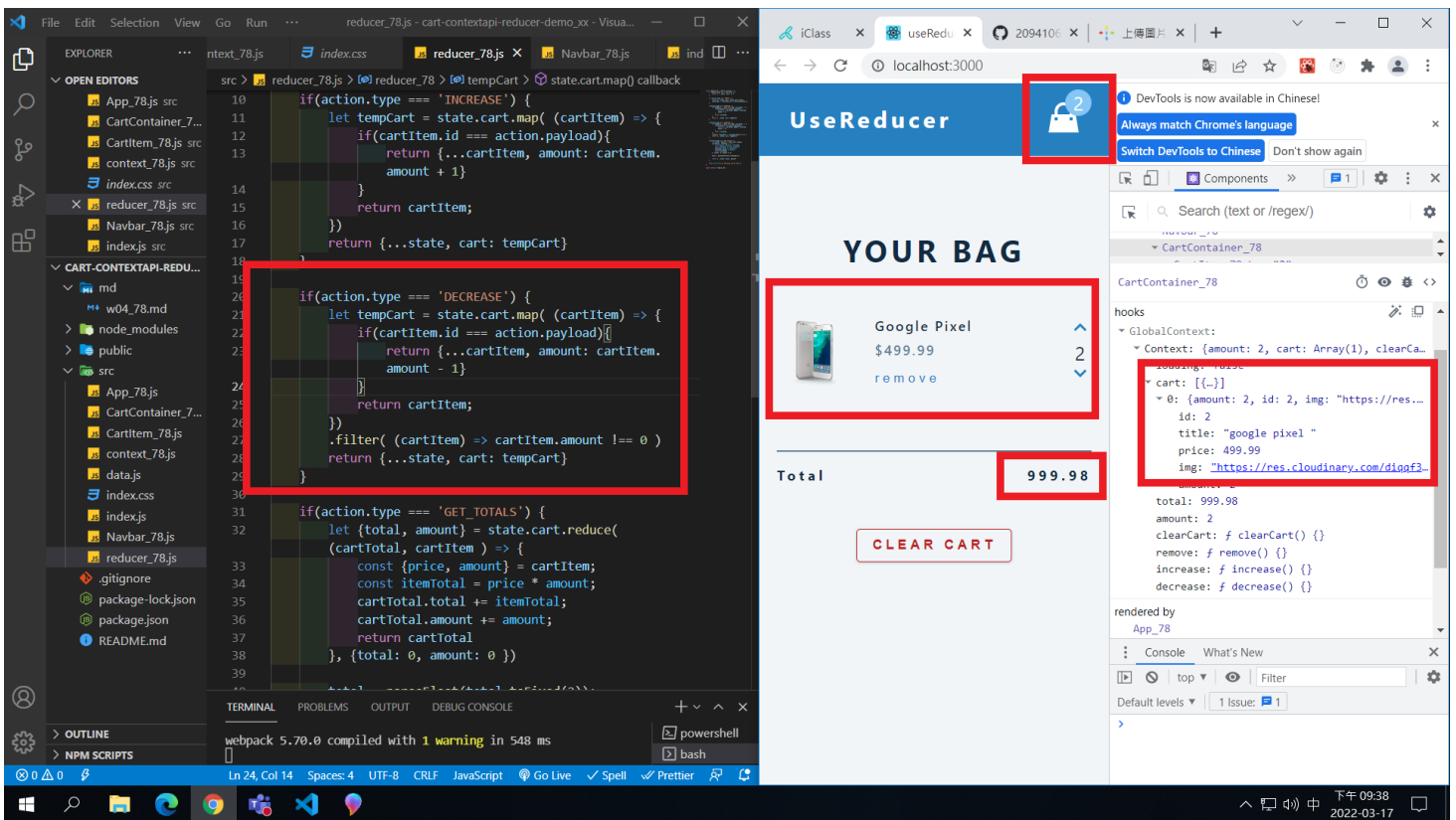
The screenshot shows a development environment with Visual Studio Code and a web browser. The code in `CartItem_78.js` is highlighted, showing the implementation of a `remove` button. The button's `onClick` handler calls `remove(id)`, which uses `useGlobalContext` to update the cart state. The browser preview shows the 'YOUR BAG' section with a Samsung Galaxy S7 item and a 'remove' button. The Redux DevTools state shows the cart array being updated.

The screenshot shows a development environment with Visual Studio Code and a web browser. The code in `reducer_78.js` is highlighted, showing the logic to filter out items from the cart when the 'CLEAR_CART' action is triggered. The browser preview shows the 'YOUR BAG' section with a Samsung Galaxy S7 item and a 'CLEAR CART' button. The Redux DevTools state shows the cart array being cleared.

p4



p5



p6

File Edit Selection View Go ... reducer_78.js - cart-contextapi-reducer-demo_x...

EXPLORER em_78.js reducer_78.js reducer_78.js Navbar_78

OPEN EDITORS

reducer_78.js

src > reducer_78.js > reducer_78

```
    cart.filter( (cartItem) =>
      cartItem.id !== action.payload )
    }
    return cartItem;
  })
  return { ...state, cart: tempCart }
}

if(action.type === 'GET_TOTALS') {
  let {total, amount} = state.cart.reduce(
    (cartTotal, cartitem) => {
      const {price, amount} = cartitem;
      const itemTotal = price * amount;
      cartTotal.total += itemTotal;
      cartTotal.amount += amount;
      return cartTotal
    }, {total: 0, amount: 0 } )

  total = parseFloat(total.toFixed(2));

  return {...state, total, amount}
}

throw new Error('no matching action type');
}

export default reducer_78;
```

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE


ms powershell bash


Ln 42, Col 46 Spaces: 4 UTF-8 CRLF JavaScript Go Live Spell Prettier


localhost:3000

UseReducer

YOUR BAG

 Samsung Galaxy S7
\$599.99
remove

 Google Pixel
\$499.99
remove

 Xiaomi Redmi Note 2
\$699.99
remove

Total 3699.94

CLEAR CART

Components

Search (text or /regex/)

AppProvider_78

Context.Provider

App_78

Context.Provider

props

children: <App_78 />

value: {amount: 6, cart: Array(3), clearCart: f clear...

amount: 6

cart: [({amount: 2, id: 1, img: "https://res.cloudi...

1: {amount: 2, id: 2, img: "https://res.cloudi...

amount: 2

id: 2

img: "https://res.cloudinary.com/diqqf3eq2/imag...

price: 499.99

title: "google pixel "

new entry: ""

2: {amount: 3, id: 3, img: "https://res.cloudi...

amount: 3

id: 3

img: "https://res.cloudinary.com/diqqf3eq2/imag...

price: 699.99

title: "Xiaomi Redmi Note 2"

new entry: ""

clearCart: f clearCart() {}

decrease: f decrease() {}

increase: f increase() {}

loading: false

remove: f remove() {}

下午 09:00

2022-03-17