

# W15-P1: Setup AppProvider\_xx for App\_xx and get state data

The screenshot shows a web application running on localhost:3000. The application is titled "UseReducer" and displays a "YOUR BAG" section. The bag contains four items: Samsung Galaxy S8 (\$399.99), Google Pixel (\$499.99), Xiaomi Redmi Note 2 (\$699.99), and Samsung Galaxy S7 (\$599.99). Each item has a "remove" button. The total is \$0. A "CLEAR CART" button is at the bottom. The DevTools Components panel shows the component tree with AppProvider\_27, Context.Provider, App\_27, and NavBar. The hooks panel shows the state of GlobalContext\_xx, including the cart array and total.

The screenshot shows the Visual Studio Code editor with two files open: index.js and Context\_27.js. The index.js file shows the import of AppProvider\_27 and the rendering of the App\_27 component. The Context\_27.js file shows the setup of the AppContext\_27, the AppProvider\_27 component, and the useGlobalContext\_xx hook.

## W15-P2: replace data in CartContainer using global context

The screenshot shows a web application running on localhost:3000. The browser window displays a shopping cart titled "YOUR BAG" with four items: Samsung Galaxy S8 (\$399.99), Google Pixel (\$499.99), Xiaomi Redmi Note 2 (\$699.99), and Samsung Galaxy S7 (\$599.99). The total is \$123.45, and there is a "CLEAR CART" button. The DevTools components panel shows the component tree with the following structure:

```
AppProvider_27
├── Context.Provider
│   └── App_27
│       └── NavBar
│           └── CartContainer
│               ├── CartItem key="rec1J21fCIB0Pd..."
│               ├── CartItem key="recB6qcHPxb62Y..."
│               ├── CartItem key="recdRxBsE14Rr2..."
│               └── CartItem key="recwTo160XST3P..."
```

The code editor shows the `CartContainer_27.js` file with the following code:

```
src > components > JS CartContainer_27.js > CartContainer
1 import React, { useState } from 'react';
2 import CartItem from './CartItem_27';
3 // import data from '../data';
4 import { useGlobalContext_xx } from '../context_27';
5
6
7 const CartContainer = () => {
8   // const [cart, setCart] = useState(data);
9   // const [total, setTotal] = useState(0);
10  // const [amount, setAmount] = useState(0);
11  const {cart, total, amount} = useGlobalContext_xx();
12  if (cart.length === 0) {
13    return (
14      <section className='cart'>
15        /* cart header */
16        <header>
17          <h2>your bag</h2>
18          <h4 className='empty-cart'>is currently empty</h4>
19        </header>
20      </section>
21    );
22  }
23  return (
24    <section className='cart'>
25      /* cart header */
26      <header>
27        <h2>your bag</h2>
28      </header>
29      /* cart items */
30      <div>
31        {cart.map((item) => {
32          return <CartItem key={item.id} {...item} />;
33        })}
34      </div>
35      /* cart footer */
36      <footer>
37        <hr />
38        <div className='cart-total'>
```

## W15-P3: put clear cart function in global context

The screenshot shows the code implementation for the clear cart function. The `Reducer_27.js` file has the following code:

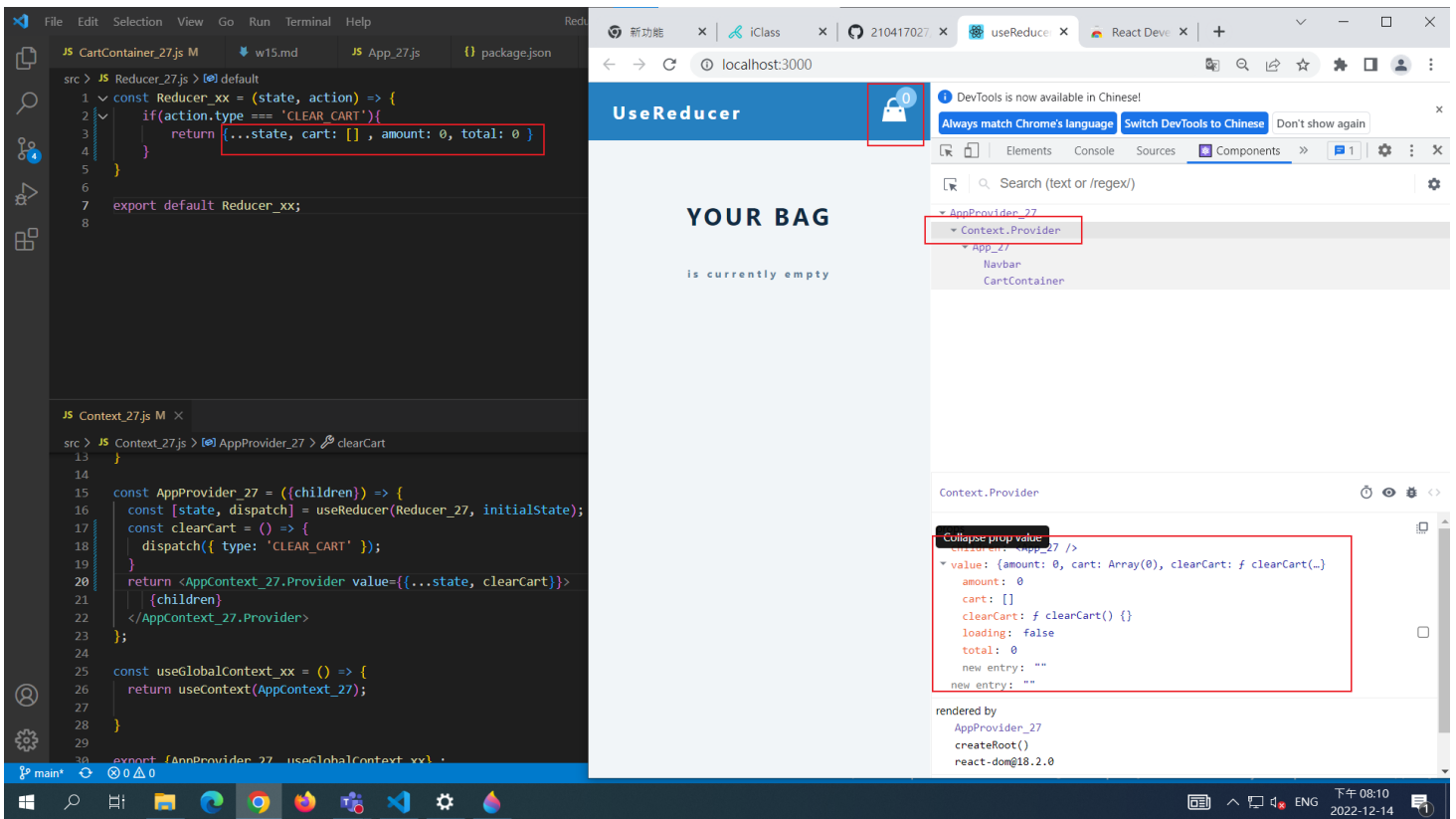
```
src > JS Reducer_27.js > default
1 const Reducer_xx = (state, action) => {
2   if (action.type === 'CLEAR_CART') {
3     return {...state, cart: [], amount: 0, total: 0 };
4   }
5 }
6
7 export default Reducer_xx;
```

The `Context_27.js` file has the following code:

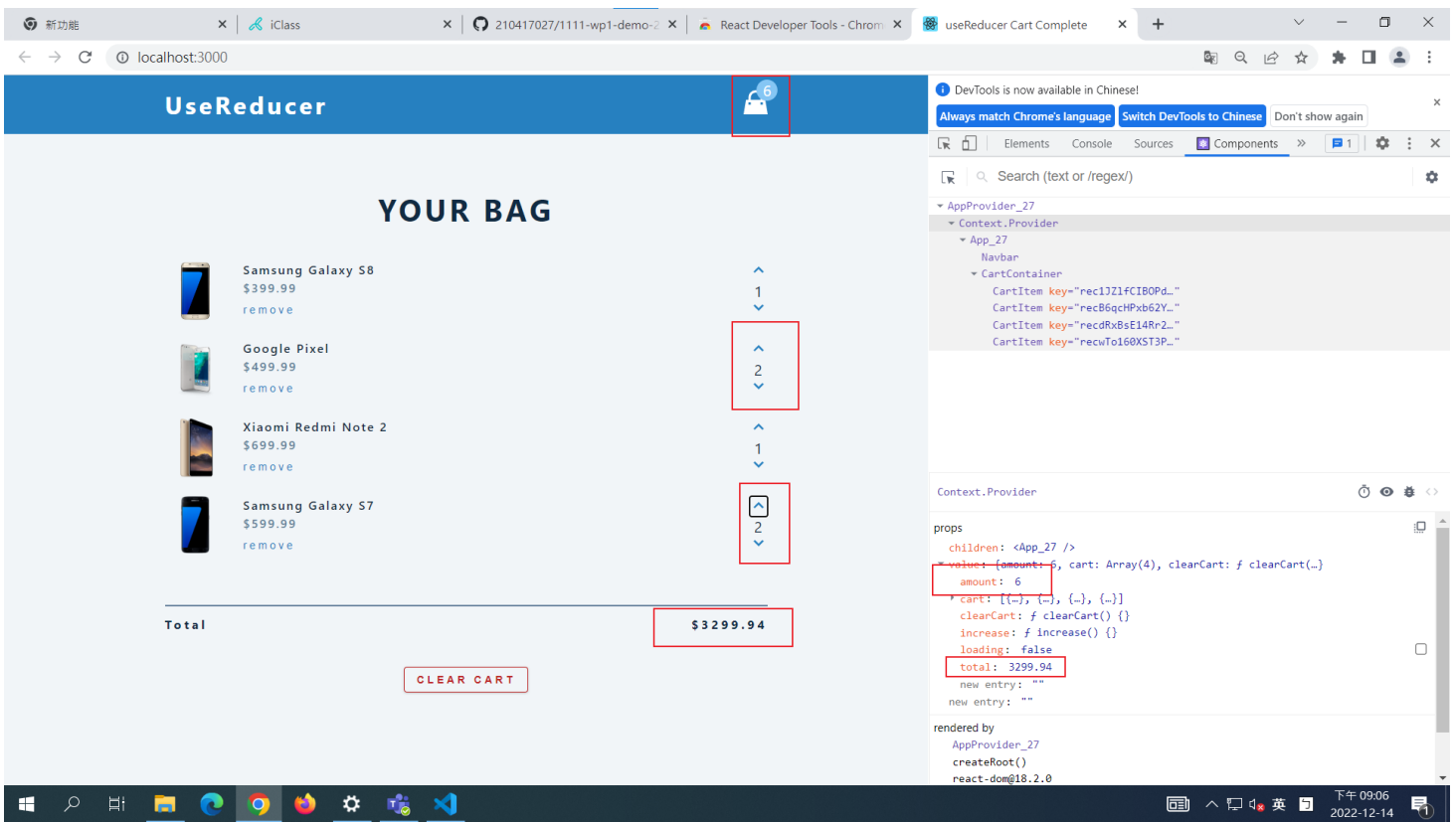
```
src > JS Context_27.js > AppProvider_27 > clearCart
13
14
15 const AppProvider_27 = ({children}) => {
16   const [state, dispatch] = useReducer(Reducer_27, initialState);
17   const clearCart = () => {
18     dispatch({ type: 'CLEAR_CART' });
19   }
20   return <AppContext_27.Provider value={{...state, clearCart}}>
21     {children}
22   </AppContext_27.Provider>
23 };
24
25 const useGlobalContext_xx = () => {
26   return useContext(AppContext_27);
27 }
28
29
30 export {AppProvider_27, useGlobalContext_xx};
```

The `CartContainer_27.js` file has the following code:

```
src > components > JS CartContainer_27.js > CartContainer
38
39 <div className='cart-total'>
40   total <span>${total}</span>
41 </div>
42 </div>
43 <button className='btn clear-btn'>
44   clear cart
45 </button>
46 </footer>
47 </section>
48
49
50
51 export default CartContainer;
```



## W15-P4: provide get total function in global context



```
src > JS Context_27.js > AppProvider_27 > useEffect() callback
11 amount: 4,
12 total: 123.45
13 }
14
15 const AppProvider_27 = ({children}) => {
16   const [state, dispatch] = useReducer(Reducer_27, initialState);
17   useEffect(() => {
18     dispatch({type: 'GET_TOTALS'});
19     [,state.cart]
20   },[state.cart])
21   const clearCart = () => {
22     dispatch({ type: 'CLEAR_CART' });
23   }
24
25   const increase = (id) => {
26     dispatch({type: 'INCREASE', payload: id});
27   }
28
29   return <AppContext_27.Provider value={{...state, clearCart, increase}}>
30     {children}
31   </AppContext_27.Provider>
32 </AppProvider_27>
33 </AppContext_27>
34
35 const useGlobalContext_xx = () => {
36   return useContext(AppContext_27);
37 }
38
39 export {AppProvider_27, useGlobalContext_xx};
40
41
src > JS Reducer_27.js > Reducer_xx
1 const Reducer_xx = (state, action) => {
2   if(action.type === 'CLEAR_CART'){
3     return {...state, cart: [], amount: 0, total: 0}
4   }
5   if(action.type === 'INCREASE') {
6     let tempCart = state.cart.map((cartItem) => {
7       if(cartItem.id === action.payload) {
8         return { ...cartItem, amount: cartItem.amount + 1 };
9       }
10      return cartItem;
11    });
12    return {...state, cart: tempCart}
13  }
14  if(action.type === 'GET_TOTALS'){
15    let {total, amount} = state.cart.reduce((cartTotal, cartItem) => {
16      const {price, amount} = cartItem;
17      const itemTotal = price * amount;
18      cartTotal.total += itemTotal;
19      cartTotal.amount += amount;
20      return cartTotal;
21    },{
22      total: 0,
23      amount: 0
24    });
25    total = parseFloat(total.toFixed(2));
26    return {...state, total, amount: amount}
27  }
28 }
29
30 export default Reducer_xx;
```

W15-P5: put remove cart item function in global context

W15-P6: provide decrease function in global context

```
src > components > JS CartItem_27.js > CartItem > decrease
8 <img src={img} alt={title} />
9 <div>
10   <h4>{title}</h4>
11   <h4 className='item-price'>${price}</h4>
12   /* remove button */
13   <button className='remove-btn' onClick={() => removeButton}>
14     remove
15   </button>
16 </div>
17 <div>
18   /* increase amount */
19   <button className='amount-btn' onClick={() => increase(id)}>
20     <svg xmlns='http://www.w3.org/2000/svg' viewBox='0 0 20 20'>
21       <path d='M10.707 7.051 10 6.343 4.343 12.141 1.414 10 9.172 4.243' />
22     </svg>
23   </button>
24   /* amount */
25   <p className='amount'>{amount}</p>
26   /* decrease amount */
27   <button className='amount-btn' onClick={() => decrease(id)}>
28     <svg xmlns='http://www.w3.org/2000/svg' viewBox='0 0 20 20'>
29       <path d='M9.293 12.951 7.051 10 9.172 4.243 1.414 10 9.172 12.951' />
30     </svg>
31   </button>
32 </div>
33 </article>
34 </div>
35
36 port default CartItem;
37
38
src > JS Context_27.js > ...
24 const removeButton = () => {
25   dispatch({ type: 'CLEAR_BUTTON' });
26 }
27
28 const increase = (id) => {
29   dispatch({type: 'INCREASE', payload: id});
30 }
31
32
33 const decrease = (id) => {
34   dispatch({type: 'DECREASE', payload: id});
35 }
36
37
38 return <AppContext_27.Provider value={{...state, clearCart, increase,decrea
39   {children}
40 </AppContext_27.Provider>
41
src > JS Reducer_27.js > ...
14
15 return {...state, cart: tempCart}
16
17 if(action.type === 'DECREASE') {
18   let tempCart = state.cart.map((cartItem) => {
19     if(cartItem.id === action.payload) {
20       return { ...cartItem, amount: cartItem.amount - 1 };
21     }
22     return cartItem;
23   });
24   return {...state, cart: tempCart}
25 }
26
27 if(action.type === 'GET_TOTALS'){
28   let {total, amount} = state.cart.reduce((cartTotal, cartItem) => {
29     const {price, amount} = cartItem;
30     const itemTotal = price * amount;
31     cartTotal.total += itemTotal;
```

