PAH:

App.js

import React, { useReducer } from 'react'

const cartReducer = (state, action) => {

  switch (action.type) {

    case 'ADD\_TO\_CART':

      return {

        ...state,

        cart: [...state.cart, action.payload],

      };

    case 'REMOVE\_FROM\_CART':

      return {

        ...state,

        cart: state.cart.filter(item => item.id !== action.payload),

      };

    default:

      return state;

  }

};

const App = () => {

  const [state, dispatch] = useReducer(cartReducer, { cart: [] });

  const addToCart = (item) => {

    dispatch({ type: 'ADD\_TO\_CART', payload: item });

  };

  const removeFromCart = (itemId) => {

    dispatch({ type: 'REMOVE\_FROM\_CART', payload: itemId });

  };

  return (

    <div>

      <h1>My E-Commerce App</h1>

      <div>

        <h2>Product List</h2>

        <ul>

          <li>

            Apple - $20

            <button onClick={() => addToCart({ id: 1, name: 'Apple ', price: 20 })}>

              Add to Cart

            </button>

          </li>

          <li>

            orange - $200

            <button onClick={() => addToCart({ id: 1, name: 'orange', price: 200 })}>

              Add to Cart

            </button>

          </li>

          <li>

            Banana- $20

            <button onClick={() => addToCart({ id: 1, name: 'Banana', price: 40 })}>

              Add to Cart

            </button>

          </li>

          <li>

            Kiwi - $20

            <button onClick={() => addToCart({ id: 1, name: 'Kiwi', price: 20 })}>

              Add to Cart

            </button>

          </li>

        </ul>

      </div>

      <div>

        <h2>Your Cart</h2>

        <ul>

          {state.cart.map(item => (

            <li key={item.id}>

              {item.name} - ${item.price}

              <button onClick={() => removeFromCart(item.id)}>

                Remove from Cart

              </button>

            </li>

          ))}

        </ul>

      </div>

    </div>

  );

};

export default App;

output:



