```
// App.jsx

import { BrowserRouter as Router, Routes, Route, Link } from 'react-router-dom';

import ProductList from './pages/ProductList';

import Cart from './pages/Cart';

import { CartProvider } from './context/CartContext';

import './App.css';


function App() {

  return (

    <CartProvider>

      <Router>

        <nav className="navbar">

          <Link to="/products">Products</Link>

          <Link to="/cart">Cart</Link>

        </nav>

        <Routes>

          <Route path="/products" element={<ProductList />} />

          <Route path="/cart" element={<Cart />} />

        </Routes>

      </Router>

    </CartProvider>

  );

}


export default App;
```

```
// context/CartContext.jsx
import { createContext, useContext, useState } from 'react';

const CartContext = createContext();

export function CartProvider({ children }) {
  const [cart, setCart] = useState([]);

  const addToCart = (product) => {
    setCart((prevCart) => {
      const item = prevCart.find((p) => p.id === product.id);
      if (item) {
        return prevCart.map((p) => (p.id === product.id ? { ...p, quantity: p.quantity + 1 } : p));
      }
      return [...prevCart, { ...product, quantity: 1 }];
    });
  };

  const removeFromCart = (id) => {
    setCart(cart.filter((item) => item.id !== id));
  };

  const updateQuantity = (id, amount) => {
    setCart((prevCart) =>
      prevCart.map((item) =>
        item.id === id ? { ...item, quantity: Math.max(1, item.quantity + amount) } : item
      )
    );
  };

  const totalItems = cart.reduce((total, item) => total + item.quantity, 0);
  const totalPrice = cart.reduce((total, item) => total + item.price * item.quantity, 0);

  return (
      <CartContext.Provider value={{ cart, addToCart, removeFromCart, updateQuantity, totalItems,
totalPrice }}>
      {children}
    </CartContext.Provider>
  );
}

export function useCart() {
  return useContext(CartContext);
}

// pages/ProductList.jsx
import { useEffect, useState } from 'react';
import { useCart } from '../context/CartContext';
import axios from 'axios';

function ProductList() {
  const [products, setProducts] = useState([]);
  const { addToCart } = useCart();
  useEffect(() => {
    axios.get('https://fakestoreapi.com/products?limit=30')
      .then(response => setProducts(response.data))
      .catch(() => alert('Failed to fetch products'));
  }, []);

  return (
    <div className="product-list">
      {products.map((product) => (
        <div key={product.id} className="product-card">
          <img src={product.image} alt={product.title} />
          <h3>{product.title}</h3>
          <p>${product.price}</p>
          <button onClick={() => addToCart(product)}>Add to Cart</button>
        </div>
      ))}
    </div>
  );
}

export default ProductList;

// pages/Cart.jsx
import { useCart } from '../context/CartContext';

function Cart() {
  const { cart, removeFromCart, updateQuantity, totalItems, totalPrice } = useCart();
  return (
    <div className="cart">
      <h2>Cart ({totalItems} items)</h2>
      {cart.map((item) => (
        <div key={item.id} className="cart-item">
          <h3>{item.title}</h3>
          <p>${item.price}</p>
          <div>
            <button onClick={() => updateQuantity(item.id, -1)}>-</button>
            <span>{item.quantity}</span>
            <button onClick={() => updateQuantity(item.id, 1)}>+</button>
            <button onClick={() => removeFromCart(item.id)}>Remove</button>
          </div>
        </div>
      ))}
      <h3>Total Price: ${totalPrice.toFixed(2)}</h3>
    </div>
  );
}

export default Cart;

// App.css
.navbar {
  display: flex;
  justify-content: space-around;
```

```css
  padding: 1rem;
  background: #333;
  color: white;
}
.product-list, .cart {
  padding: 2rem;
}
.product-card, .cart-item {
  border: 1px solid #ccc;
  margin: 1rem;
  padding: 1rem;
}
button {
  margin: 0.5rem;
}
```