**Exercise 24: Introduction to Redux**

Tạo components/Cart.js

import React from 'react';

import { useSelector } from 'react-redux';

const Cart = () => {

    const cartItems = useSelector(state => state.cart);

    const calculateTotal = () => {

        return cartItems.reduce((total, item) => {

            const quantity = item.quantity || 1;

            return total + (item.price \* quantity);

        }, 0).toFixed(2);

    };

    const getTotalItems = () => {

        return cartItems.reduce((total, item) => total + (item.quantity || 1), 0);

    };

    return (

        <div className="cart">

            <h2>🛒 Shopping Cart ({getTotalItems()} items)</h2>

            {cartItems.length === 0 ? (

                <p>Your cart is empty. Start adding some products! 🛍️</p>

            ) : (

                <>

                    <ul>

                        {cartItems.map((item, index) => (

                            <li key={`${item.id}-${index}`}>

                                <p><strong>{item.name}</strong></p>

                                <p>Price: ${item.price}</p>

                                <p>Quantity: {item.quantity || 1}</p>

                                <p>Subtotal: ${((item.price \* (item.quantity || 1)).toFixed(2))}</p>

                                <p>ID: {item.id}</p>

                                <p>Categories: {item.catalogs.join(', ')}</p>

                            </li>

                        ))}

                    </ul>

                    <div style={{

                        marginTop: '20px',

                        padding: '15px',

                        backgroundColor: '#e8f5e8',

                        borderRadius: '8px',

                        textAlign: 'center'

                    }}>

                        <h3 style={{ color: '#27ae60', margin: 0 }}>

                            Total: ${calculateTotal()}

                        </h3>

                    </div>

                </>

            )}

        </div>

    );

};

export default Cart;

Tạo components/ProductList.js

import React from 'react';

import { useDispatch } from 'react-redux';

import { addToCart, updateCartItem, removeFromCart } from '../redux/actions';

const products = [

    {

        id: '123456',

        name: 'Example Product',

        price: 9.99,

        description: 'This is an example product.',

        catalogs: ['catalog1', 'catalog2'],

    },

    {

        id: '234567',

        name: 'Wireless Headphones',

        price: 79.99,

        description: 'High-quality wireless headphones with noise cancellation.',

        catalogs: ['electronics', 'audio'],

    },

    {

        id: '345678',

        name: 'Smart Watch',

        price: 199.99,

        description: 'Fitness tracker with heart rate monitor and GPS.',

        catalogs: ['electronics', 'wearables'],

    },

    {

        id: '456789',

        name: 'Coffee Maker',

        price: 89.99,

        description: 'Automatic drip coffee maker with programmable timer.',

        catalogs: ['kitchen', 'appliances'],

    },

];

const ProductList = () => {

    const dispatch = useDispatch();

    const handleAddToCart = (product) => {

        dispatch(addToCart(product));

    };

    const handleUpdateCart = (productId) => {

        dispatch(updateCartItem(productId));

    };

    const handleDeleteFromCart = (productId) => {

        dispatch(removeFromCart(productId));

    };

    return (

        <div className="product-list">

            <h2>Products</h2>

            {products.map(product => (

                <div key={product.id} className="product-item">

                    <div>

                        <p>ID: {product.id}</p>

                        <p>Name: {product.name}</p>

                        <p>Price: ${product.price}</p>

                        <p>Catalogs: {product.catalogs.join(', ')}</p>

                    </div>

                    <div>

                        <button onClick={() => handleAddToCart(product)}>Add to Cart</button>

                        <button onClick={() => handleUpdateCart(product.id)}>Update to Cart</button>

                        <button onClick={() => handleDeleteFromCart(product.id)}>Delete from Cart</button>

                    </div>

                </div>

            ))}

        </div>

    );

};

export default ProductList;

Tạo redux/actions.js

export const addToCart = (product) => ({

    type: 'ADD\_TO\_CART',

    payload: product,

});

export const updateCartItem = (id) => ({

    type: 'UPDATE\_CART\_ITEM',

    payload: id,

});

export const removeFromCart = (id) => ({

    type: 'REMOVE\_FROM\_CART',

    payload: id,

});

Tạo redux/reducers.js

const initialState = {

    cart: [],

};

const Reducer = (state = initialState, action) => {

    switch (action.type) {

        case 'ADD\_TO\_CART':

            // Kiểm tra xem sản phẩm đã tồn tại chưa

            const existingItem = state.cart.find(item => item.id === action.payload.id);

            if (existingItem) {

                // Nếu đã tồn tại, tăng quantity

                return {

                    ...state,

                    cart: state.cart.map(item =>

                        item.id === action.payload.id

                            ? { ...item, quantity: (item.quantity || 1) + 1 }

                            : item

                    ),

                };

            } else {

                // Nếu chưa tồn tại, thêm mới với quantity = 1

                return {

                    ...state,

                    cart: [...state.cart, { ...action.payload, quantity: 1 }],

                };

            }

        case 'UPDATE\_CART\_ITEM':

            return {

                ...state,

                cart: state.cart.map(item =>

                    item.id === action.payload

                        ? { ...item, quantity: (item.quantity || 1) + 1 }

                        : item

                ),

            };

        case 'REMOVE\_FROM\_CART':

            return {

                ...state,

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

            };

        default:

            return state;

    }

};

export default Reducer;

Tạo redux/store.js

import { createStore } from 'redux';

import cartReducer from './reducers';

const store = createStore(cartReducer);

export default store;

App.js  
  
import React from 'react';

import { Provider } from 'react-redux';

import store from './redux/store';

import ProductList from './components/ProductList';

import Cart from './components/Cart';

import './App.css';

function App() {

    return (

        <Provider store={store}>

            <div className="App">

                <h1>Shopping Cart - Exercise 24</h1>

                <ProductList />

                <Cart />

            </div>

        </Provider>

    );

}

export default App;

index.js  
import React from 'react';

import ReactDOM from 'react-dom/client';

import App from './App';

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(

  <React.StrictMode>

    <App />

  </React.StrictMode>

);

App.css

/\* App.css - Giao diện dễ nhìn \*/

\* {

  margin: 0;

  padding: 0;

  box-sizing: border-box;

}

body {

  font-family: 'Arial', sans-serif;

  background-color: #f0f2f5;

  color: #333;

  line-height: 1.6;

}

.App {

  max-width: 1200px;

  margin: 0 auto;

  padding: 20px;

}

.App h1 {

  text-align: center;

  color: #2c3e50;

  margin-bottom: 30px;

  font-size: 2.2rem;

  font-weight: bold;

  text-shadow: 2px 2px 4px rgba(0,0,0,0.1);

}

/\* Product List Styles \*/

.product-list {

  background: white;

  border-radius: 10px;

  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

  padding: 25px;

  margin-bottom: 30px;

  border: 1px solid #e0e0e0;

}

.product-list h2 {

  color: #34495e;

  margin-bottom: 20px;

  font-size: 1.5rem;

  padding-bottom: 10px;

  border-bottom: 2px solid #3498db;

}

.product-item {

  display: flex;

  justify-content: space-between;

  align-items: center;

  padding: 20px;

  margin-bottom: 15px;

  border: 1px solid #ddd;

  border-radius: 8px;

  background: #fafafa;

  transition: all 0.3s ease;

}

.product-item:hover {

  background: #f0f8ff;

  border-color: #3498db;

  transform: translateY(-2px);

  box-shadow: 0 4px 12px rgba(0, 0, 0, 0.15);

}

.product-item div:first-child {

  flex: 1;

}

.product-item p {

  margin: 8px 0;

  font-size: 1rem;

}

.product-item p:first-child {

  font-weight: bold;

  color: #2c3e50;

  font-size: 1.1rem;

}

.product-item p:nth-child(3) {

  color: #e74c3c;

  font-weight: bold;

  font-size: 1.1rem;

}

.product-item div:last-child {

  display: flex;

  gap: 10px;

  flex-wrap: wrap;

}

button {

  padding: 10px 15px;

  border: none;

  border-radius: 5px;

  cursor: pointer;

  font-size: 0.9rem;

  font-weight: 500;

  transition: all 0.3s ease;

  min-width: 100px;

}

button:hover {

  transform: translateY(-1px);

  box-shadow: 0 2px 8px rgba(0, 0, 0, 0.2);

}

button:active {

  transform: translateY(0);

}

/\* Button Colors \*/

button:nth-child(1) {

  background: #27ae60;

  color: white;

}

button:nth-child(1):hover {

  background: #219a52;

}

button:nth-child(2) {

  background: #f39c12;

  color: white;

}

button:nth-child(2):hover {

  background: #e67e22;

}

button:nth-child(3) {

  background: #e74c3c;

  color: white;

}

button:nth-child(3):hover {

  background: #c0392b;

}

/\* Cart Styles \*/

.cart {

  background: white;

  border-radius: 10px;

  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

  padding: 25px;

  border: 1px solid #e0e0e0;

}

.cart h2 {

  color: #34495e;

  margin-bottom: 20px;

  font-size: 1.5rem;

  padding-bottom: 10px;

  border-bottom: 2px solid #e74c3c;

}

.cart p {

  text-align: center;

  color: #7f8c8d;

  font-size: 1.1rem;

  padding: 30px 0;

  font-style: italic;

}

.cart ul {

  list-style: none;

}

.cart li {

  background: #f8f9fa;

  border: 1px solid #e9ecef;

  border-radius: 8px;

  padding: 15px;

  margin-bottom: 10px;

  transition: all 0.3s ease;

}

.cart li:hover {

  background: #e3f2fd;

  border-color: #3498db;

  transform: translateX(5px);

}

.cart li p {

  margin: 5px 0;

  text-align: left;

  padding: 0;

  font-size: 0.95rem;

}

.cart li p:first-child {

  font-weight: bold;

  color: #2c3e50;

  font-size: 1.05rem;

}

.cart li p:nth-child(2) {

  color: #e74c3c;

  font-weight: bold;

}

/\* Responsive Design \*/

@media (max-width: 768px) {

  .App {

    padding: 15px;

  }

  .App h1 {

    font-size: 1.8rem;

  }

  .product-item {

    flex-direction: column;

    gap: 15px;

    text-align: center;

  }

  .product-item div:last-child {

    justify-content: center;

  }

  button {

    flex: 1;

    max-width: 120px;

  }

}

@media (max-width: 480px) {

  .product-item div:last-child {

    flex-direction: column;

    width: 100%;

  }

  button {

    max-width: 100%;

  }

}

/\* Smooth animations \*/

@keyframes fadeIn {

  from {

    opacity: 0;

    transform: translateY(20px);

  }

  to {

    opacity: 1;

    transform: translateY(0);

  }

}

.cart li {

  animation: fadeIn 0.3s ease;

}

/\* Enhanced visual feedback \*/

.product-item:active {

  transform: translateY(0);

}

/\* Better spacing and readability \*/

.product-item p {

  line-height: 1.4;

}

.cart li p {

  line-height: 1.4;

}







