

## Practical Works 3 - Navigation & Global State

**Objective:** Transform the application into a true Single Page Application (SPA) by adding multiple pages (Product Details, Cart) and managing global state using the Context API so the "Add to Cart" buttons actually work.

### Deliverables for the end of TP3:

1. **Working Navigation:** Clicking on the logo returns to Home. Clicking on a Product Card goes to the Details page. Clicking the Cart goes to the Cart page.
2. **Persistent Cart:** Adding items from the Home Page or the Product Details page should update the red notification badge in the Navbar immediately.
3. **Cart Math:** The Cart page must correctly multiply the quantity of items by their price, and display the correct Grand Total.

### 1. Installing React Router

To navigate between our **Home** page, **Product** page, and **Cart** without refreshing the browser, we **need React Router**. Open your terminal inside the **project-store folder** and run:

```
npm install react-router-dom
```

### 2. The "Brain" of the Cart: Context API

If we put the cart data inside **Home.jsx**, the Navbar and the **ProductPage** won't be able to see it. We need a "Global State."

Create a new folder called **context** inside **src**. Inside it, create **CartContext.jsx**:

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

export const CartContext = createContext();

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

  // Function to add a product to the cart
  const addToCart = (product) => {
    setCart((prevCart) => {
      // Check if item is already in cart
      const existingItem = prevCart.find((item) => item.id === product.id);
      if (existingItem) {
        return prevCart.map((item) =>
          item.id === product.id ? { ...item, quantity: item.quantity + 1 } : item
        );
      }
      // If new item, add with quantity 1
      return [...prevCart, { ...product, quantity: 1 }];
    });
    alert(` ${product.title} added to cart!`);
  };

  // Calculate total items (for Navbar)
  const totalItems = cart.reduce((acc, item) => acc + item.quantity, 0);
```

```
// Calculate total price (for Cart Page)
const totalPrice = cart.reduce((acc, item) => acc + item.price * item.quantity, 0);

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

### 3. Setting Up the Router & Provider in App.jsx

Now, we wrap our entire application in the **CartProvider** and **BrowserRouter**.

Update **src/App.jsx**:

```
// src/App.jsx
import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';
import { CartProvider } from './context/CartContext';
import Navbar from './components/Navbar';
import Home from './pages/Home';
import ProductDetails from './pages/ProductDetails'; // We will create this next
import Cart from './pages/Cart'; // We will create this next

export default function App() {
  return (
    <CartProvider>
      <Router>
        <div className="min-h-screen bg-gray-50">
          <Navbar />
          <main>
            <Routes>
              <Route path="/" element={<Home />} />
              <Route path="/product/:id" element={<ProductDetails />} />
              <Route path="/cart" element={<Cart />} />
            </Routes>
          </main>
        </div>
      </Router>
    </CartProvider>
  );
}
```

### 4. Updating Components to Use the Cart & Links

**A. Update src/components/Navbar.jsx** We need to use the **Link** component from **React Router**, and read the **totalItems** from our **Context**.

```
import { useContext } from 'react';
import { Link } from 'react-router-dom';
import { CartContext } from './context/CartContext';

export default function Navbar() {
  const { totalItems } = useContext(CartContext); // Get data from Context

  return (
    <nav className="flex justify-between items-center p-4 bg-white shadow-md sticky top-0 z-50">
      <Link to="/" className="text-2xl font-bold text-orange-600">DZ-Shop</Link>

      <div className="flex gap-5 font-semibold items-center">
        <span>Login</span>
        <Link to="/cart" className="relative cursor-pointer">
          <span>Cart</span>
          <span className="absolute -top-2 -right-3 bg-red-500 text-white text-xs rounded-full px-2 py-0.5">
            {totalItems}
          </span>
        </Link>
      </div>
    </nav>
  );
}
```

```

    </Link>
  </div>
</nav>
);
}

```

**B. Update `src/components/ProductCard.jsx`** Make the card clickable to go to details, and make the button actually add to the cart.

```

import { useContext } from 'react';
import { Link } from 'react-router-dom';
import { CartContext } from '../context/CartContext';

export default function ProductCard({ product }) {
  const { addToCart } = useContext(CartContext);

  return (
    <div className="bg-white border p-4 rounded-lg shadow-sm hover:shadow-lg transition flex flex-col">
      <Link to={` /product/${product.id}`} className="flex-1">
        <div className="h-48 flex items-center justify-center overflow-hidden mb-4">
          <img src={product.image} alt={product.title} className="max-h-full object-contain" />
        </div>
        <h3 className="text-sm font-semibold truncate">{product.title}</h3>
        <p className="text-gray-500 text-xs mt-1 capitalize">{product.category}</p>
      </Link>

      <div className="flex justify-between items-center mt-3">
        <span className="text-orange-600 font-bold">${product.price}</span>
        <button
          onClick={() => addToCart(product)}
          className="bg-orange-100 text-orange-600 px-3 py-1 rounded text-xs font-bold hover:bg-orange-600 hover:text-white transition">
          Add to Cart
        </button>
      </div>
    </div>
  );
}

```

## 5. Creating the Product Details Page

Create `src/pages/ProductDetails.jsx`. This page reads the `:id` from the URL, fetches that specific product, and displays it.

```

import { useState, useEffect, useContext } from 'react';
import { useParams } from 'react-router-dom';
import { CartContext } from '../context/CartContext';

export default function ProductDetails() {
  const { id } = useParams(); // Gets the ID from the URL
  const { addToCart } = useContext(CartContext);
  const [product, setProduct] = useState(null);

  useEffect(() => {
    fetch(` https://fakestoreapi.com/products/${id} `)
      .then(res => res.json())
      .then(data => setProduct(data));
  }, [id]);

  if (!product) return <div className="p-8 text-center text-xl">Loading product...</div>;

  return (
    <div className="max-w-4xl mx-auto p-8 bg-white mt-10 rounded-lg shadow-md flex flex-col md:flex-row gap-10">
      <div className="w-full md:w-1/2 flex justify-center">
        <img src={product.image} alt={product.title} className="max-h-96 object-contain" />
      </div>
      <div className="w-full md:w-1/2 flex flex-col justify-center">

```

```

<h2 className="text-3xl font-bold text-gray-800 mb-4">{product.title}</h2>
<p className="text-gray-500 capitalize mb-2">{product.category}</p>
<p className="text-orange-600 text-4xl font-bold mb-6">${product.price}</p>
<p className="text-gray-700 leading-relaxed mb-8">{product.description}</p>

<button
  onClick={() => addToCart(product)}
  className="bg-orange-600 text-white py-3 px-8 rounded-lg font-bold text-lg hover:bg-orange-700 transition w-full md:w-
auto">
  Add to Cart
</button>
</div>
</div>
);
}

```

## 6. Creating the Cart Page

Create **src/pages/Cart.jsx**. This page lists everything in the cart and displays the total price using the **.reduce()** function we set up in the **Context**.

```

import { useContext } from 'react';
import { CartContext } from '../context/CartContext';
import { Link } from 'react-router-dom';

export default function Cart() {
  const { cart, totalPrice } = useContext(CartContext);

  if (cart.length === 0) {
    return (
      <div className="p-20 text-center">
        <h2 className="text-3xl font-bold text-gray-800 mb-4">Your Cart is Empty</h2>
        <Link to="/" className="text-orange-600 font-bold hover:underline">← Go back to shopping</Link>
      </div>
    );
  }

  return (
    <div className="max-w-4xl mx-auto p-8 mt-10">
      <h2 className="text-3xl font-bold mb-8 text-gray-800">Shopping Cart</h2>

      <div className="bg-white rounded-lg shadow-md p-6 mb-6">
        {cart.map((item, index) => (
          <div key={index} className="flex items-center justify-between border-b pb-4 mb-4 last:border-0 last:pb-0 last:mb-0">
            <div className="flex items-center gap-4 w-1/2">
              <img src={item.image} alt={item.title} className="h-16 w-16 object-contain" />
              <h4 className="text-sm font-semibold truncate" title={item.title}>{item.title}</h4>
            </div>
            <div className="w-1/4 text-center">
              <span className="text-gray-500">Qty: {item.quantity}</span>
            </div>
            <div className="w-1/4 text-right">
              <span className="font-bold text-orange-600">${(item.price * item.quantity).toFixed(2)}</span>
            </div>
          </div>
        ))}
      </div>

      <div className="bg-white rounded-lg shadow-md p-6 flex justify-between items-center">
        <span className="text-xl font-bold text-gray-800">Total:</span>
        <span className="text-3xl font-bold text-orange-600">${totalPrice.toFixed(2)}</span>
      </div>
    </div>
  );
}

```