**Shopping Cart Application - Design & Flow**

**1. Introduction**

This document provides an overview of the Shopping Cart application, covering its architecture, component structure, state management, and flow.

**2. Folder Structure**

The application follows a structured approach for better maintainability:  
- src/  
 - components/  
 - Navbar.jsx  
 - context/  
 - CartContext.jsx  
 - ProductContext.jsx  
 - pages/  
 - Layout.jsx  
 - ProductsPage.jsx  
 - CartPage.jsx  
 - App.js  
 - main.jsx

**3. State Management**

The application uses React's Context API for state management. The CartContext and ProductContext provide global state for cart operations and product handling.

**4. Application Flow**

**4.1 Navbar Component**

The Navbar includes navigation links, product and cart icons, and updates dynamically based on the cart state.

**4.2 Product Listing**

The ProductPage fetches products and displays them in a responsive grid layout. Users can add products to the cart from this page.

**4.3 Cart Functionality**

The CartPage displays selected items with quantity control, total price calculation, and a remove option. State persistence is managed using LocalStorage.

**5. LocalStorage Integration**

Cart and product data are stored in LocalStorage to maintain state across page reloads. Utility functions handle storing, retrieving, and updating cart data.

**6. Technologies Used**

- React.js (with Vite)  
- Tailwind CSS for styling  
- React Icons for UI enhancements  
- Context API for state management  
- LocalStorage for persistent data handling

**7. Conclusion**

This document outlines the Shopping Cart application's structure, implementation, and flow. The architecture ensures scalability and maintainability.