Day 4: Dynamic Frontend Components - [Furniture]

Technical Report: Dynamic Product Display with Add to Cart and Checkout Functionality

1. Overview: This report outlines the implementation of a dynamic product display system integrated with an "Add to Cart" and "Checkout" functionality. The system is designed to showcase products dynamically, allow users to view detailed product information, and facilitate a seamless shopping experience.

2. Features:

2.1 Dynamic Product Display:

Products are fetched dynamically from a backend (e.g., Sanity, Firebase, or a custom API).

The product grid includes details such as:

- Product Image
- Name
- Price
- Add to Cart (Button)
- Stock
- Rating

Each product includes a "See Details" button to view more information.

2.2 Product Details Page:

Clicking the "See Details" button opens a detailed view of the product.

The detailed view includes:

- High-resolution product image(s)
- Full description
- Product specifications (e.g., material, dimensions)
- Price
- Add to Cart button
- See Less Button

2.3 Add to Cart Functionality:

Users can add products to the cart by clicking the "Add to Cart" button on the product details page.

2.4 Sorting , Category , Filters and other Buttons:

A dropdown menu is provided for specific attributes, such as:

- ✧ **Category**: Sofas, Chairs, Vases, Lamps
- ✧ **Product Type**: E.g., modern, classic
- ✧ **Brand**: A list of available brands

Currently, clicking on "Category" and "Price" buttons only opens their respective dropdowns, but further functionality will be added to enable filtering and sorting.

The cart icon updates dynamically to reflect the number of items added.

2.5 Checkout Component:

The checkout process includes:

- ❖ A form for user details (e.g., name, address, payment information)
- ❖ Summary of selected products with their prices
- ❖ A "Place Order" button
- ❖ If the cart is empty, an alert is triggered:

"Please add items to the cart before proceeding to checkout.

3. Implementation Details

3.1 Technology Stack:

- ✧ Frontend: React (Next.js) with TypeScript
- ✧ Backend: Sanity for data management
- ✧ Data Management: Sanity
- ✧ Data Fetching: Mock Api

3.2 Folder Structure:

src/app/

product/: Displays the product grid dynamically

[id]/: Handles individual product details

cart/: Manages the cart functionality and integrates the **Checkout** component

components/

ProductCard: Displays individual product cards

ProductDetails: Handles detailed product view

Cart: Manages cart-related components

Checkout: Handles the checkout form and order placement

3.3 Dynamic Dropdown:

Dropdown menus are implemented using React state and hooks.

Each dropdown dynamically populates options such as product categories and brands based on the fetched data.

3.4 User Alerts:

Alerts are implemented using JavaScript's `window.alert()` or a custom modal component.

Example: If the cart is empty, the alert message is shown when attempting to proceed to checkout.

4. Current Work and Future Enhancements

4.1 Ongoing Tasks:

- ✧ Integrating and testing the dropdown functionality.
- ✧ Adding functionality for filtering and sorting by category and price.
- ✧ Ensuring proper validation of the checkout form.

4.2 Planned Features:

- ✧ Search functionality to filter products by name or category.
- ✧ Recommendations based on user preferences.
- ✧ Multi-language support.
- ✧ Enhanced alert modals for a better user experience.

5. **Conclusion:** This system ensures a dynamic and user-friendly shopping experience. The modular architecture and dynamic dropdown functionality allow for easy scalability and integration of additional features. By addressing current tasks and planned enhancements, the platform will offer a comprehensive solution for e-commerce.

➢ Prepared by: Areeba Zafar (Student Leader)