Hackathon Day 4

Dynamic Frontend Components

Introduction:

In this documentation, we present the dynamic frontend components developed for **Furniro** as part of the hackathon project. The goal was to implement various dynamic functionalities that enhance the user experience by providing real-time data, smooth navigation, and interaction with the marketplace platform.

The project was built using **Next.js, Tailwind CSS,** and integrated with **Sanity CMS** for content management. On **Day 4**, the focus was on developing core dynamic frontend components, including:

- **Product Listing Page**
- **Product Detail Pages**
- **Cart and Wishlist Functionality**
- **Pagination**
- **Related Products Section**
- **Search Bar**

These components were designed to be **responsive, user-friendly, and seamlessly integrated** into the marketplace environment. The following sections outline the key components developed, their functionalities, and the corresponding code implementations.

Functional Deliverables

1. Product Listing Page with Dynamic Data

The **Product Listing Page** dynamically renders products available in the marketplace. It fetches data from **Sanity CMS** and displays it in a **grid format**.

Key Features

✓ **Responsive design:** Adjusts automatically for mobile and desktop views.
✓ **Product cards with hover effects:** Enhances interactivity.
✓ **Dynamic rendering:** New or updated products are reflected instantly without reloading the page.

2. Individual Product Detail Pages

Each **product card** links to an **individual Product Detail Page**, which showcases **comprehensive product information** such as:

- Description
- Available sizes/colors
- Option to add the product to the cart

Key Features

✅ **Dynamic routing** based on product ID.
✅ **Real-time product details** fetched from the API.
✅ **"Add to Cart" functionality** directly from the page.

3. Add to Cart Page

The **Cart Page** provides an overview of items the user has added to their cart.

Key Features

✅ Modify product quantity or remove items.
✅ **Live total price calculations.**
✅ **React State Management** ensures UI updates in real-time.

4. Wishlist Page

The **Wishlist Page** allows users to **save their favorite products** for future purchases.

Key Features

✅ Add and remove items from the wishlist.
✅ Navigate to the **Product Details Page** via the "View Details" button.
✅ Wishlist data is stored using **Local Storage** or **React Context API**.
✅ Seamless transition to the cart for easy checkout.

5. Pagination Component

Handling large product datasets efficiently was a priority. The **Pagination Component** allows users to navigate through different product pages.

Key Features

✅ **Optimized load times** with paginated product listings.
✅ Users can **navigate through pages** via **page numbers or next/previous buttons**.
✅ The product list updates dynamically based on **user page selection**.

---

6. Search Bar

The **Search Bar** enables users to **quickly find products** by searching for names, categories, or tags.

Key Features

✅ **Real-time search** using React hooks.
✅ Filters products based on user input.
✅ Uses **client-side filtering** and **server-side queries** for better performance.

---

7. Header Component

The **Header Component** plays a crucial role in site navigation, ensuring users can **access key sections** like the homepage, product categories, cart, and user profile.

Key Features

✅ **Logo & Navigation:** Displays the site logo, home button, and links to product categories.
✅ **Search Bar:** Integrated into the header for easy product searches.
✅ **Cart Icon:** Displays the number of items in the cart, allowing quick access.

---

8. Related Products Component

The **Related Products Component** is displayed on the **Product Detail Page**, encouraging **cross-selling and upselling**.

Key Features

✅ Displays **related products** based on categories, tags, or purchase history.
✅ Fetches related product data from **Sanity CMS or API**.
✅ Provides an **interactive way** to explore more products.

---

9. Category Page

The **Category Page** groups products based on predefined categories (e.g., sofas, chairs, tables). When a user selects a category, they are redirected to a **filtered product listing page**.

Key Features

✅ **Dynamic routing** for category-specific pages.
✅ Fetches **category-specific products** from the API.
✅ **Filtering and sorting options** for better user experience.
✅ **Breadcrumb navigation** to track user location within the site.

Code Deliverables

The following key components were developed:
✅ **ProductCard Component**
✅ **ProductList Component**
✅ **Pagination Component**
✅ **Search Bar Component**
✅ **Header Component**
✅ **Category Page Component**

Challenges and Solutions

Challenge 1: Handling Large Data Sets

● **Problem:** Rendering large product catalogs led to slow performance.
✅ **Solution:** Implemented **pagination** and **lazy loading** to ensure efficient data fetching.

Challenge 2: State Management Across Components

● **Problem:** Managing the cart and wishlist across multiple pages.
✅ **Solution:** Used **React Context API** to maintain a **global state**, ensuring real-time updates.

Challenge 3: Categorizing Products

● **Problem:** Displaying products by category required dynamic routing and filtering.
✅ **Solution:** The **Category Page** dynamically fetches products based on **user-selected categories**.

Conclusion

By **Day 4** of the hackathon, we successfully implemented several **key dynamic components** for the marketplace project. These components enhance the **user experience** by providing **real-time updates, easy navigation, and seamless interactions**.

Self-Validation Checklist

✅ Frontend Component Development

✓ Product Listing Page, Product Detail Pages, Category Page, and Cart functionality completed
✗ Yet to implement advanced features like product reviews and ratings

✅ Styling and Responsiveness

✓ All pages and components are responsive using Tailwind CSS
✗ Testing responsiveness on various devices is ongoing

✅ Code Quality

✓ Code is well-structured, modular, and reusable
✗ Code documentation is still being refined

✅ Documentation and Submission

✓ Functional components are documented properly
✗ Final submission for Day 4 is pending

Final Review

✓ **Frontend Components:** Fully functional, including category page and header.
✓ **User Experience:** Smooth navigation, responsive design, and real-time updates.
✓ **Performance Optimization:** Pagination and lazy loading enhance efficiency.
✗ **Areas to Improve:** Final testing on **mobile devices** for responsiveness and load times.

This revised version ensures **clarity, readability, and professionalism** while keeping the technical details intact. Let me know if you need further refinements! 🚀