Day 4 - Dynamic Frontend Components - VAULTSNEAKS

Objective

On Day 4, the focus was on designing and developing dynamic frontend components for the shoe marketplace. The goal was to enhance the user experience by integrating data from Sanity CMS and creating scalable, reusable components.

Key Learning Outcomes:

- 1. Building dynamic frontend components to display data from Sanity CMS.
- 2. Implementing reusable and modular components.
- 3. Understanding state management techniques.
- 4. Applying responsive design and UX/UI best practices.

Components Developed:

- 1. Product Listing Component: Displayed products dynamically in a grid layout.
- 2. **Product Detail Component**: Provided detailed views of individual products using dynamic routing.
- 3. **Cart Component**: Showed added items, quantities, and total price, with state management for cart items.
- 4. **Search Bar**: Implemented functionality to filter products by name or tags.
- 5. **Product Comparison Page**: Developed a feature allowing users to compare multiple products side-by-side, highlighting differences in features, prices, and reviews.

Steps for Implementation:

- 1. **Setup**: Ensured the Next.js project was connected to Sanity CMS for data fetching.
- 2. Component Building: Developed each component with a focus on modularity and reusability.
- 3. **Styling**: Used Tailwind CSS to ensure the components were responsive across devices.
- 4. **State Management with Zustand**: Implemented **Zustand** for global state management, providing a lightweight and flexible solution for managing application state across components.

Expected Output:

- A fully functional product listing page displaying dynamic data.
- Individual product detail pages with accurate data rendering
- Advanced category filters and a search bar for refined product views.
- A product comparison page for side-by-side comparison of product features.

Challenges and Solutions:

- Challenge: Handling dynamic routing for product detail pages.
- **Solution**: Utilized Next.js dynamic routes to ensure seamless navigation between product listings and details.

Best Practices Followed:

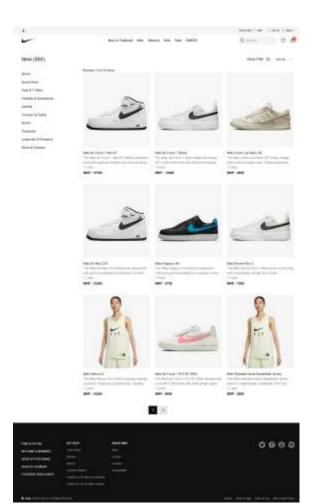
• Modular and reusable component design.

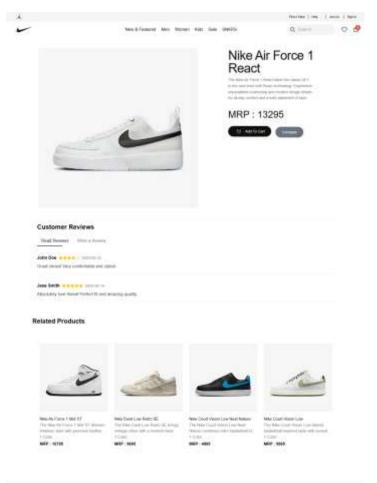
- State management for dynamic data binding using Zustand.
- Responsive and user-friendly UI design.

Day 4 Checklist: Self-Validation

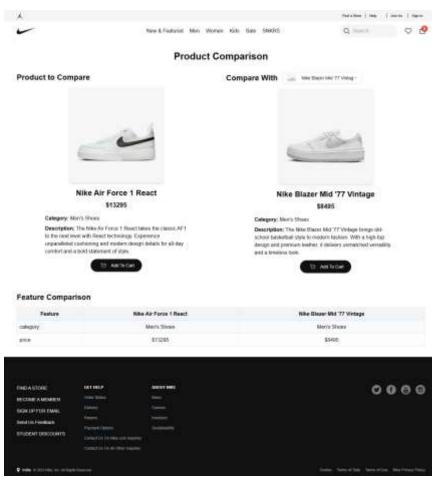
This checklist provides a quick validation of the key tasks accomplished on Day 4, ensuring all critical steps in building dynamic frontend components were thoroughly completed.

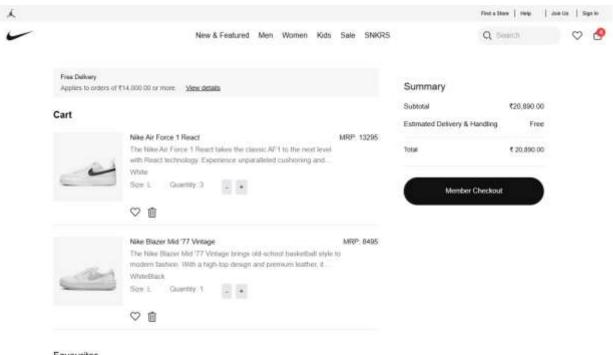
| Task | Status |
|----------------------------------|----------|
| Component Development | ✓ |
| API Integration with Sanity CMS | ✓ |
| State Management with Zustand | √ |
| Responsive Design Implementation | √ |
| Pagination and Filtering | √ |
| Submission Preparation | √ |











Favourites

There are no tiems saved to your favourites.

