



# Day 4 Dynamic Frontend Components-Nike Store

Unlocking the Power of Dynamic Marketplace Components: A Comprehensive Guide to Modularity, Reusability, and Seamless Integration with Sanity CMS. Explore each feature in detail, followed by an insightful conclusion that highlights our innovative approach.

## Step 1: Functionalities Overview:

The project implements the following core functionalities:

1. **Product Listing Page**
2. **Dynamic Route**
3. **Cart Functionality**
4. **Working Categories Filters**
5. **Price Calculation**

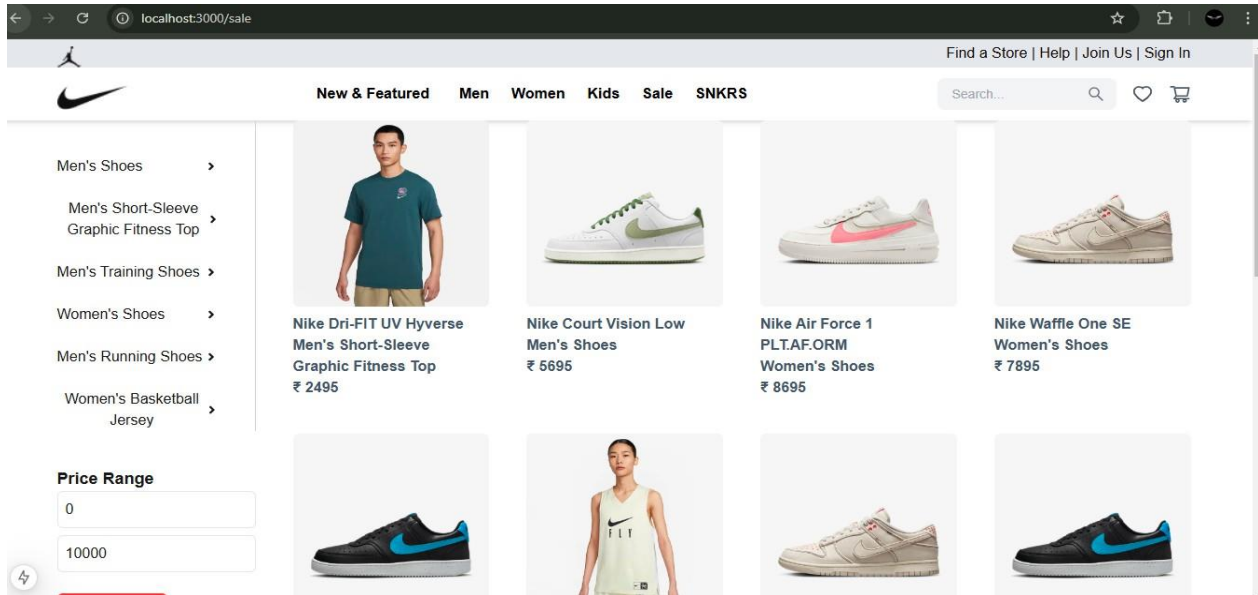
Each functionality contributes to building a responsive and scalable marketplace

## Step 2: Functionalities in Detail:

### 1. Product Listing Page:

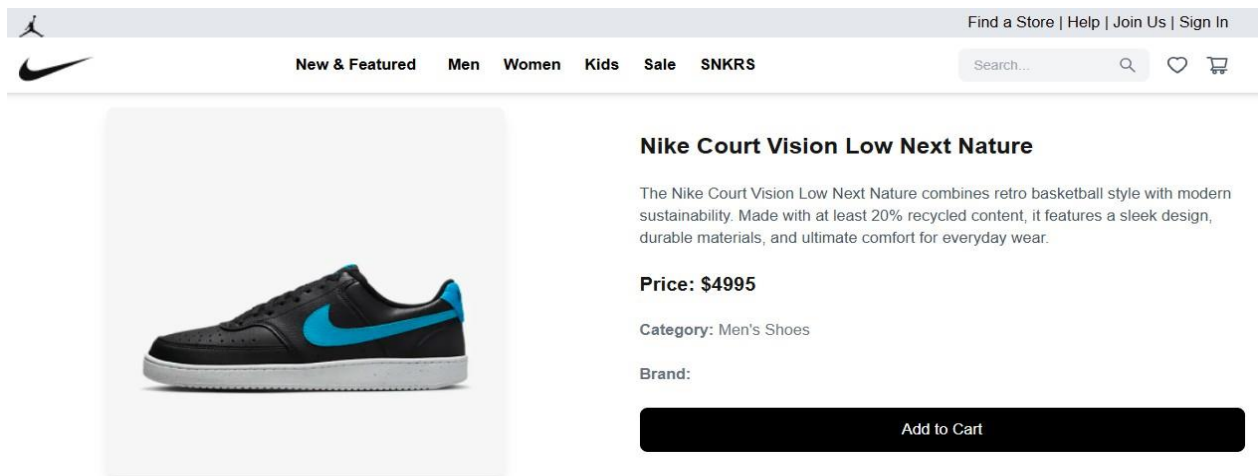
The Product Listing Page serves as the vibrant heart of the marketplace, offering users an engaging and visually appealing way to explore all available products. With dynamic content pulled directly from Sanity CMS, products are seamlessly displayed in a sleek grid or list layout, providing an intuitive browsing experience.

- **Sorting & Filtering:** Easily organize products by price, category, or popularity, making it simple for users to find exactly what they're looking for.
- **Pagination:** Smoothly navigate through large product listings, enhancing performance and ensuring a fast, seamless experience.
- **Responsive Design:** Optimized for all devices, ensuring the page looks great and functions flawlessly on desktops, tablets, and mobiles

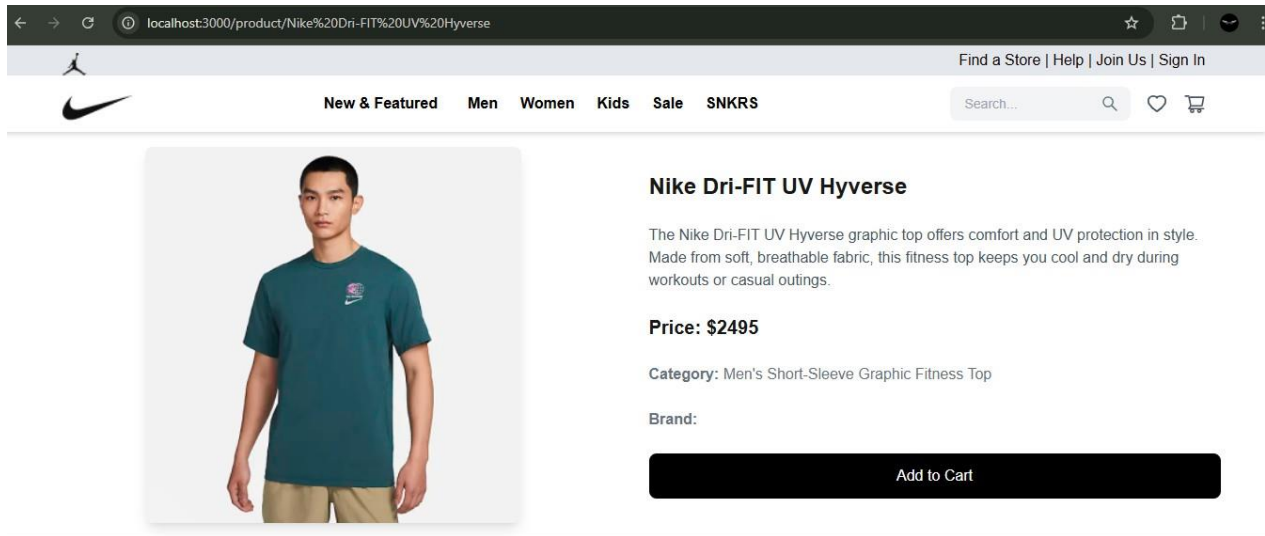


## 2. Dynamic Route:

Unlock the power of personalized browsing with dynamic product detail pages! Each product gets its own dedicated page, offering users in-depth information and a tailored shopping experience.



Individual product detail pages with accurate routing and data rendering.

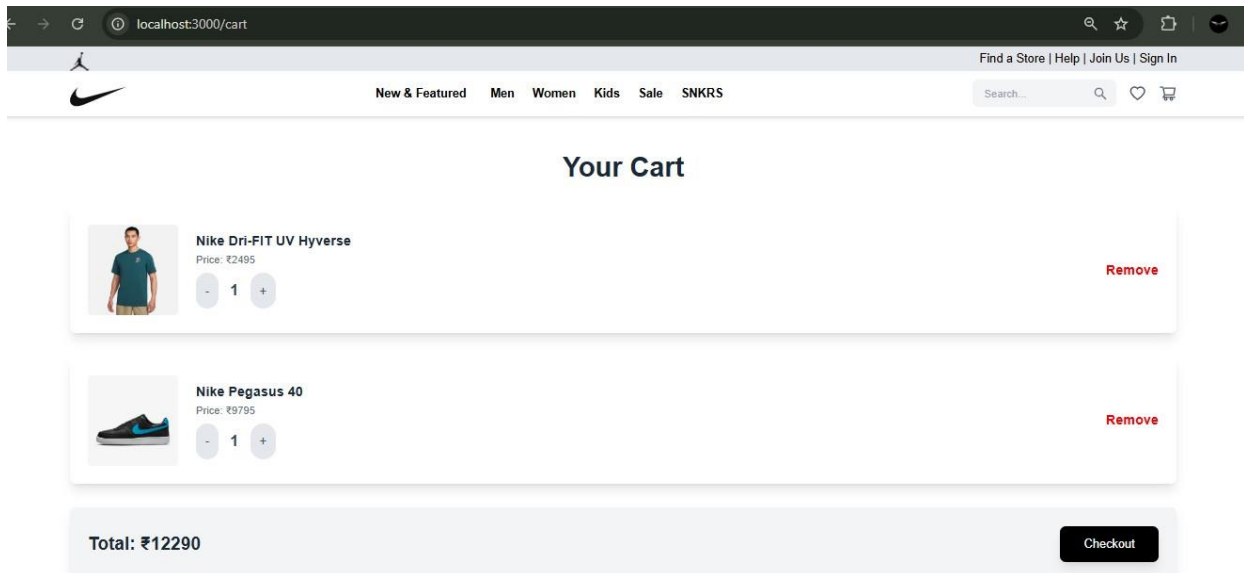


### Key Features of Dynamic Routes:

- **Unique Product URLs:** Every product is assigned a distinct identifier (ID or slug), allowing for automatically generated URLs (e.g., /product/[id]), ensuring each product gets its own dedicated page.
- **SEO Boost:** Server-rendered pages enhance SEO, ensuring your product pages rank higher in search results while delivering faster initial load times for a smoother user experience.
- **Rich Product Details:** Each dynamic page showcases comprehensive product information, including descriptions, high-quality images, price, stock availability, and customer reviews, providing users with everything they need to make informed purchasing decisions.
- **Scalable & Effortless:** With this approach, adding new products is a breeze. As new items are added to the system, corresponding pages are automatically generated, saving time and effort while maintaining a seamless shopping experience.

## 3. Cart Functionality:

The **Cart Functionality** is the heart of a flawless shopping experience, effortlessly managing the user's selected items and ensuring a smooth, hassle-free journey. It tracks every choice, dynamically updating quantities, prices, and totals, while providing a concise summary of costs, including taxes, shipping, and discounts. Users have full control to easily add or remove items, adjust quantities, and view real-time updates, all within an intuitive, sleek interface.

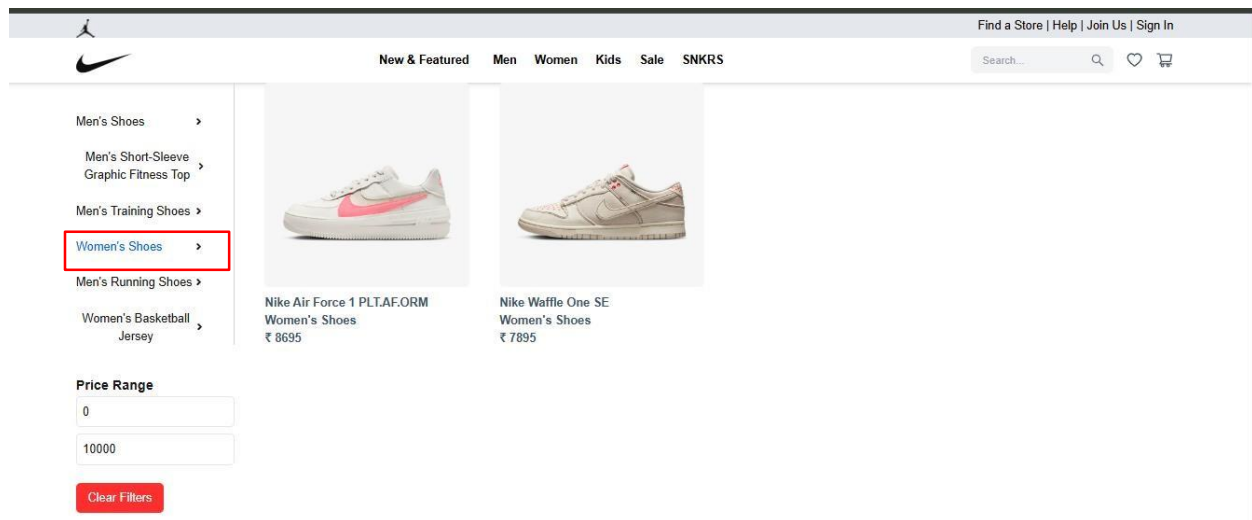
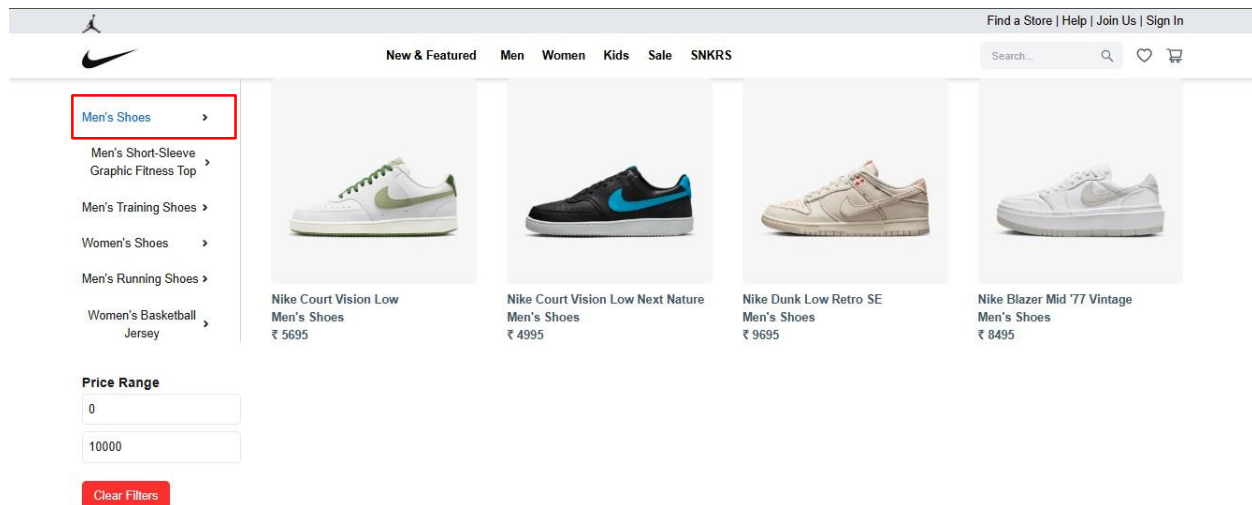


### **Detailed Description:**

- Users can add products to their cart directly from the product listing or detail page.
- The cart dynamically updates quantities and calculates the total cost, ensuring a real-time experience.
- A mini-cart displays a quick summary of selected items, while a detailed cart page offers options to edit or remove items.
- State management tools, such as React Context or Redux, are used to maintain the cart state across the application.
- Cart data persistence is achieved using local storage or session storage, ensuring the cart remains intact even if the page is refreshed

## **4. Working Categories Filter:**

The Filter Functionality allows users to easily narrow their search by categories, price, ratings, and more. Filters update in real-time, instantly adjusting the product display to match user preferences, ensuring a quick and efficient browsing experience. This intuitive system helps users find exactly what they're looking for, improving satisfaction and speeding up the shopping process.



## Technical Report: Building and Integrating Components for the Sale Page

### Overview:

This technical report provides an insightful summary of the steps taken to build and integrate the essential components for the Sale Page. It highlights the challenges encountered during development, the innovative solutions implemented, and the best practices adhered to throughout the process. The goal was to create a seamless, responsive, and user-friendly Sale Page that enhances the shopping experience.

# Steps Taken to Build and Integrate Components

## 1. Project Setup

- The project was initialized using **Next.js** to leverage server-side rendering and component-based development.
- **Sanity CMS** was configured to efficiently fetch and manage product data, ensuring smooth integration with the front-end.

## 2. Data Fetching and Integration

- A **Sanity CMS query** was defined to retrieve product data, including essential details such as product ID, name, category, price, and image.
- The `fetchData` function was implemented using Sanity's client library, allowing for asynchronous data fetching and storage in the products state.
- Unique product categories were extracted, and an **"All Categories"** option was added to ensure all products could be displayed by default.

## 3. UI Components Development

- A **responsive layout** was designed using **Tailwind CSS**, with distinct sections for the sidebar (including categories and price filters) and a product display grid.
- **Dynamic category filtering** was implemented, allowing users to filter products based on their selected category.
- A **price range filter** was added with input fields for minimum and maximum price values, giving users more control over their search.

## 4. Sorting Feature

- A **dropdown menu** was created to allow users to sort products by price, offering options such as **"Default"** and **"Price: High to Low"**.
- The sorting logic was integrated into the filtered product pipeline to ensure the displayed products were dynamically updated based on user preferences.

## 5. Responsive Sidebar

- A **toggle able sidebar** was developed for mobile screens, ensuring accessibility and a seamless user experience on all devices.
-

# Best Practices Followed During Development

## 1. Component Reusability:

- UI components were broken down into **smaller, reusable modules**, ensuring modularity, maintainability, and ease of updates.

## 2. Responsive Design:

- **Tailwind CSS** was used to create a **mobile-first design** that adapts seamlessly across all screen sizes, ensuring a great experience on both desktop and mobile.

## 3. Typescript for Type Safety:

- **Typescript** was leveraged to enforce type safety, reducing runtime errors and improving overall code reliability, making the development process smoother.

## 4. Code Readability and Comments:

- The code base was thoroughly documented, with **comments** explaining key logic, decisions, and functionality to improve readability and maintainability.

## 5. User Experience (UX):

- Focused on creating an **intuitive interface** with clear visual cues and easy navigation, ensuring that users could browse and purchase products with ease.

---

## Conclusion:

Through careful planning, problem-solving, and the use of cutting-edge technologies such as **Next.js**, **Sanity CMS**, and **Tailwind CSS**, the Sale Page was successfully built and integrated. The project followed best practices for modularity, performance, and user experience, ensuring a seamless shopping journey for users.