

## Day 4!

# Detailed Documentation for Dynamic Components and Functionalities

This documentation provides an in-depth analysis of the key functionalities for a dynamic marketplace, emphasizing modularity, reusability, and integration with Sanity CMS and API. Each feature is described comprehensively, followed by a conclusion summarizing the approach..

## Step 1: Functionalities Overview

The project implements the following core functionalities:

- 1. Product Listing Page**
- 2. Dynamic Route**
- 3. Cart Functionality**
- 4. Checkout**
- 5. Price Calculation**
- 6. Notification**

Each functionality contributes to building a responsive and scalable marketplace.

## Step 2: Functionalities in Detail

## 1.Product Listing Page

The image shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure for 'E-COMMERCE-WEBSITE' with folders 'src', 'app', 'components', 'contact', 'fonts', 'myAccount', 'product', and 'shop'. The 'components' folder is expanded, showing files like 'Blog.tsx', 'Footer.tsx', 'Hero.tsx', 'Heroone.tsx', 'Instagram.tsx', 'Navbar.tsx', 'Newarrivals.tsx', 'Pagination.tsx', 'ProductListing.tsx' (selected), and 'TopPicks.tsx'. The code editor shows the content of 'ProductListing.tsx', which includes a React component definition and JSX for a product grid. The code uses Tailwind CSS classes and a data prop for product information.

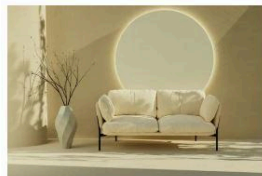
```

src > app > components > ProductListing.tsx > ProductListing
13  const ProductListing = () => {
14
15      93      <option value="Chair">Chair</option>
16      94      <option value="Table">Table</option>
17      95      </select>
18      96      </div>
19
20      97
21      98      { /* Product Grid */ }
22      99      <div className="grid sm:grid-cols-2 grid-cols-1 lg:grid-cols-4 gap-8 mt-4">
23      100      {currentProducts.length > 0 ? (
24      101      currentProducts.map((product) => (
25      102      <div
26      103      key={product._id}
27      104      className="border rounded-lg shadow-md p-4 hover:shadow-lg"
28      105      >
29      106      <Link href={` /product/${product.slug.current}`} >
30      107      {product.image && (
31      108      <Image
32      109      width={400}
33      110      height={400}
34      111      src={urlFor(product.image).url()}
35      112      alt={product.name}
36      113      className="w-full max-w-xs h-auto object-contain hover:scale-110 transition-trans
37      114      />
38      115      )}
39      116      <h3 className="text-lg font-medium text-gray-600 mt-4">
40      117      {product.name}
41      118      </h3>
42      119      <p className="text-lg font-semibold text-gray-800 mt-2">
43      120      ${product.price}
44      121      </p>

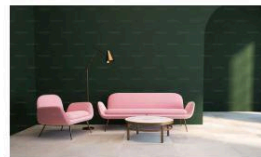
```



Pink Lounge Chair  
\$1600



Cozy Sofa  
\$520

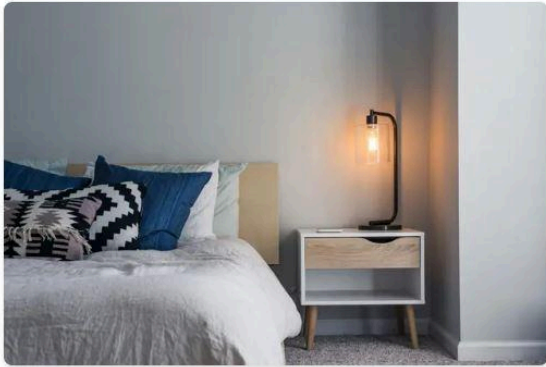


Leisure Sofa Chair Set  
\$1800



Solid Bed  
\$100

## 2. Dynamic Route



### Matilda Velvet Bed

600

★★★★★ | 5 Customer Review

A vibrant pink velvet chair with a retro-inspired design.

Category : Bed

Tags : Sofa, Chair, Bed, Table

Share : [f](#) [in](#) [t](#)

## Dynamic Route 2



### Nautilus Lounge Chair

1450

★★★★★ | 5 Customer Review

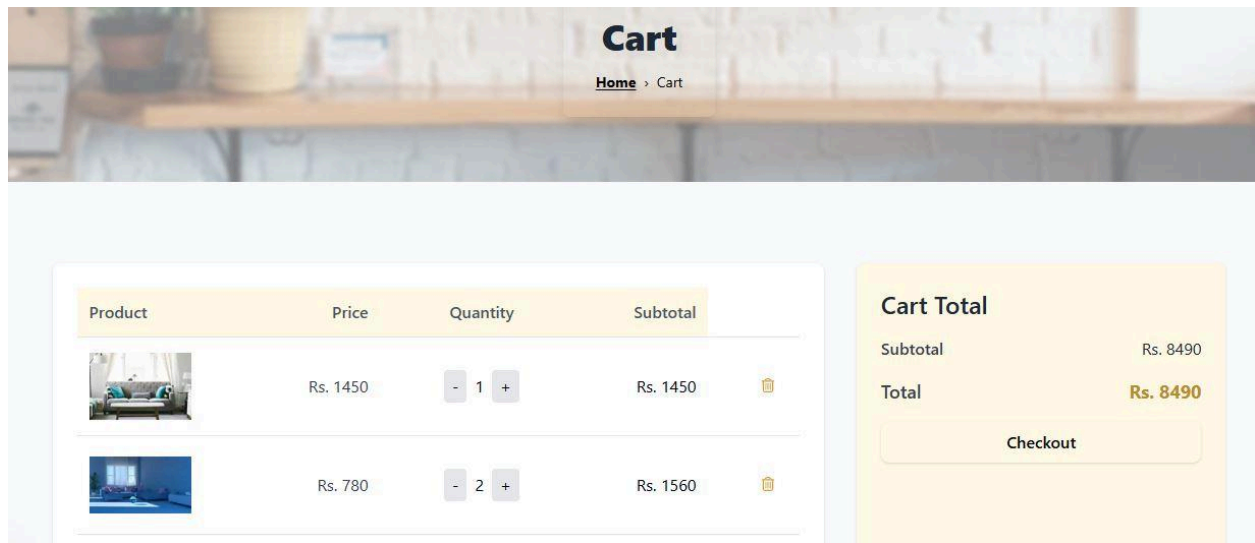
A lounge chair with a shell-inspired design and premium upholstery.

Category : Sofa

Tags : Sofa, Chair, Bed, Table

Share : [f](#) [in](#) [t](#)

### 3.Cart Functionality



### 4.Checkout Page

#### Billing Details

**First Name**


**Last Name**


**City**


**Address**

**Email**

#### Order Summary

  
Nautilus Lounge Chair x 1

  
Blue Bed x 2

  
Hans Wegner Style Three-Legged  
Shell Chair x 2

---

**Discount:** \$0

**SubTotal:** \$4990

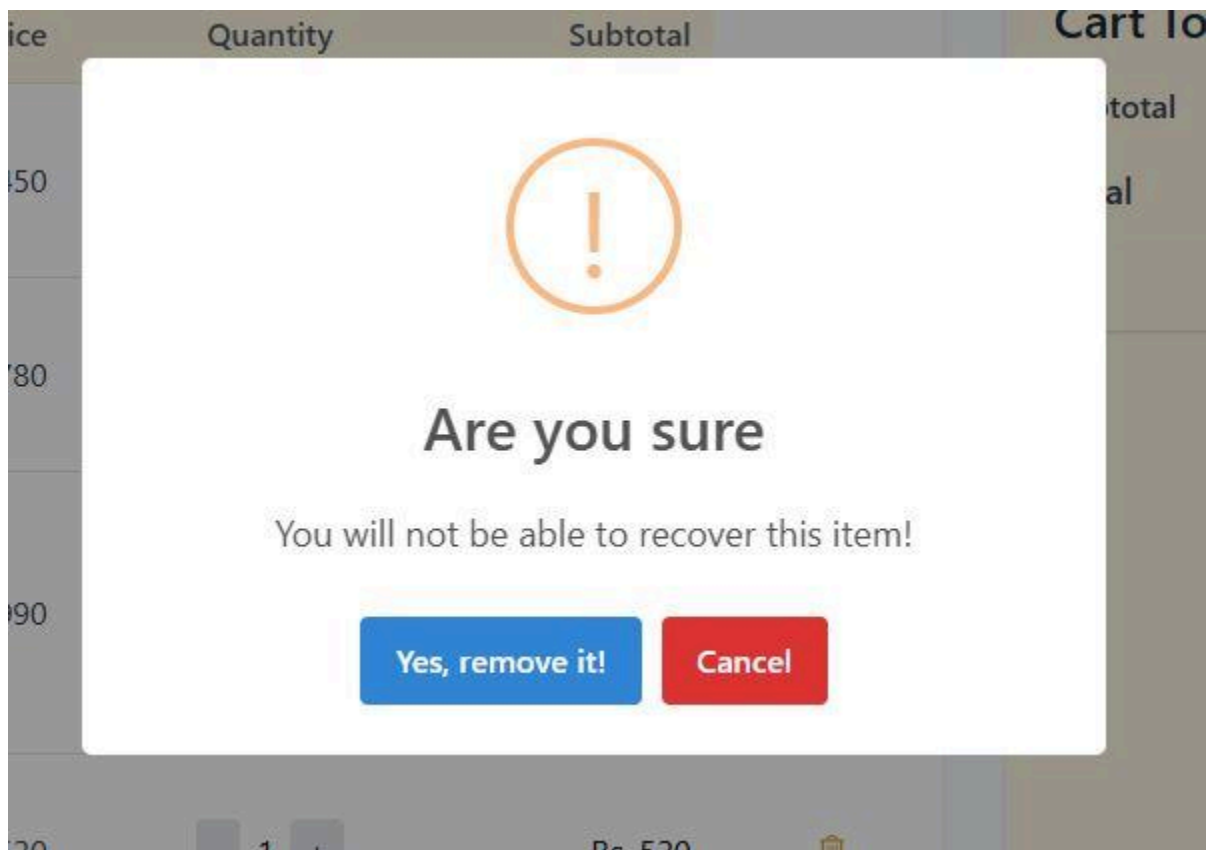
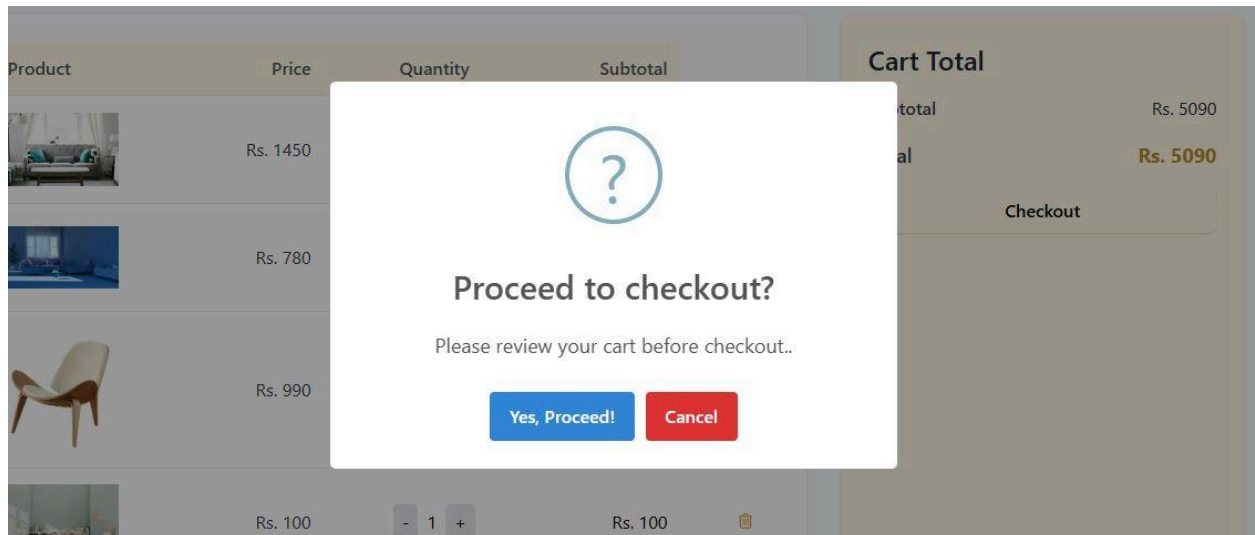
**Total:** \$4990.00

## 5. Price Calculation

```
5 //total amount in cartitems
6
7 const subTotal = cartItems.reduce(
8   (total, item) => total + item.price * item.stockLevel, 0)
9
```

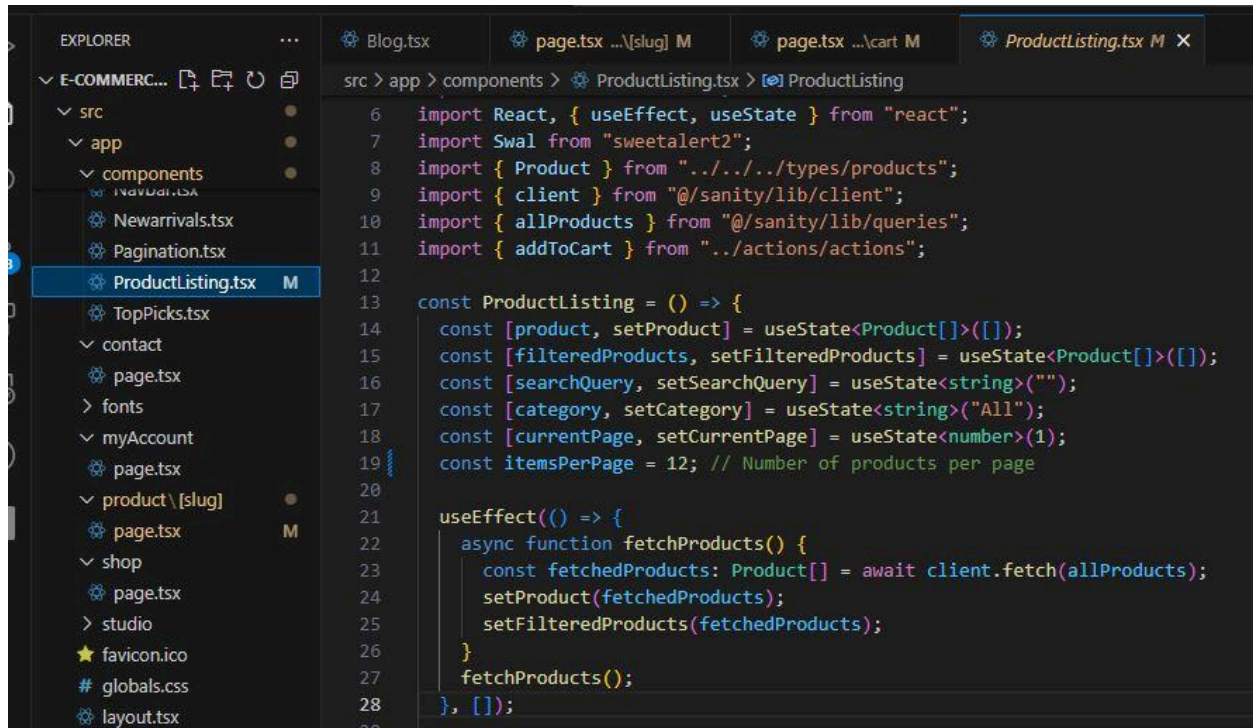
```
<div className="text-right pt-8">
  <p className="text-lg font-bold text-[#B88E2F] flex justify-between">
    Discount: <span className="font-medium text-black">${discount}</span>
  </p>
  <p className="text-lg font-bold text-[#B88E2F] flex justify-between">
    SubTotal:
    <span className="font-medium text-black">${subTotal}</span>
  </p>
  <p className="text-lg font-bold text-[#B88E2F] flex justify-between">
    Total:
    <span className="font-medium text-black"> ${subTotal.toFixed(2)}</span>
  </p>
</div>
```

## 6.Real Time Notification



### Step 3: Integration with Sanity CMS

Sanity CMS serves as the backend for managing and retrieving product data dynamically..



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure for 'E-COMMERC...' with a 'src' directory containing 'app', 'components', 'contact', 'fonts', 'myAccount', 'product\[slug]', 'shop', 'studio', 'favicon.ico', 'globals.css', and 'layout.tsx'. The 'components' directory is expanded, showing 'Newarrivals.tsx', 'Pagination.tsx', 'ProductListing.tsx' (selected), 'TopPicks.tsx', and 'contact'. The code editor shows the 'ProductListing.tsx' file with the following code:

```
6 import React, { useEffect, useState } from "react";
7 import Swal from "sweetalert2";
8 import { Product } from "../../types/products";
9 import { client } from "@sanity/lib/client";
10 import { allProducts } from "@sanity/lib/queries";
11 import { addToCart } from "../actions/actions";
12
13 const ProductListing = () => {
14   const [product, setProduct] = useState<Product[]>([]);
15   const [filteredProducts, setFilteredProducts] = useState<Product[]>([]);
16   const [searchQuery, setSearchQuery] = useState<string>("");
17   const [category, setCategory] = useState<string>("All");
18   const [currentPage, setCurrentPage] = useState<number>(1);
19   const itemsPerPage = 12; // Number of products per page
20
21   useEffect(() => {
22     async function fetchProducts() {
23       const fetchedProducts: Product[] = await client.fetch(allProducts);
24       setProduct(fetchedProducts);
25       setFilteredProducts(fetchedProducts);
26     }
27     fetchProducts();
28   }, []);
```



## Conclusion

This documentation outlines a comprehensive approach to building dynamic and responsive marketplace components. By leveraging Sanity CMS for backend management and modular frontend development techniques, the application achieves scalability, efficiency, and a superior user experience.

Each functionality—from product listing to inventory management—plays a vital role in delivering a professional marketplace that meets real-world needs. Future enhancements, such as integrating advanced analytics or AI-based recommendations, can further elevate the platform...