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

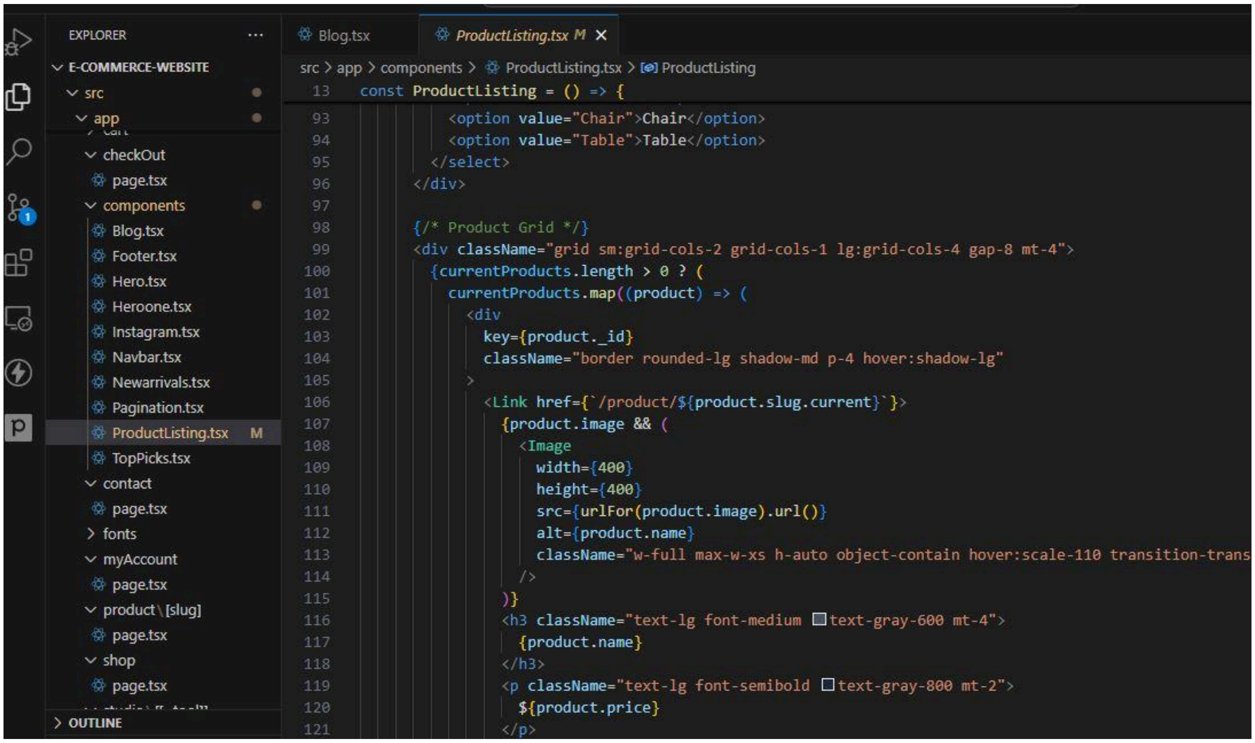
The project implements the following core functionalities:

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.

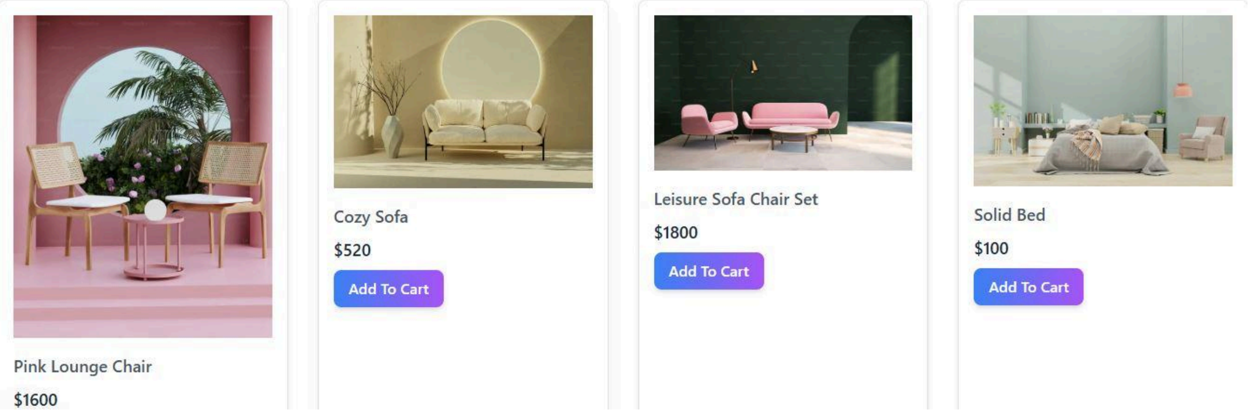
Step 2: Functionalities in Detail

1.Product Listing Page

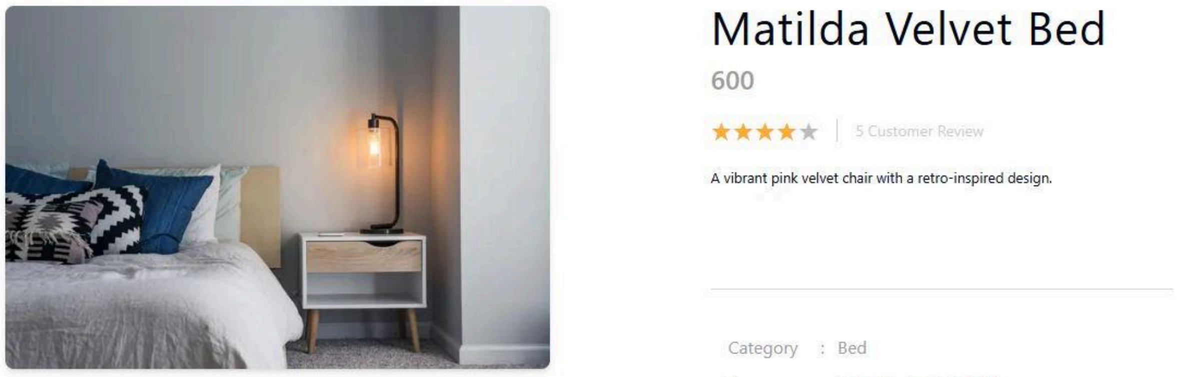


```
const ProductListing = () => {
          <option value="Chair">Chair</option>
          <option value="Table">Table</option>
        </select>
      </div>

      {/* Product Grid */}
      <div className="grid sm:grid-cols-2 grid-cols-1 lg:grid-cols-4 gap-8 mt-4">
        {currentProducts.length > 0 ? (
          currentProducts.map((product) => (
            <div
              key={product._id}
              className="border rounded-lg shadow-md p-4 hover:shadow-lg"
            >
              <Link href={`/product/${product.slug.current}`}>
                {product.image && (
                  <Image
                    width={400}
                    height={400}
                    src={urlFor(product.image).url()}
                    alt={product.name}
                    className="w-full max-w-xs h-auto object-contain hover:scale-110 transition-trans
                  />
                )}
                <h3 className="text-lg font-medium text-gray-600 mt-4">
                  {product.name}
                </h3>
                <p className="text-lg font-semibold text-gray-800 mt-2">
                  ${product.price}
                </p>
```



2. Dynamic Route



Matilda Velvet Bed

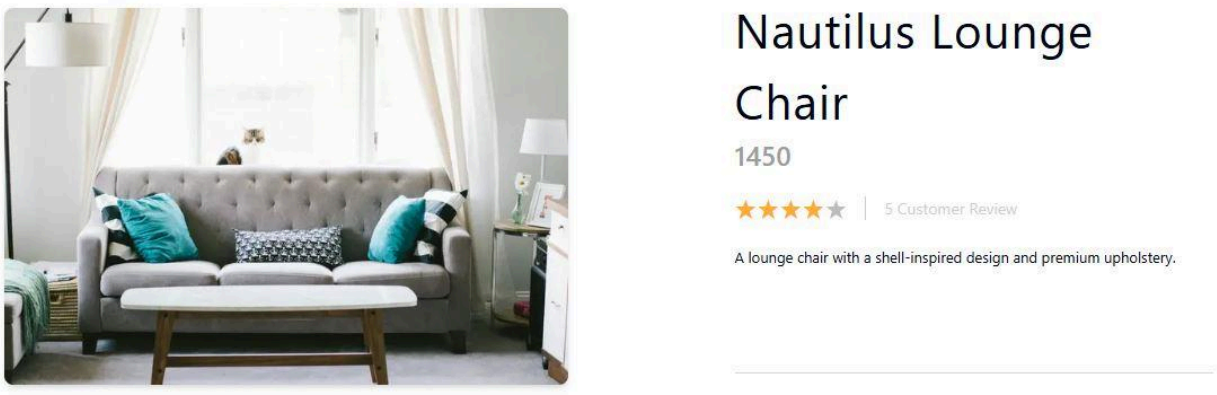
600

★★★★☆ | 5 Customer Review

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

Category	:	Bed
Tags	:	Sofa, Chair, Bed, Table
Share	:	

Dynamic Route 2



Nautilus Lounge Chair

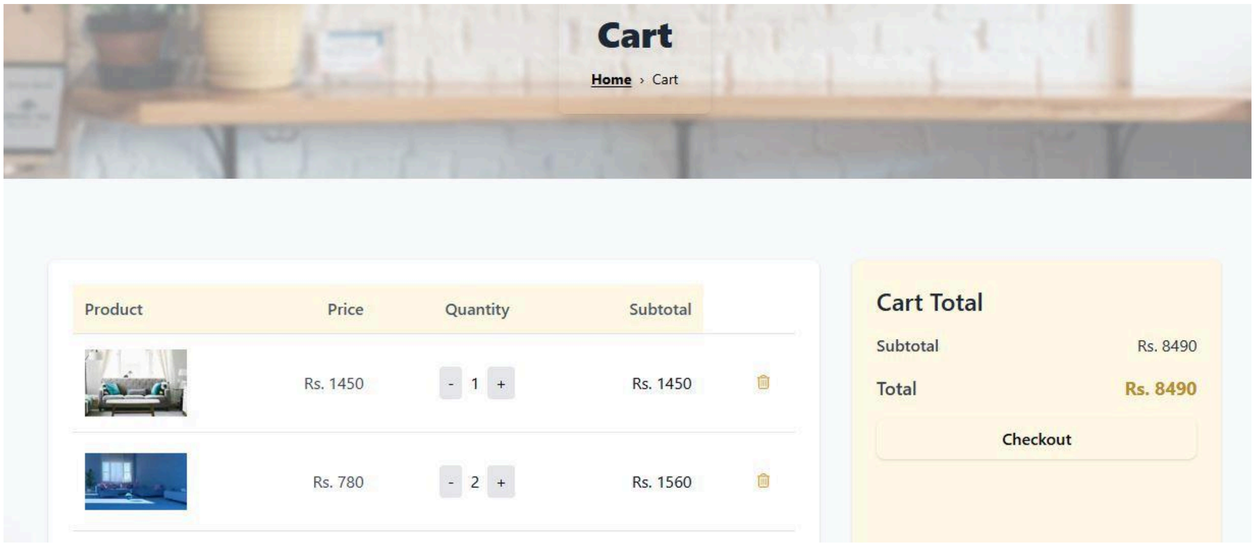
1450

★★★★☆ | 5 Customer Review

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

Category	:	Sofa
Tags	:	Sofa, Chair, Bed, Table
Share	:	

3.Cart Functionality



Cart

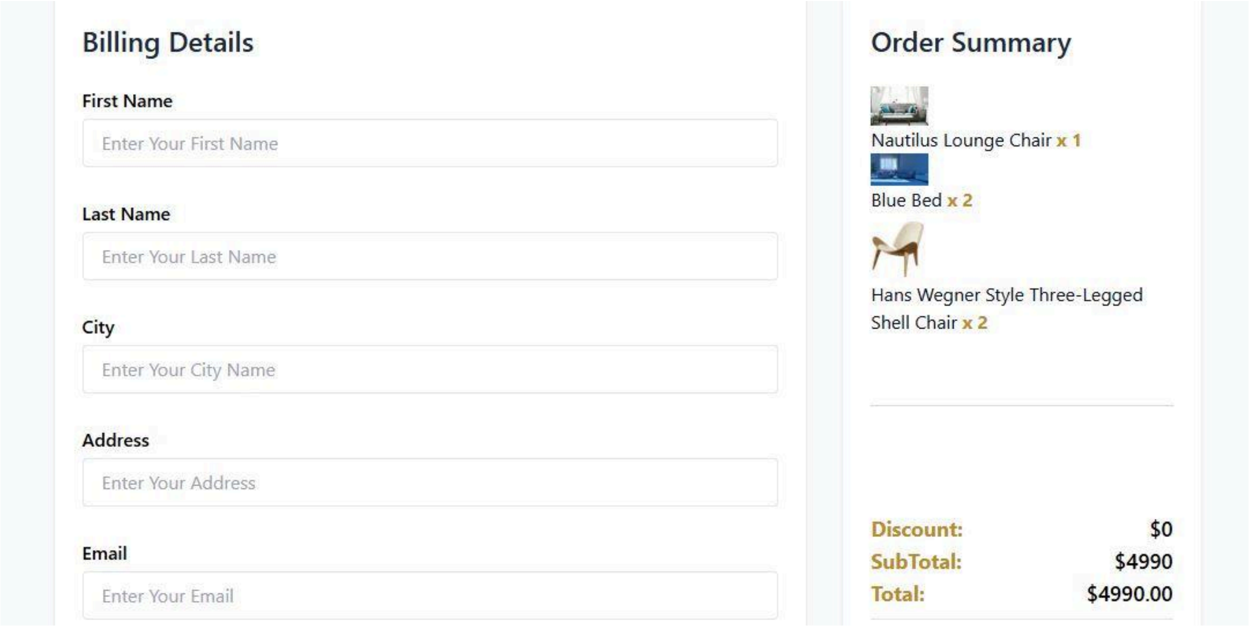
Home › Cart

Product	Price	Quantity	Subtotal
	Rs. 1450	- 1 +	Rs. 1450
	Rs. 780	- 2 +	Rs. 1560

Cart Total

Subtotal	Rs. 8490
Total	Rs. 8490

4.Checkout Page



Billing Details

First Name

Enter Your First Name

Last Name

Enter Your Last Name

City

Enter Your City Name

Address

Enter Your Address

Email

Enter Your Email

Order Summary

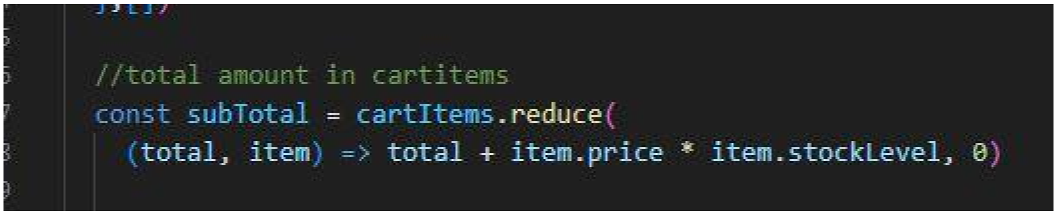
Nautilus Lounge Chair x 1

Blue Bed x 2

Hans Wegner Style Three-Legged Shell Chair x 2

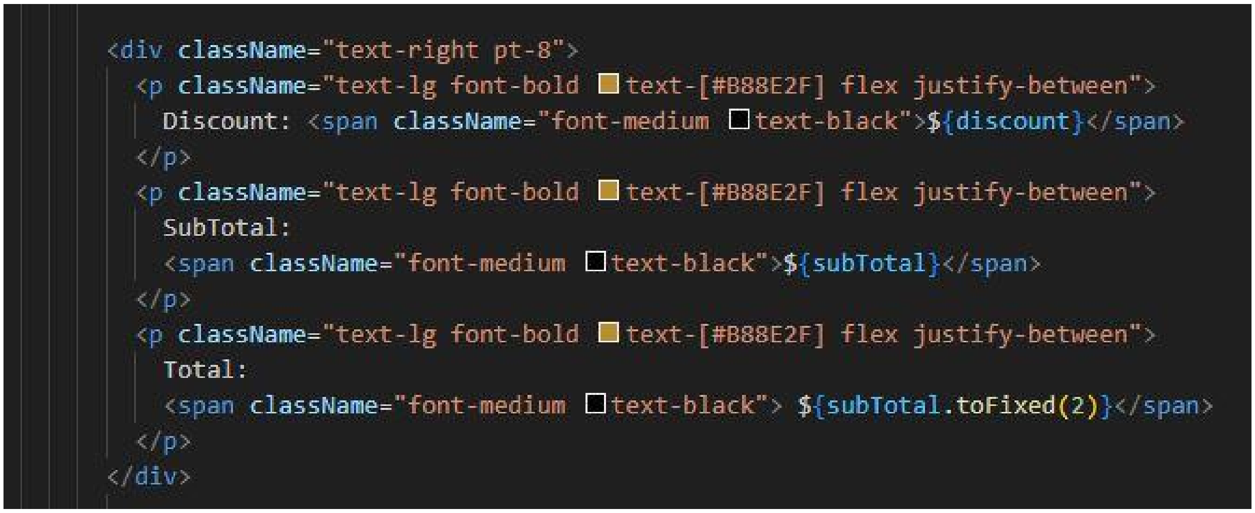
Discount:	$0
SubTotal:	$4990
Total:	$4990.00

5.Price Calculation



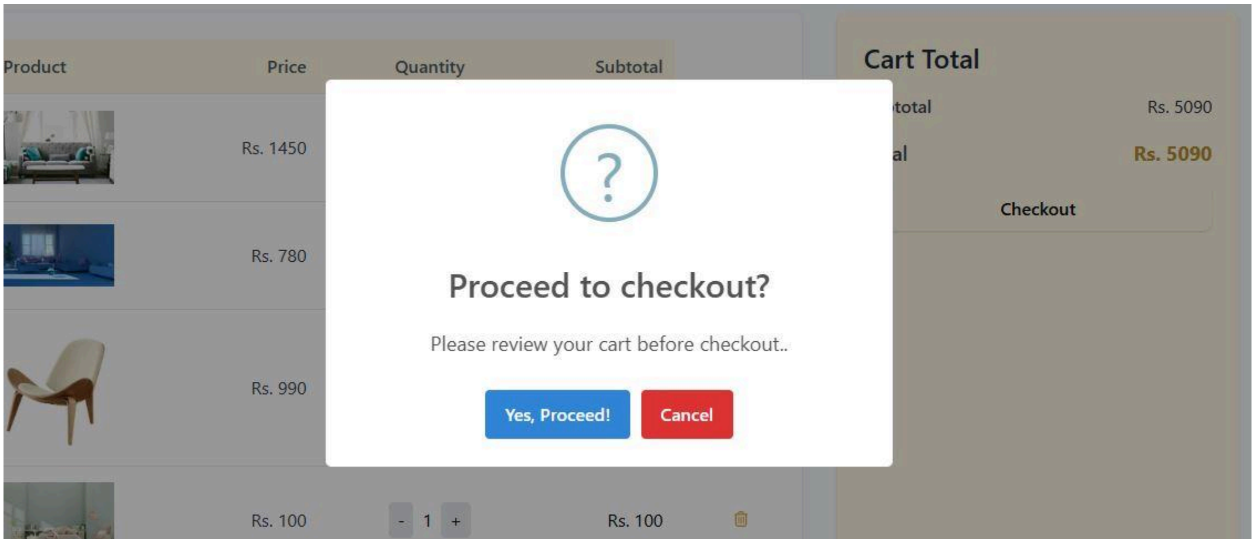
```
},[])

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



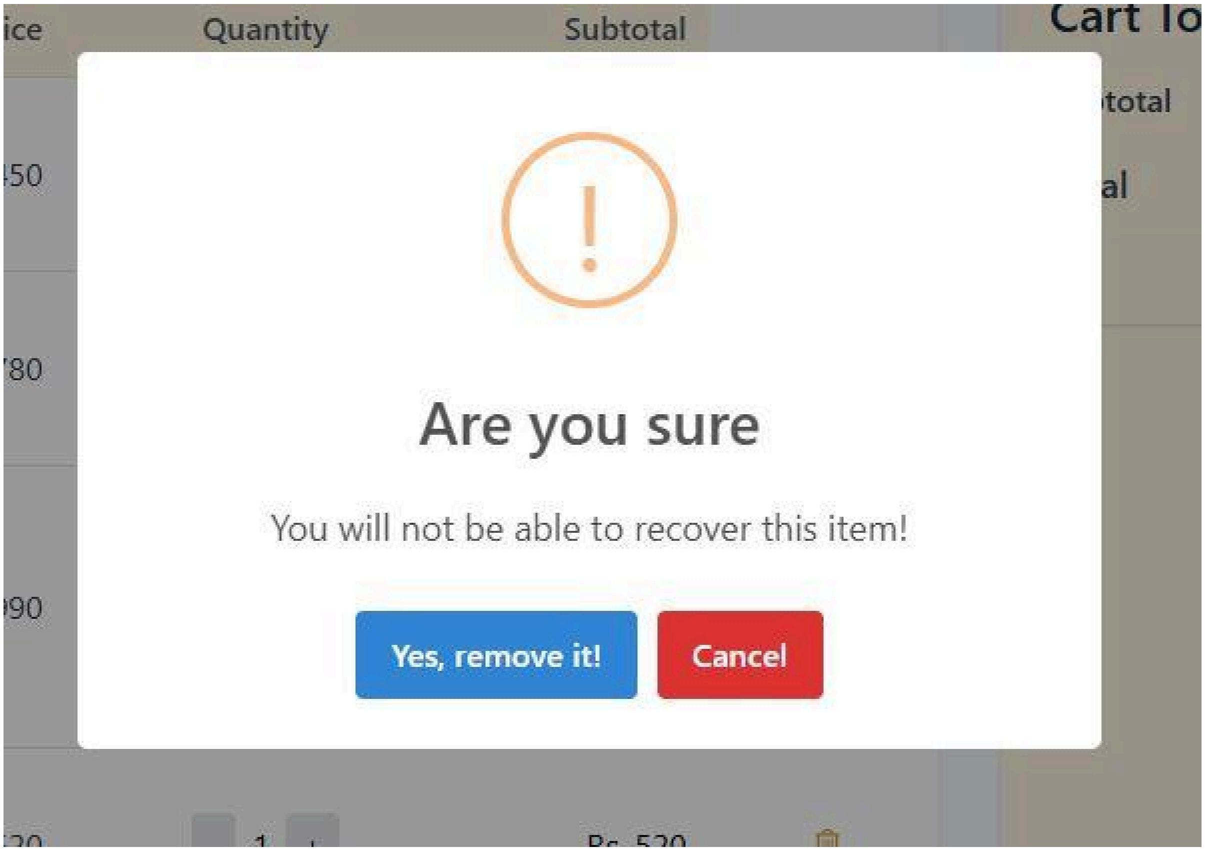
```
<div className="text-right pt-8">
  <p className="text-lg font-bold 🟨text-[#B88E2F] flex justify-between">
    Discount: <span className="font-medium ■text-black">${discount}</span>
  </p>
  <p className="text-lg font-bold 🟨text-[#B88E2F] flex justify-between">
    SubTotal:
    <span className="font-medium ■text-black">${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



Proceed to checkout?

Please review your cart before checkout..

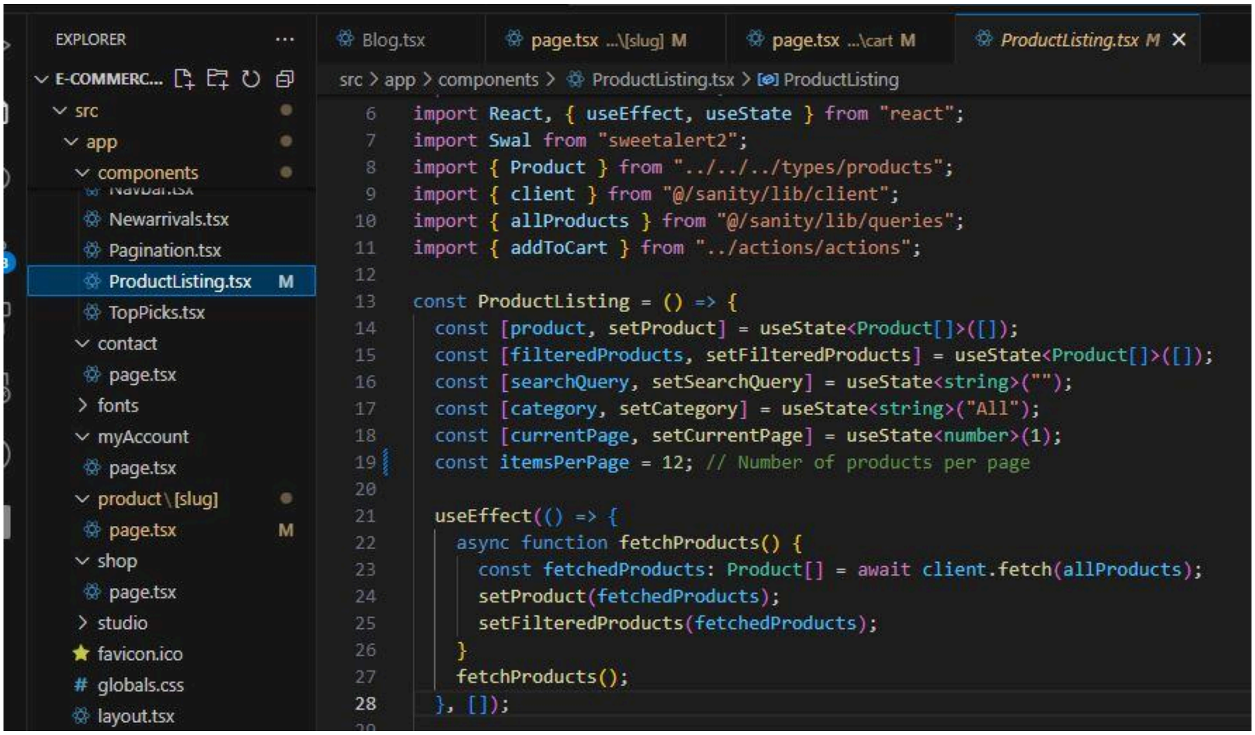


Are you sure

You will not be able to recover this item!

Step 3: Integration with Sanity CMS

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



```
import React, { useEffect, useState } from "react";
import Swal from "sweetalert2";
import { Product } from "../../../types/products";
import { client } from "@/sanity/lib/client";
import { allProducts } from "@/sanity/lib/queries";
import { addToCart } from "../actions/actions";

const ProductListing = () => {
  const [product, setProduct] = useState<Product[]>([]);
  const [filteredProducts, setFilteredProducts] = useState<Product[]>([]);
  const [searchQuery, setSearchQuery] = useState<string>("");
  const [category, setCategory] = useState<string>("All");
  const [currentPage, setCurrentPage] = useState<number>(1);
  const itemsPerPage = 12; // Number of products per page

  useEffect(() => {
    async function fetchProducts() {
      const fetchedProducts: Product[] = await client.fetch(allProducts);
      setProduct(fetchedProducts);
      setFilteredProducts(fetchedProducts);
    }
    fetchProducts();
  }, []);
```

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,efficiently,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…