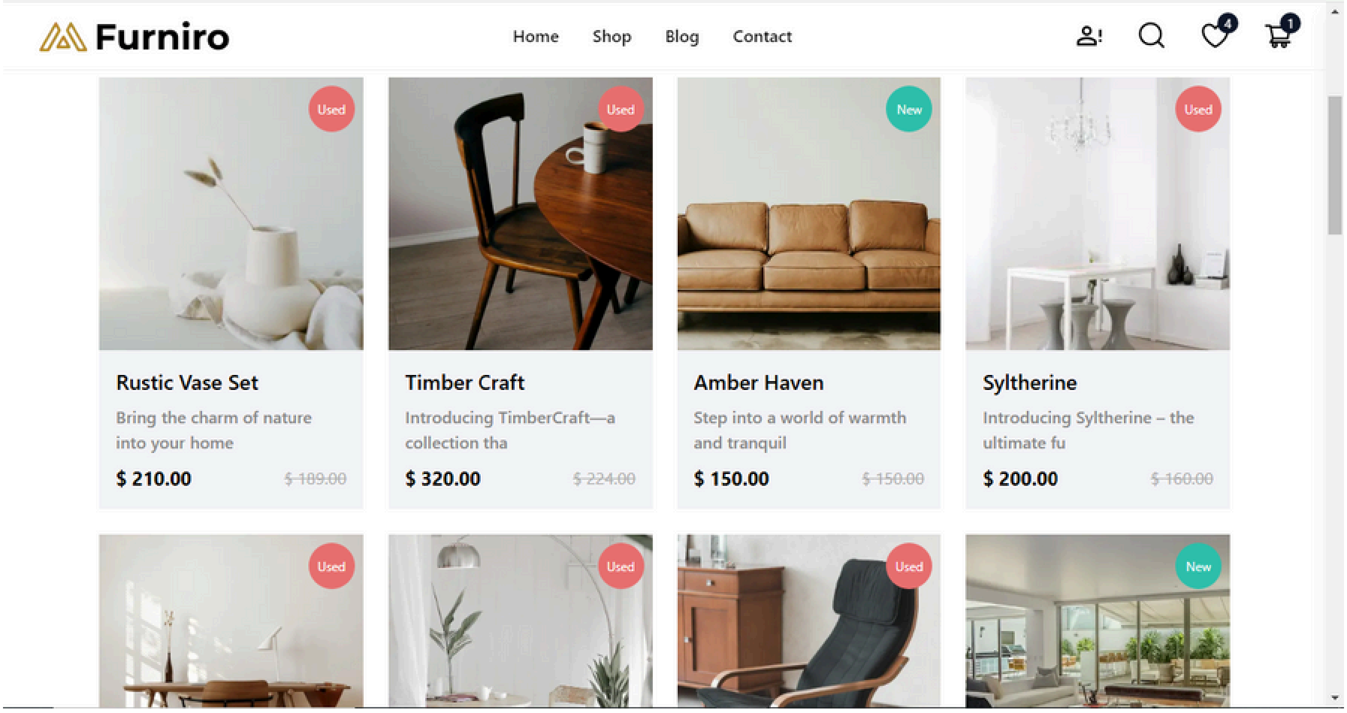
Day 4-Dynamic Frontened Components- Furniro

1. The product Listing with dynamic data



Rustic Vase Set

Bring the charm of nature into your home

$ 210.00 ~~$ 189.00~~

Timber Craft

Introducing TimberCraft—a collection tha

$ 320.00 ~~$ 224.00~~

Amber Haven

Step into a world of warmth and tranquil

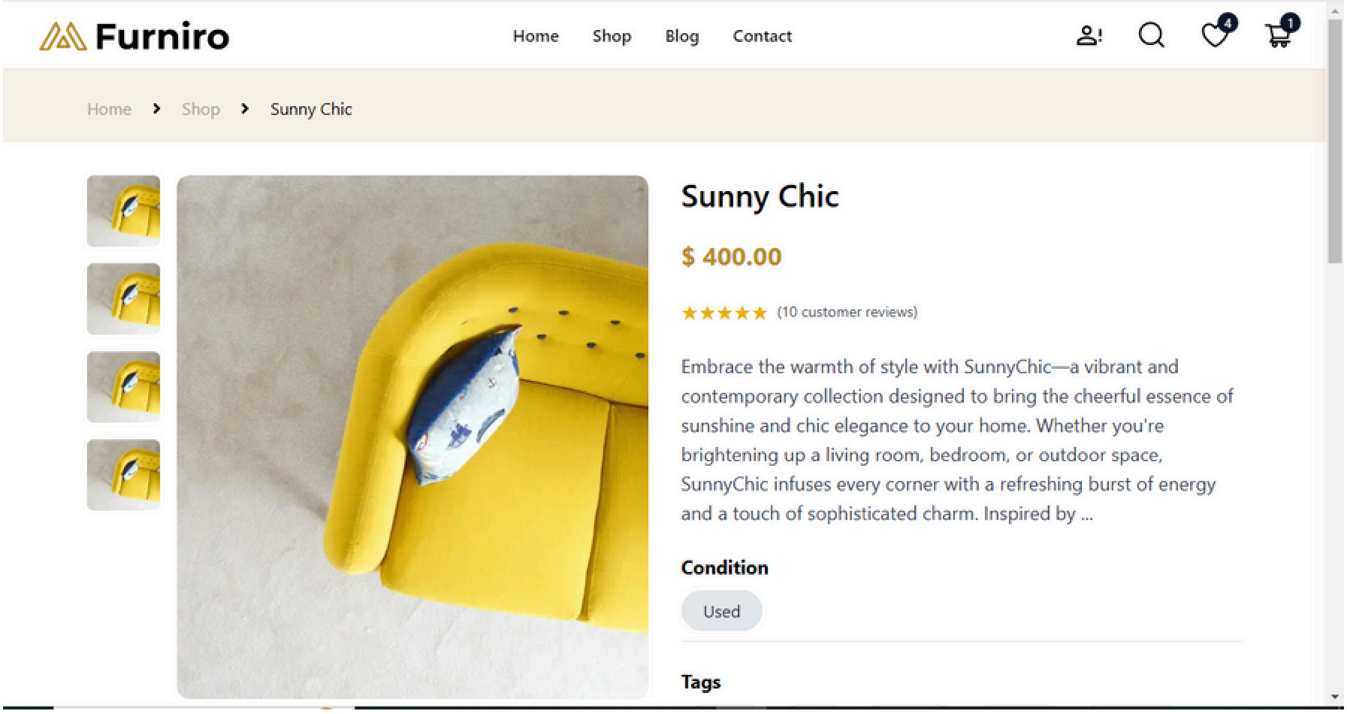
$ 150.00 ~~$ 150.00~~

Syltherine

Introducing Syltherine – the ultimate fu

$ 200.00 ~~$ 160.00~~

1. Product detail page



Sunny Chic

$ 400.00

★★★★★ (10 customer reviews)

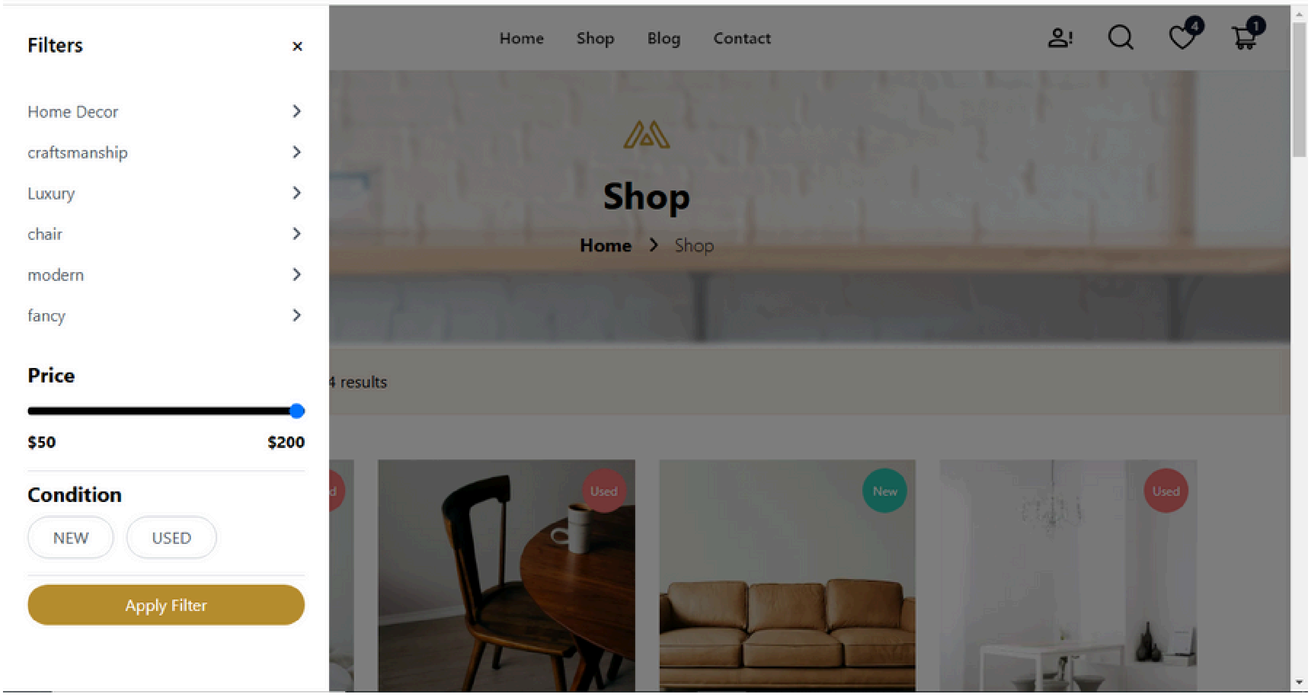
Embrace the warmth of style with SunnyChic—a vibrant and contemporary collection designed to bring the cheerful essence of sunshine and chic elegance to your home. Whether you're brightening up a living room, bedroom, or outdoor space, SunnyChic infuses every corner with a refreshing burst of energy and a touch of sophisticated charm. Inspired by ...

Condition

Used

Tags

1. filter, pagination, category filter



Product Listing

To display a list of products, I first created a GROQ query in Sanity Vision. I then fetched the data in my code using useEffect and useState. Inside the return statement, I mapped the variable stored in useState to render the product cards. After designing the product cards, the products displayed correctly on the output. One challenge I faced was creating a skeleton for the product cards to display during the loading state.

Product Detail Page

To create the product detail page, I set up a dynamic route. I retrieved the id parameter from the route and used it to construct a GROQ query in Sanity. The query includes a condition to fetch and display the product whose id matches the route parameter.

Additional Functionalities

Filter Functionality: I implemented filtering using useState. I added conditions for filter items to the GROQ query and rendered the filtered products by mapping over them to display the output.

Search Functionality: A search bar was added to the navbar. The search query matches the input value with product titles stored in Sanity and displays the corresponding results.

Pagination: Pagination logic was implemented to handle large datasets. If the total number of items (data.length) exceeds 20, products from the 21st item onwards are displayed on the next page, accessible through a second tab.