

Day 4 - Dynamic Frontend Components General E-commerce Website

Step 1: Overview of Core Functionalities

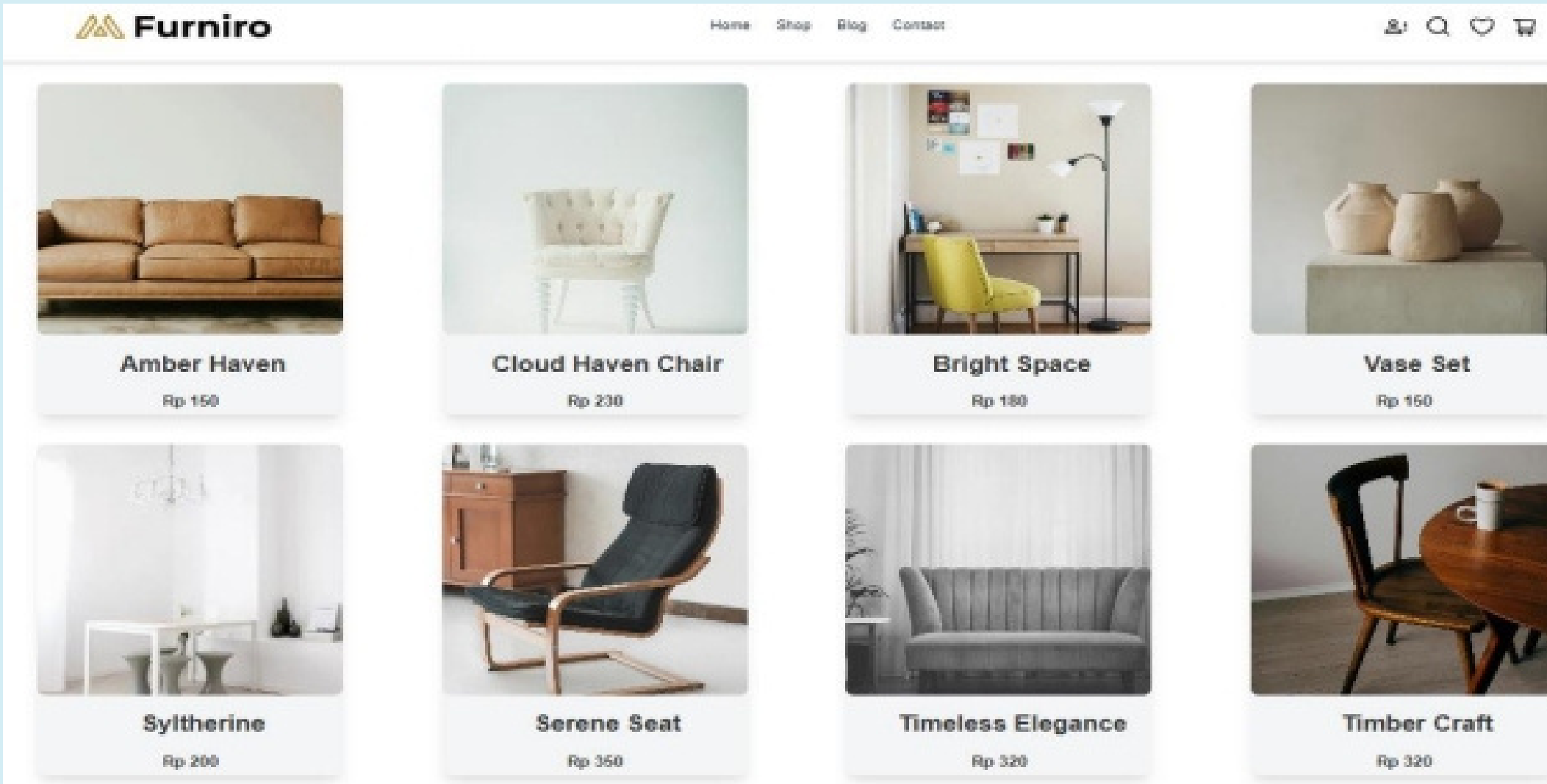
The project incorporates the following essential features to create a responsive, scalable, and efficient marketplace:

1. Product Listing Page
2. Dynamic Routing
3. Shopping Cart Functionality
4. Secure Checkout Process
5. Advanced Price Calculation
6. Product Comparison Tool

Step 2: Detailed Functionalities

1. Product Listing Interface

The Product Listing Interface acts as the primary platform for users to browse through available products. It dynamically pulls data from Sanity CMS to present products in an organized, visually appealing manner. To enhance user experience, the interface offers flexible display options, allowing users to toggle between grid and list layouts, ensuring an optimal viewing experience based on their preference.



Detailed Functionality Description

1. Product Listing Page

Advanced Sorting and Filtering

This feature allows users to sort and filter products by attributes like price, category, and popularity, enhancing the browsing experience and helping users find products quickly.

Efficient Pagination:

This feature enables smooth navigation and ensures optimal performance when dealing with large datasets, improving the overall user experience.

Responsive Design:

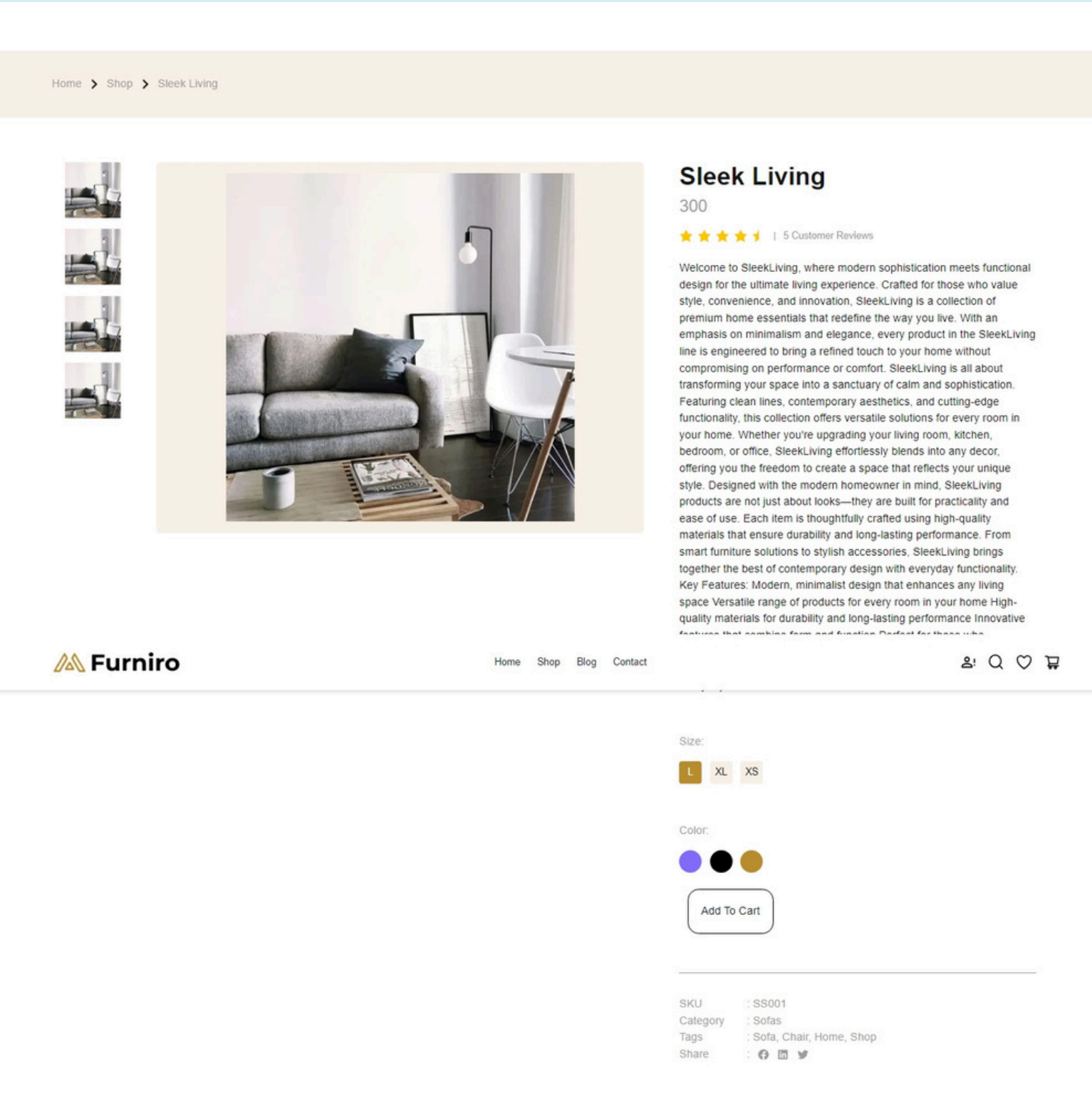
Ensures seamless compatibility across various devices, from desktops to mobile phones, providing a consistent and smooth user experience.

Real-Time Integration with Sanity CMS:

Facilitates instant synchronization of product updates from the backend, ensuring that the displayed data is always up-to-date.

2. Dynamic Routing

Enables the creation of dedicated product detail pages, providing users with comprehensive, structured, and easily accessible information about each product.



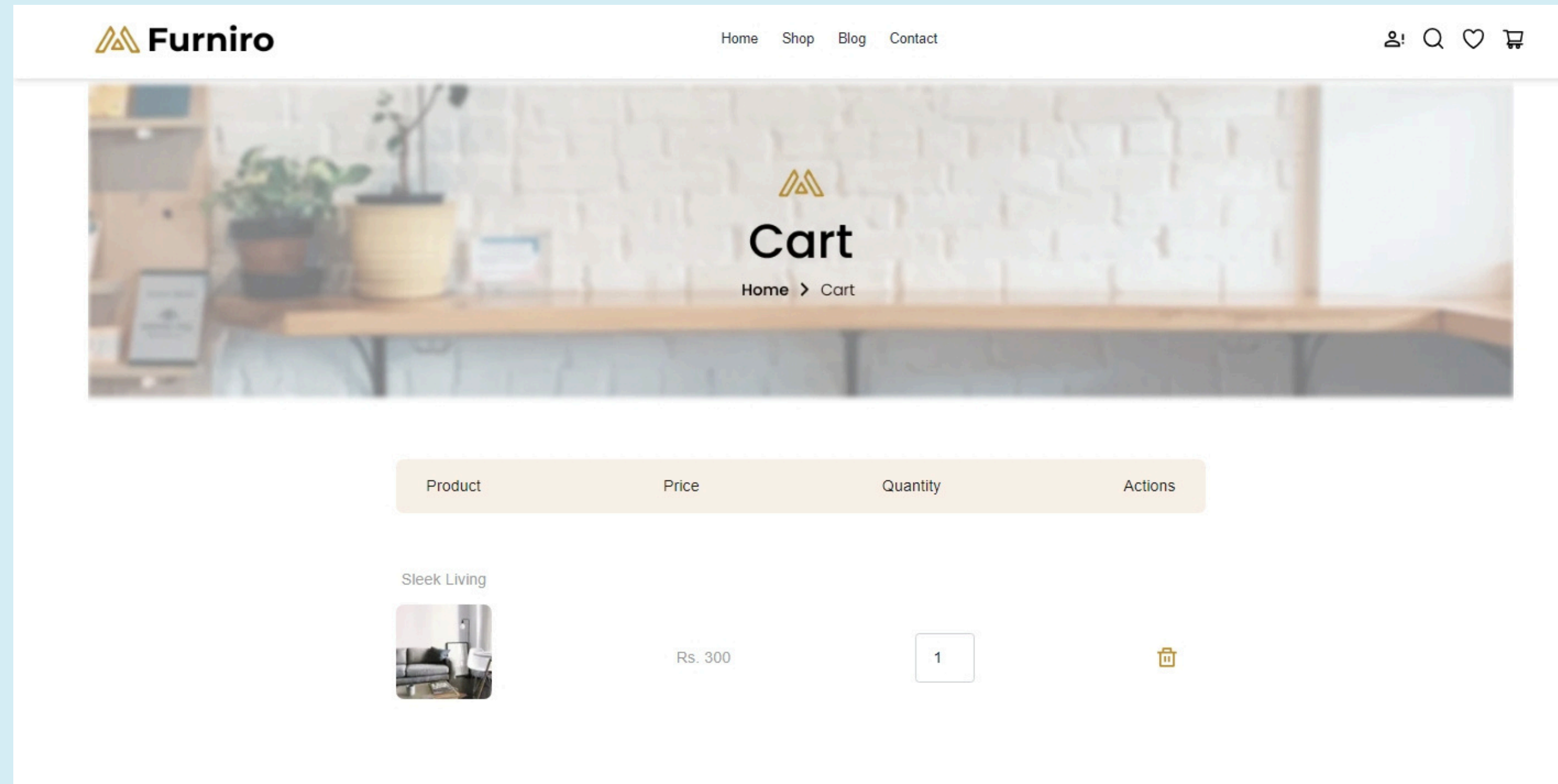
Product Pages with Dynamic Routing

1. Each product is assigned a unique ID or slug, which is used to dynamically generate its dedicated URL (e.g., /product/[id]).
2. These pages are server-rendered to optimize SEO and ensure faster initial load times, enhancing the overall user experience.
3. Dynamic routing allows for the seamless display of key product details, including descriptions, images, pricing, stock availability, and customer reviews.
4. This scalable solution ensures that new products are automatically assigned corresponding pages, eliminating the need for manual updates.

Seamless Cart Functionality

1. The cart functionality simplifies the shopping experience by efficiently managing and tracking the user's selected items
2. It handles item selection, calculates the total cost, and offers a clear summary of the user's choices, ensuring a smooth and hassle-free checkout process.

3.This feature boosts user satisfaction by providing real-time updates on product selections and the overall cost.



Checkout Process

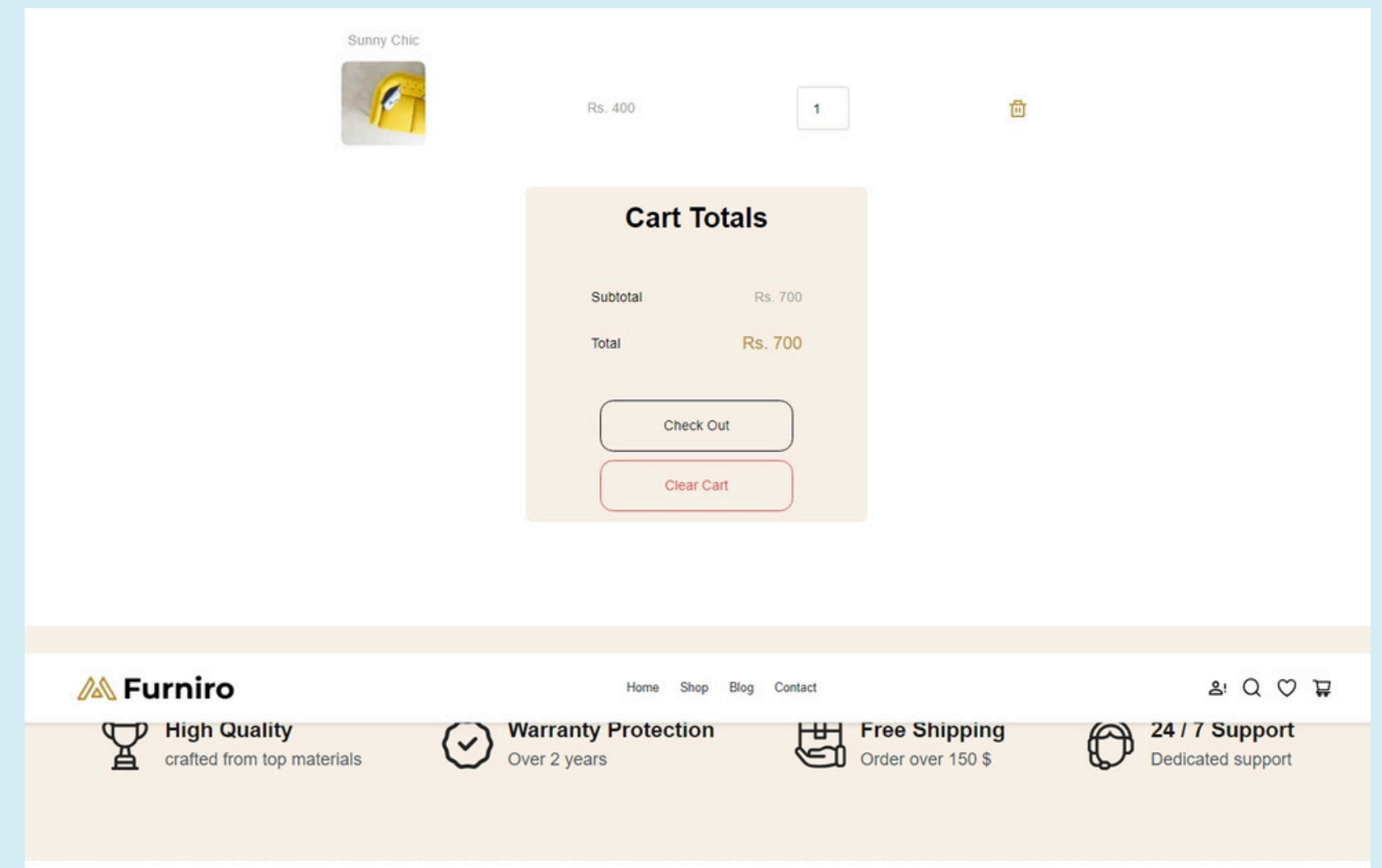
1.The checkout process is organized into clear steps: Billing Details and Payment Information, ensuring a smooth and efficient user experience.

2.A Dynamic Progress Tracker visually guides users through each step, keeping them informed throughout the process.

3.Input Validation is implemented at each stage to ensure all required fields are correctly filled, reducing the chances of errors during order submission.

4.Although payment integration can be initially simulated, the checkout system is designed to support secure payment gateways like Stripe or PayPal for safe transactions.

5.At the final stage, Order Summaries are presented, allowing users to review and confirm their order details before proceeding to the "Place Order" step, ensuring accuracy before finalizing the purchase.



Cart Functionality Breakdown

Remove Product from Cart

1. Removes a product using its unique ID, synchronizing the changes with the cart state and localStorage for seamless updates.

Empty the Cart

2. Clears all products from the cart, resets the cart state, and removes all cart data from localStorage to ensure a fresh start.

Adjust Product Quantity

3. Updates the quantity of a product, with validation to prevent invalid values (e.g., negative or zero). Changes are reflected in the cart state and localStorage for consistency.

```
1  const removeFromCart = (id: string) => {
2    const updatedCart = cartItems.filter(item => item.product._id !== id);
3    setCartItems(updatedCart);
4    localStorage.setItem("cart", JSON.stringify(updatedCart.map(item => item.product._id)));
5  };
6
7  const clearCart = () => {
8    setCartItems([]);
9    localStorage.removeItem("cart");
10 };
11
12 const updateQuantity = (id: string, newQuantity: number) => {
13   if (newQuantity <= 0) return; // Prevent negative or zero quantity
14   const updatedItems = cartItems.map(item =>
15     item.product._id === id ? { ...item, quantity: newQuantity } : item
16   );
17   setCartItems(updatedItems);
18   localStorage.setItem("cart", JSON.stringify(updatedItems.map(item => item.product._id)));
19 };
```


Placing the Order

Review and Confirmation

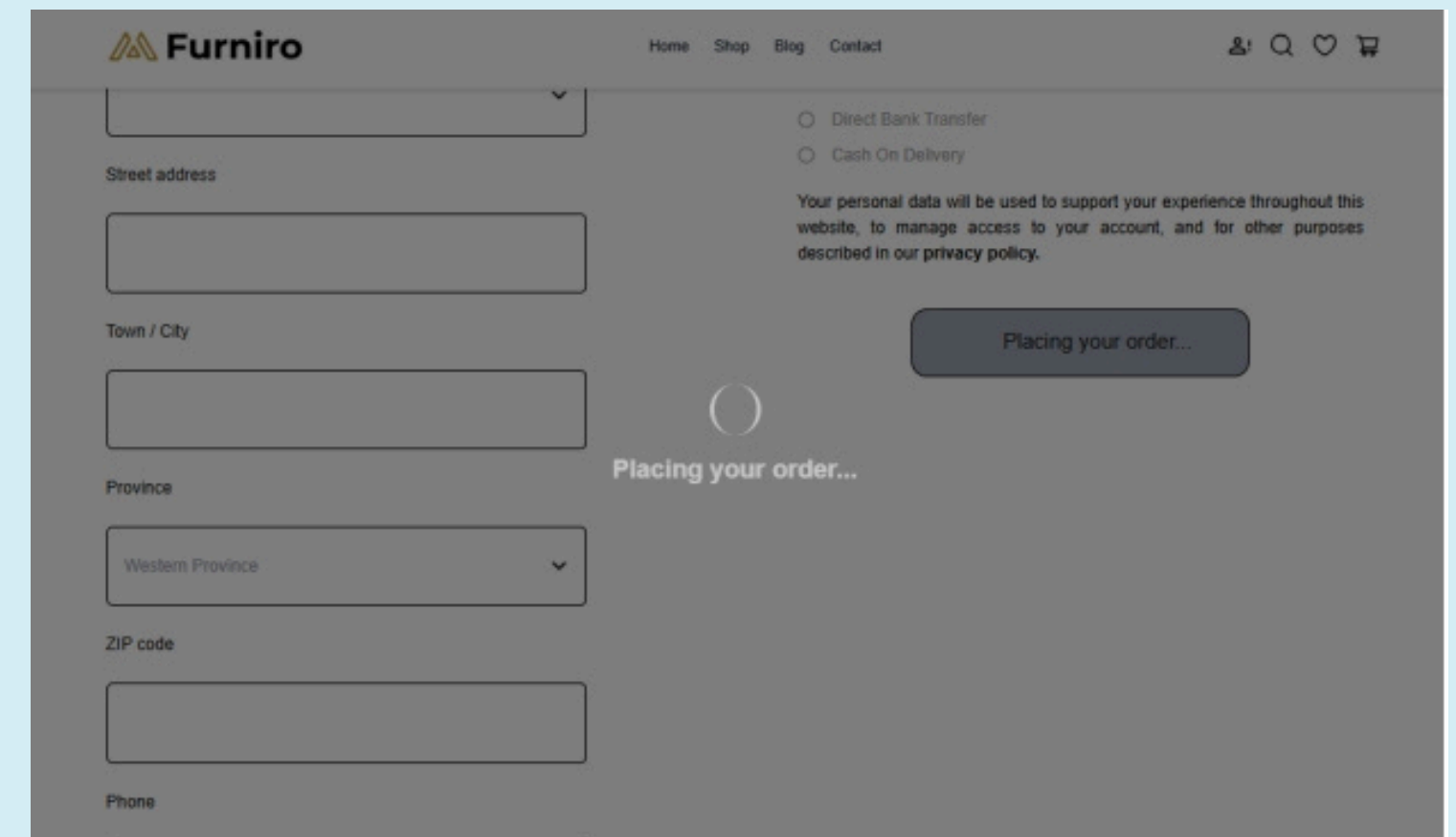
1. Offers users a detailed summary of their order, including item details, billing, and payment information, allowing them to verify accuracy before finalizing the purchase.

Secure Order Submission

2. Sends the finalized order details to the backend for secure processing and storage, ensuring a safe transaction.

Real-Time Acknowledgment

3. Displays a success notification or redirects users to a confirmation page, reassuring them that their order has been successfully placed.



The screenshot displays the Furniro checkout interface. At the top, the Furniro logo is on the left, and navigation links for Home, Shop, Blog, and Contact are on the right. User icons for profile, search, heart, and cart are also present. The main form area includes a dropdown menu, a text input for 'Street address', a text input for 'Town / City', a text input for 'Province' (currently showing 'Western Province'), a text input for 'ZIP code', and a text input for 'Phone'. To the right of the form, there are two radio button options: 'Direct Bank Transfer' and 'Cash On Delivery'. Below these is a privacy notice: 'Your personal data will be used to support your experience throughout this website, to manage access to your account, and for other purposes described in our [privacy policy](#).' A large button labeled 'Placing your order...' is positioned to the right of the form. A loading spinner and the text 'Placing your order...' are visible in the center of the page, indicating the order is being processed.

Connecting with Sanity CMS

Sanity CMS acts as the backend, enabling dynamic management and retrieval of product data.

```
1  const ShopPage = () => {
2    const [products, setProducts] = useState<Product[]>([]);
3    const [loading, setLoading] = useState<boolean>(true); // Loading state
4
5    const getProducts = async () => {
6      try {
7        const productsData = await client.fetch(
8          `*[_type=="product"]{
9            _id,
10           title,
11           price,
12           productImage
13         }`
14        );
15        setProducts(productsData);
16      } catch (error) {
17        console.error("Error fetching products:", error);
18      } finally {
19        setLoading(false);
20      }
21    };
22
23    useEffect(() => {
24      getProducts();
25    }, []);
```

Comprehensive Overview

1. Sanity CMS stores products, categories, and metadata, allowing admins to easily update content without the need to alter the codebase.

- 2.A robust client efficiently queries Sanity CMS, ensuring dynamic and reliable data fetching for a smooth experience.
- 3.Real-time updates in the CMS are immediately reflected on the frontend, providing a seamless content management workflow.
- 4.The integration is highly scalable, enabling the easy addition of new data types or fields as the marketplace grows and evolves.

Final Thinking

The approach outlined in this guide creates a dynamic and responsive marketplace, ensuring scalability and performance. Future upgrades, such as AI-driven recommendations, will further enhance the platform's capabilities, making it ideal for long-term growth

MADE BY
Nabila Bannay Khan