Day 4 - Dynamic Frontend Components – Furniva

By Wania Azam

Summary:

This report explains the work done on Hackathon Day 4. The goal was to create reusable and user-friendly frontend components for a marketplace. These components are designed to work well on any device, look great, and be easy to use. Everything was built to be flexible and ready for real-world use.

Objectives:

- **Main Goal**: Build frontend components that can be used in many places.
- **Key Deliverables**:
    - Functional and responsive designs.
    - Components that are easy to reuse and update.
    - Real-time features for a better user experience.

Components Built:

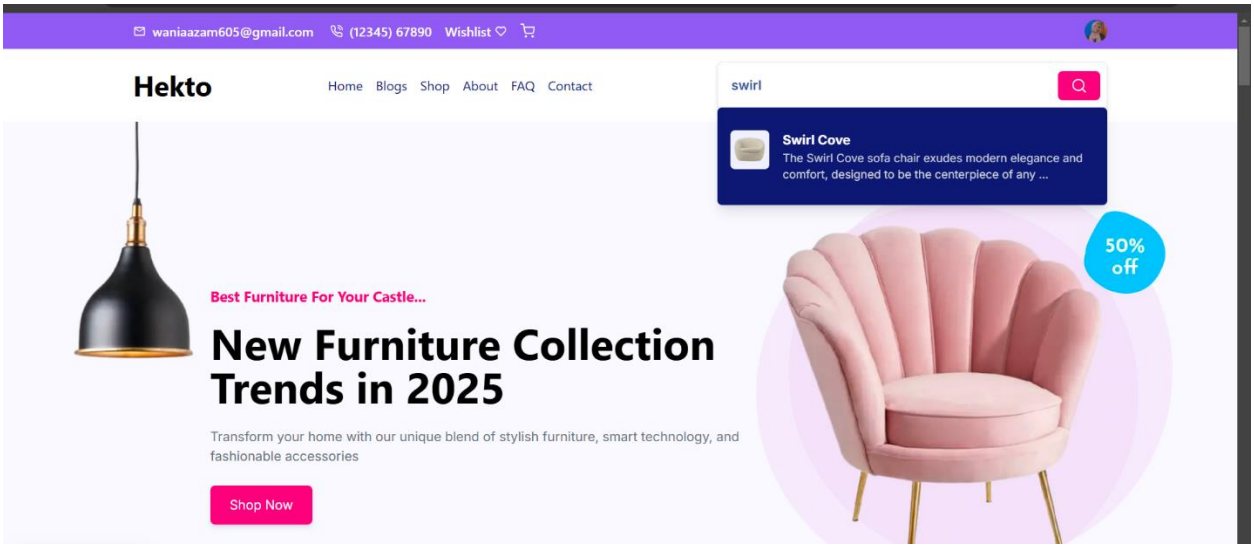
1. Header and Navigation Bar

Purpose: This is the top part of the site that helps users find what they need.

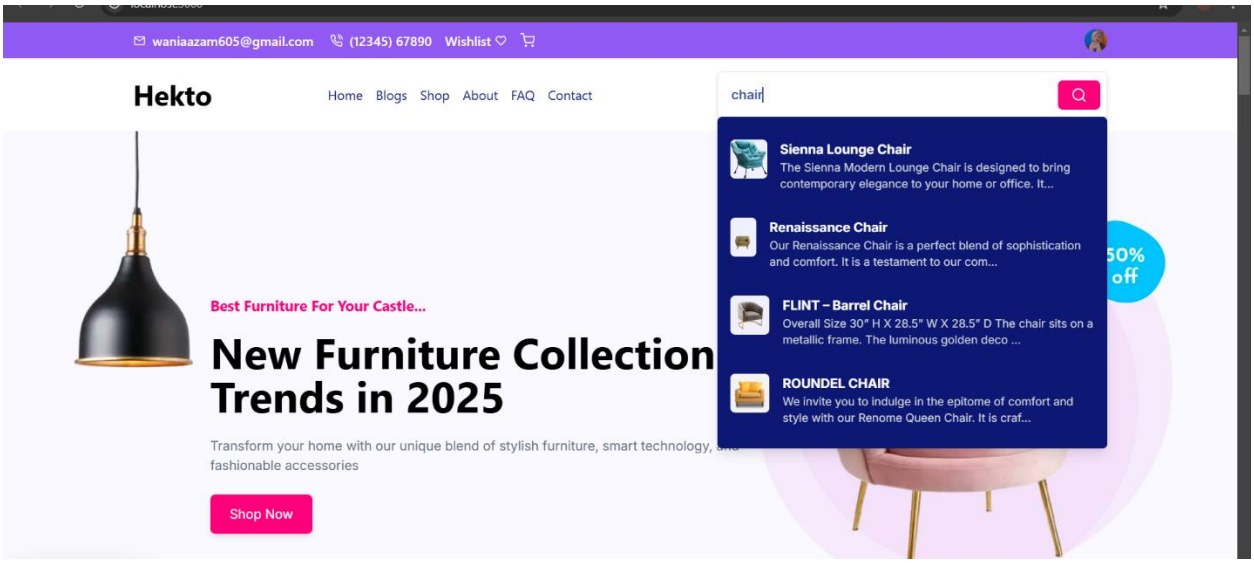
- **Features**:
    - **Store Logo and Name**: Easily changeable for any brand.

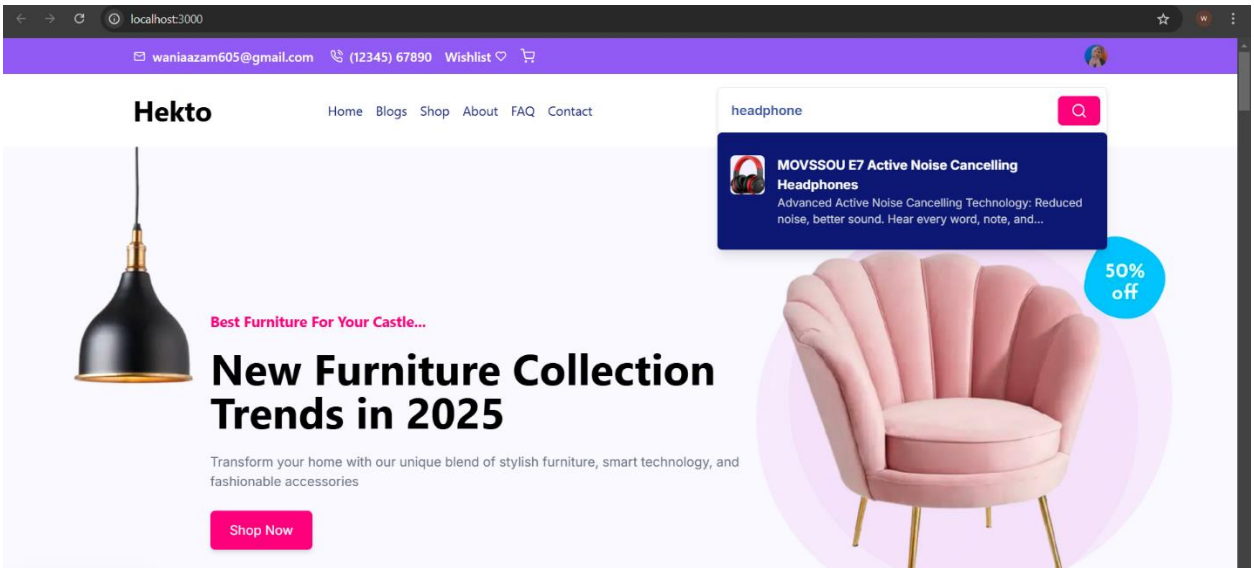
- # Search Bar

    - **Search Bar**: Quickly find products by typing in names or categories.



- user friendly Search Bar





- **Links to Important Pages**: Blog, Categories, FAQs, Contact Us.
- **Breadcrumbs**: Show where you are on the site.

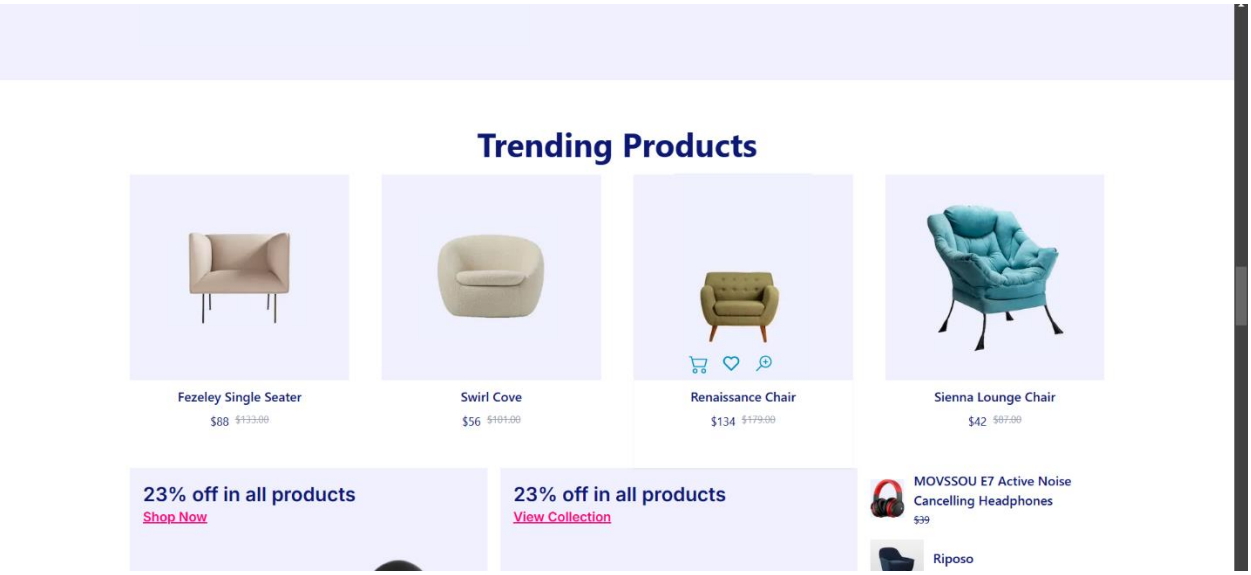
2. Product Card

Purpose: Display each product in a clean and simple way.

- **Details Shown**:
    - Product Name
    - Price
    - Image
    - Ratings and Stock Availability
- **Reusability**:
    - Works for trending, featured, or latest products.
    - Looks good on both mobile and desktop

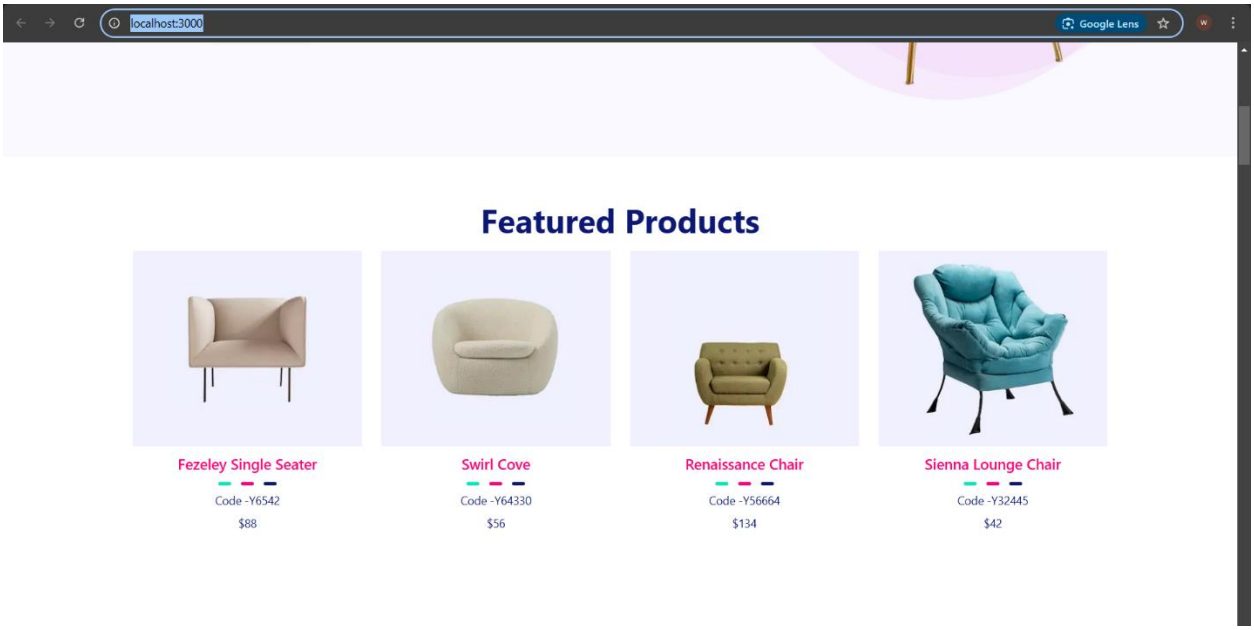
Trending Products:

- **Trending Product Of our Website.**



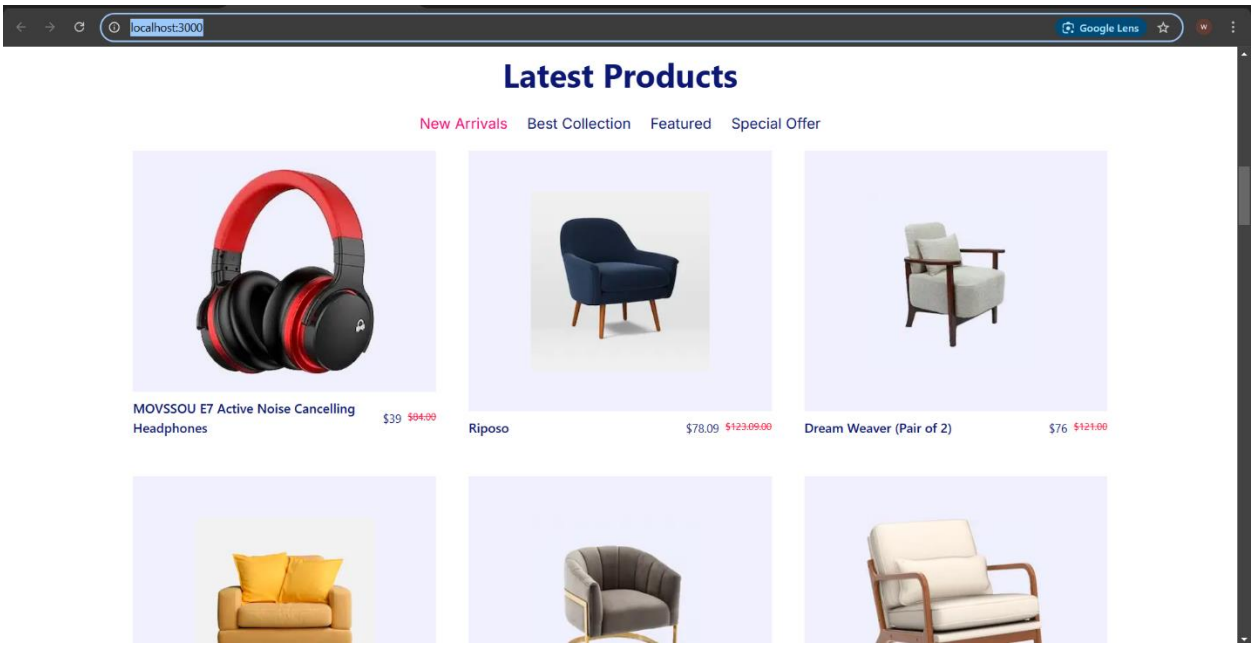
Featured Products:

- **Featured Product Of our Website.**



Latest Products:

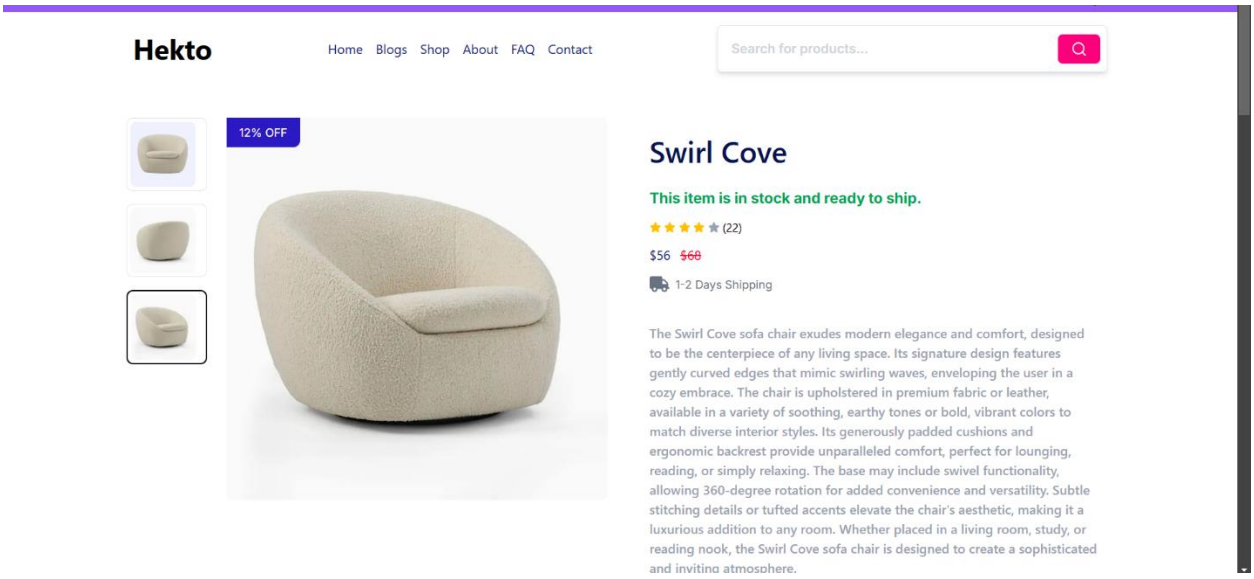
- **latest Product Of our Website.**



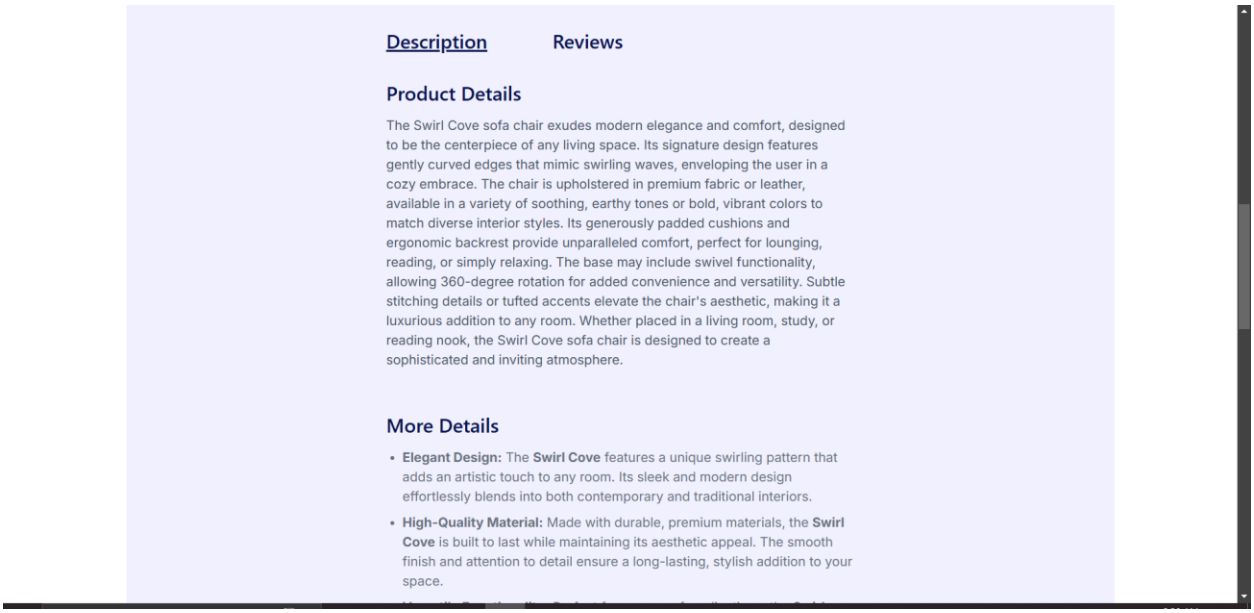
3. Product Details Page

Purpose: Give full details about a selected product.

- **Features**:
    - **Image Gallery**: View multiple product photos.

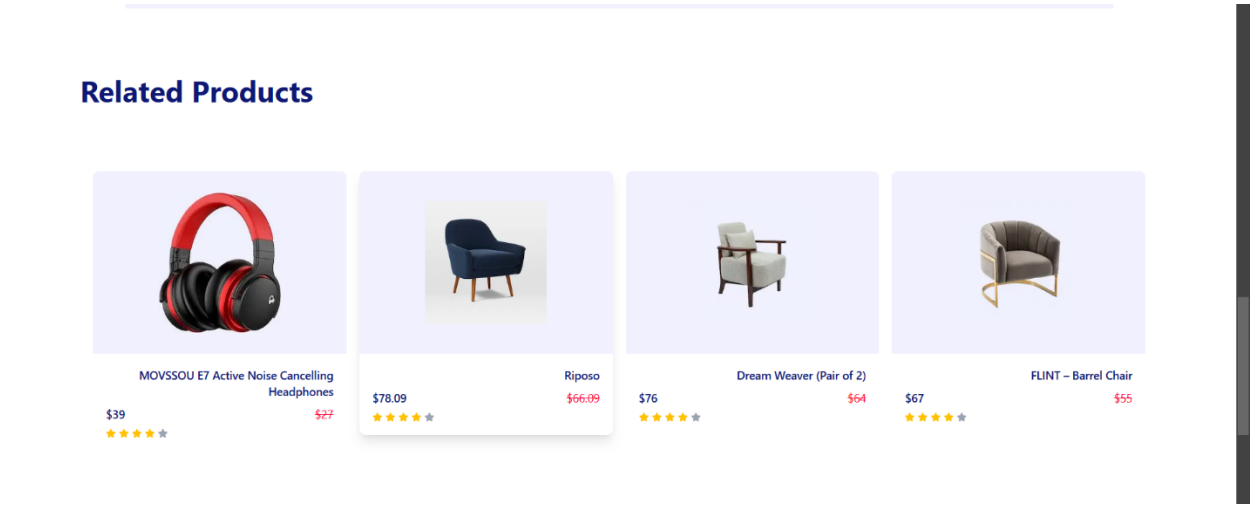


- **Description**: Clear information about the product.

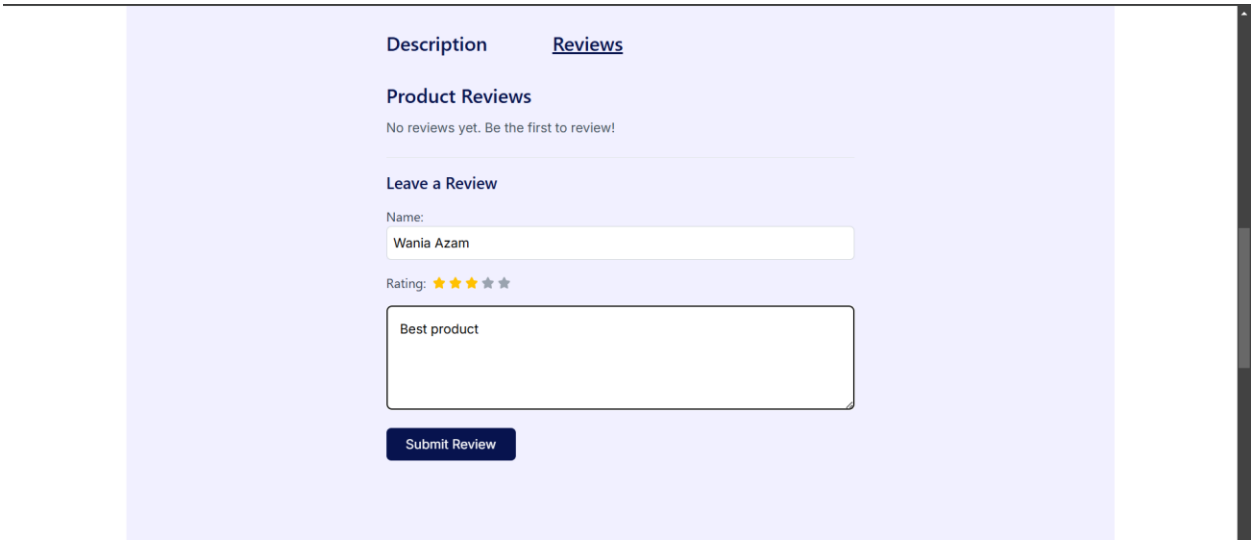


- **Pricing**: Includes discounts if available.

- **Related Products**: Suggests similar items.



- **Tags**:tags of the products.
- **Category**:category of the product.
- **Rating**:rating of the product.
- **Reviews**:reviews from the customers.



- **Two-Part Layout**:

1.Sidebar with product list: Lists categories and tags.

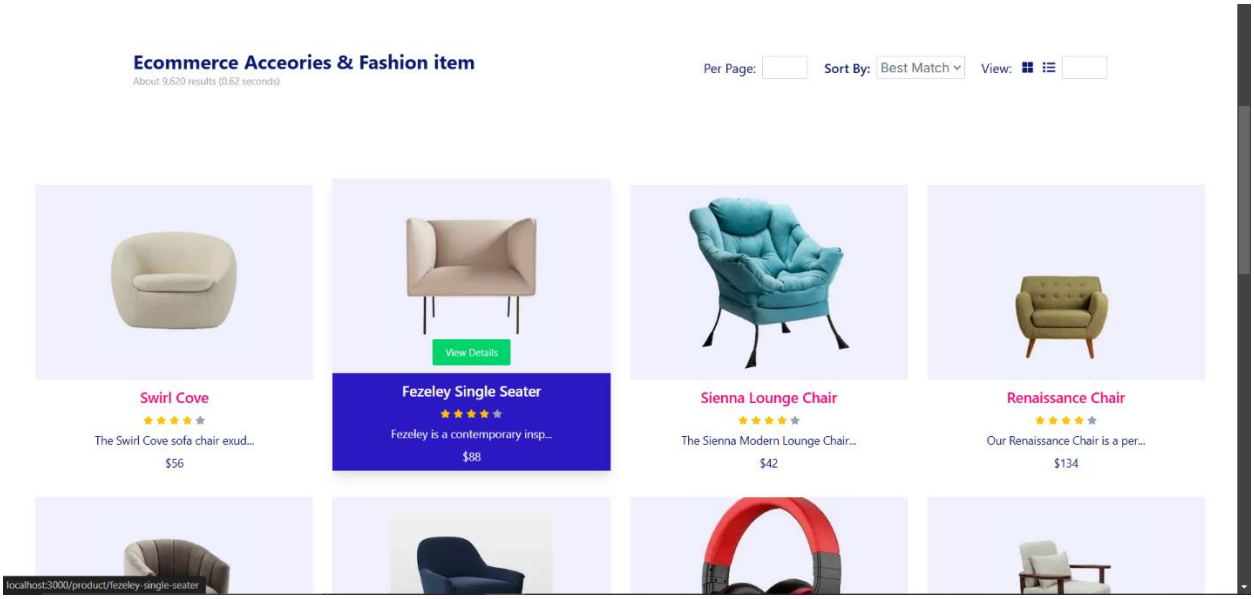
2. Grid Layout: All Products.

4. Shop Page

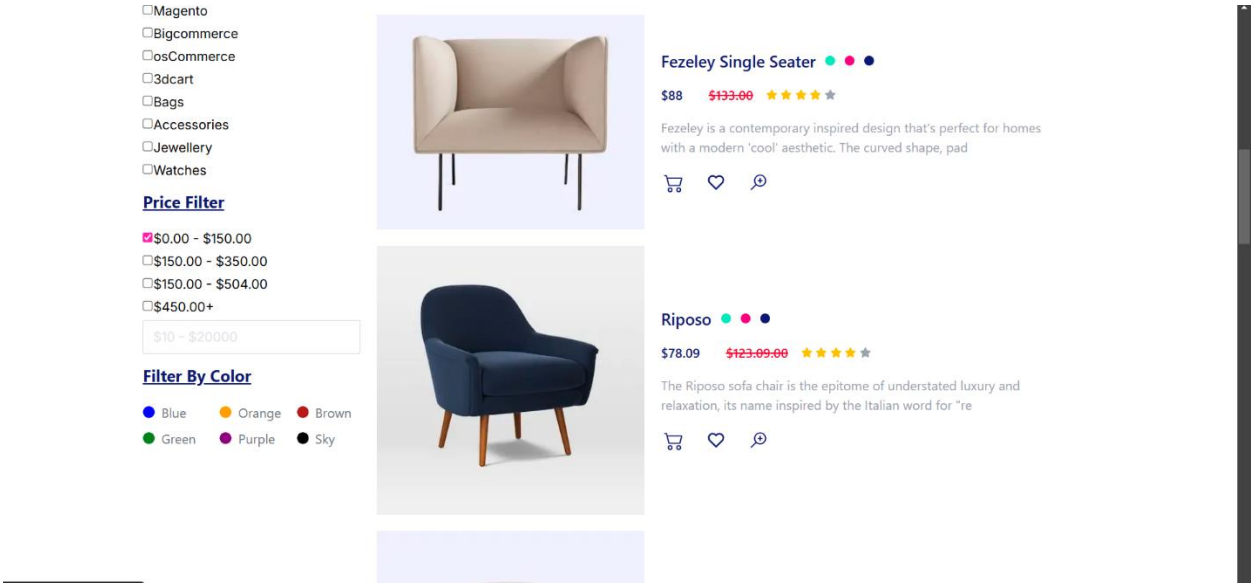
Purpose: Show all products in an easy-to-browse format.

- **Features:**
  - **Grid Layout:**

    Products organized in rows and columns.

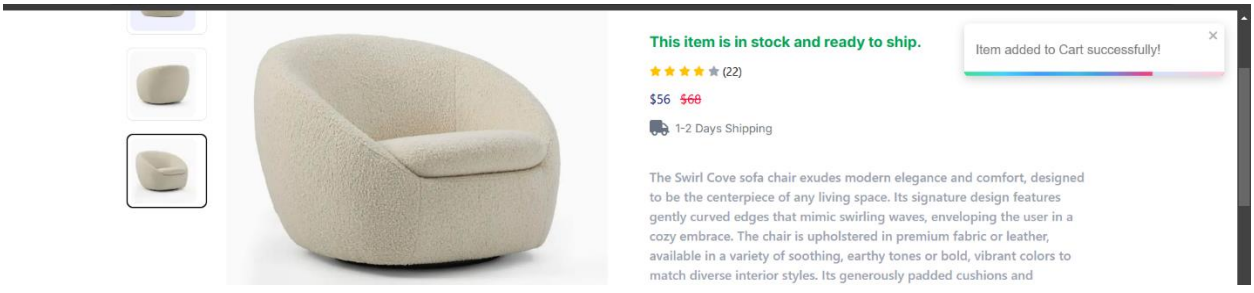


- ## Sidebar:
    - **Filters for categories, price, and ratings.**
    - **Tags for quick sorting.**



- ## Interactive Buttons:
    - **Add to Cart**
    - **Add to Wishlist**

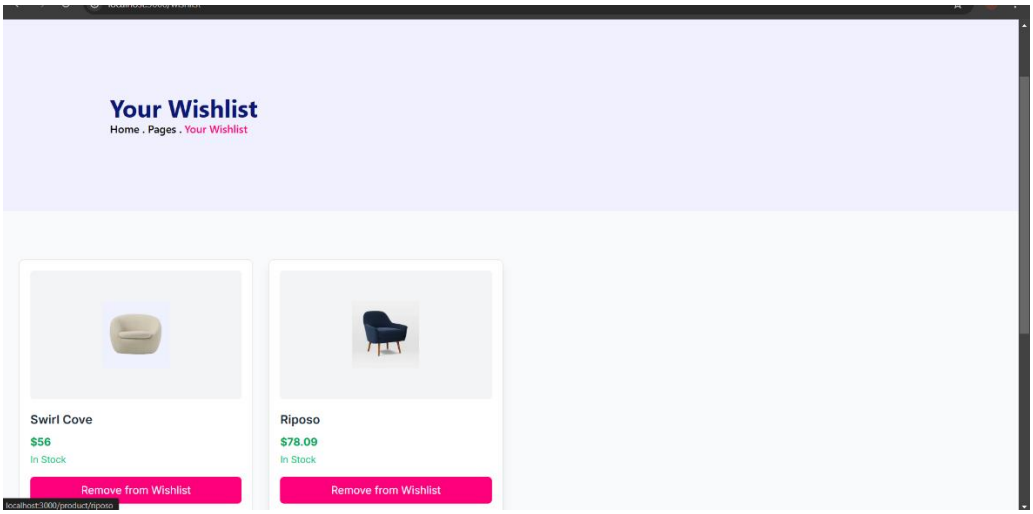
- **Notifications: Alerts users when actions are successful.**



5. Wishlist and Cart:

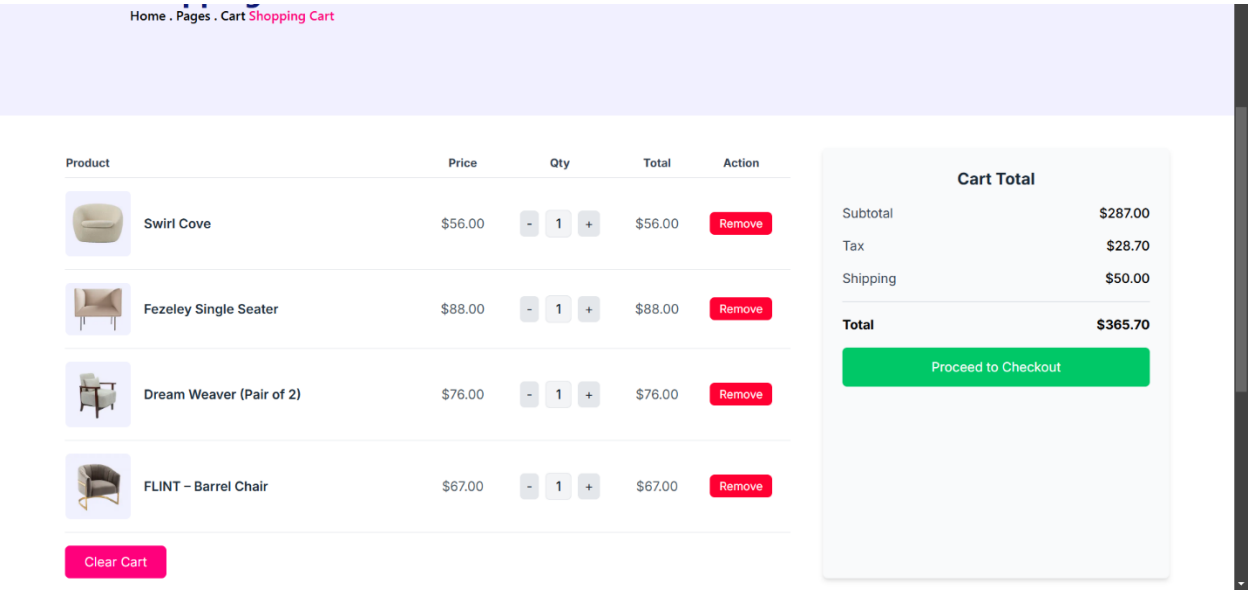
Wishlist Features:

- **Save products to view later.**
- **Remove items easily. Cart Features:**
- **Shows selected items, quantity, and total price.**
- **Options to remove items or clear the cart.**
- **Notifications for actions like adding or removing items.**

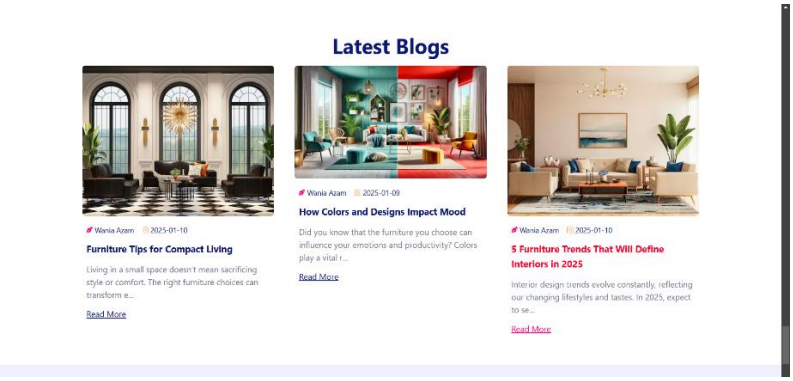


Cart Features:

- **Shows selected items, quantity, and total price.**
- **Options to remove items or clear the entire cart.**
- **Notifications for actions like adding or removing items from the cart.**

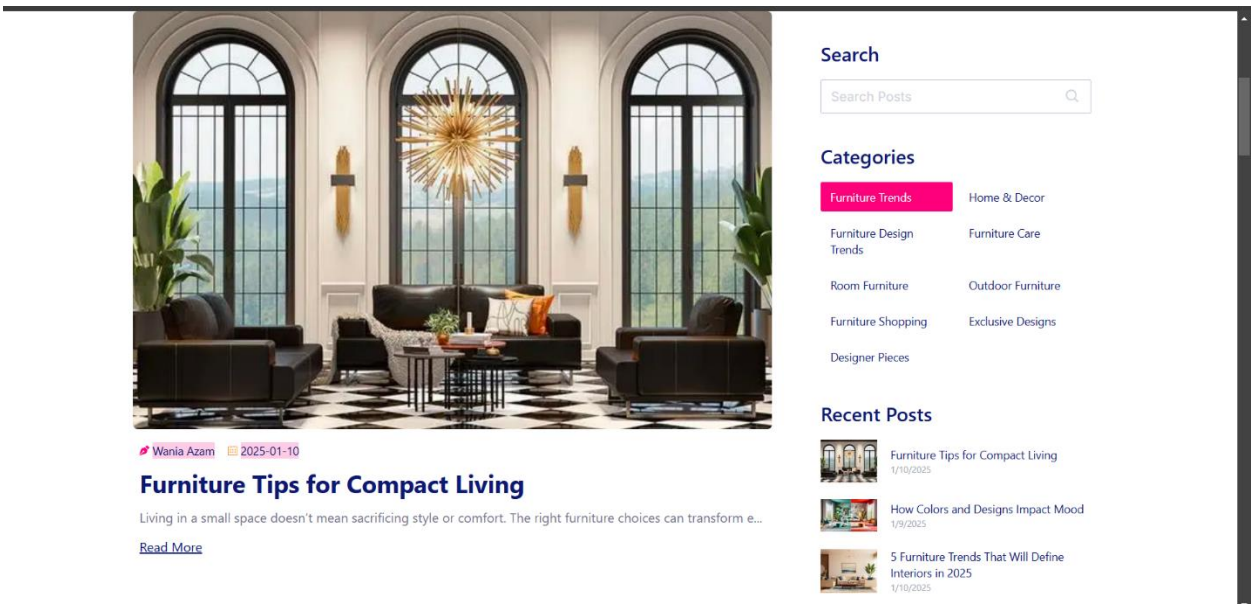


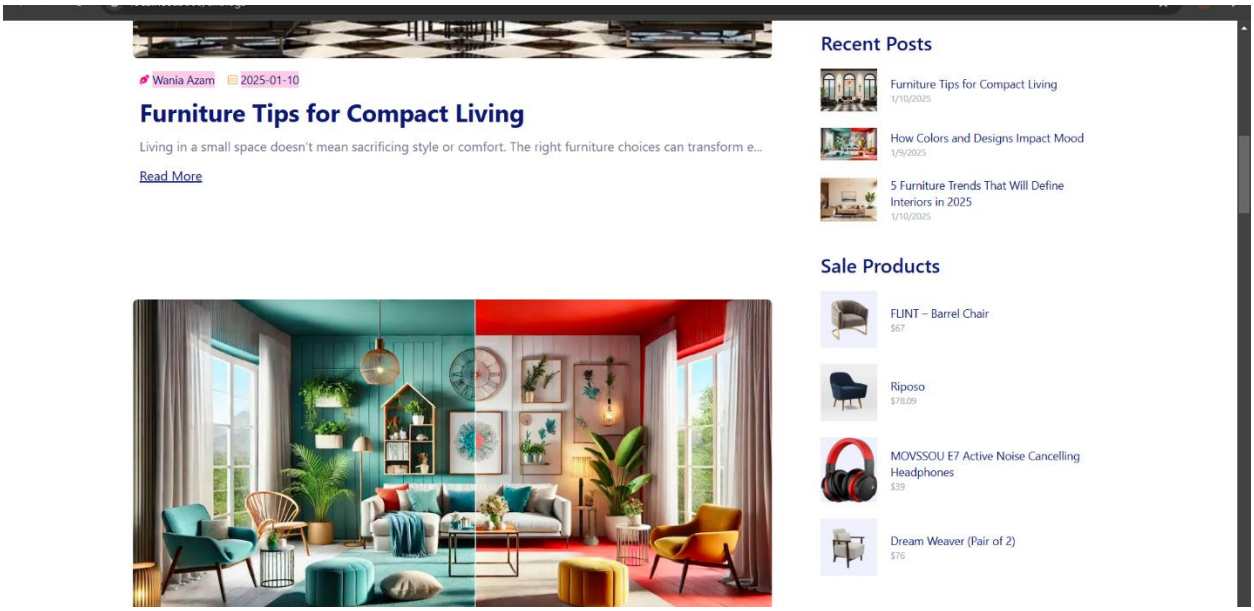
6. Latest Blog



7. ALL Blogs

- **Purpose:** Displays a list of all available blog posts in a clean, easy-to-read layout.
- **Features:**
    - **Blog Post List:** Shows titles of blog posts with short summaries. Each title is clickable and links to the **Blog Details Page**.
    - **Categories or Tags Filter:** Allows users to filter blog posts by specific topics, helping them find content they're interested in.
    - **Pagination:** Helps users navigate through multiple pages of blog posts if there are many entries.
    - **Responsive Design:** Ensures the layout looks great on mobile devices as well as desktops.

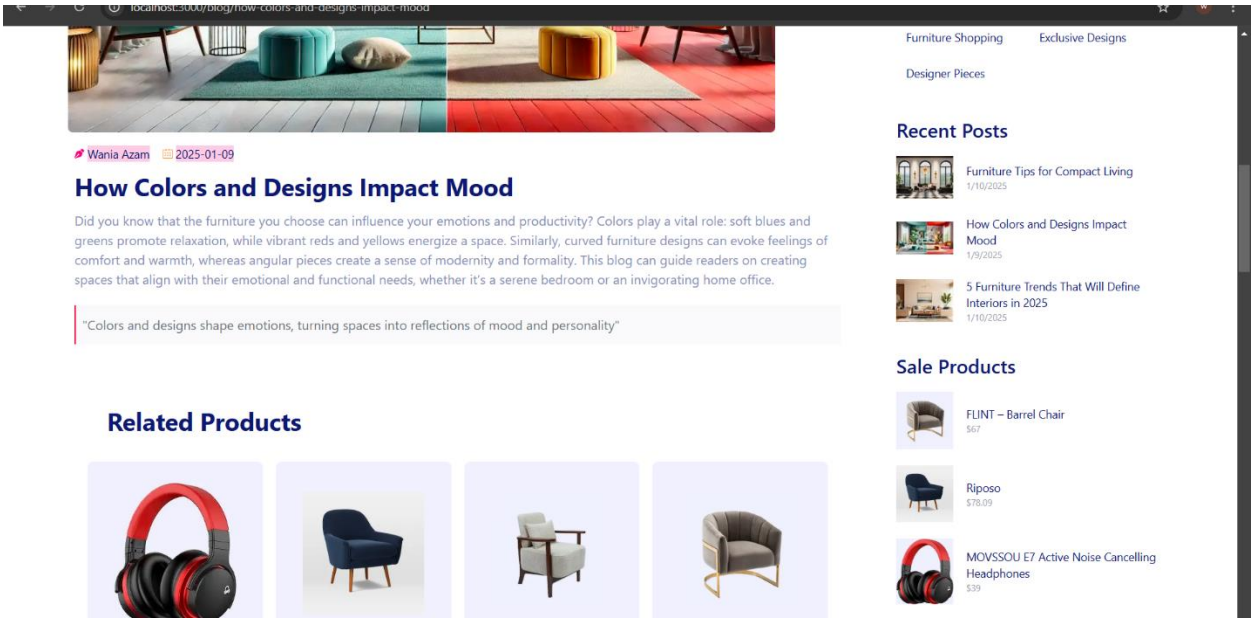


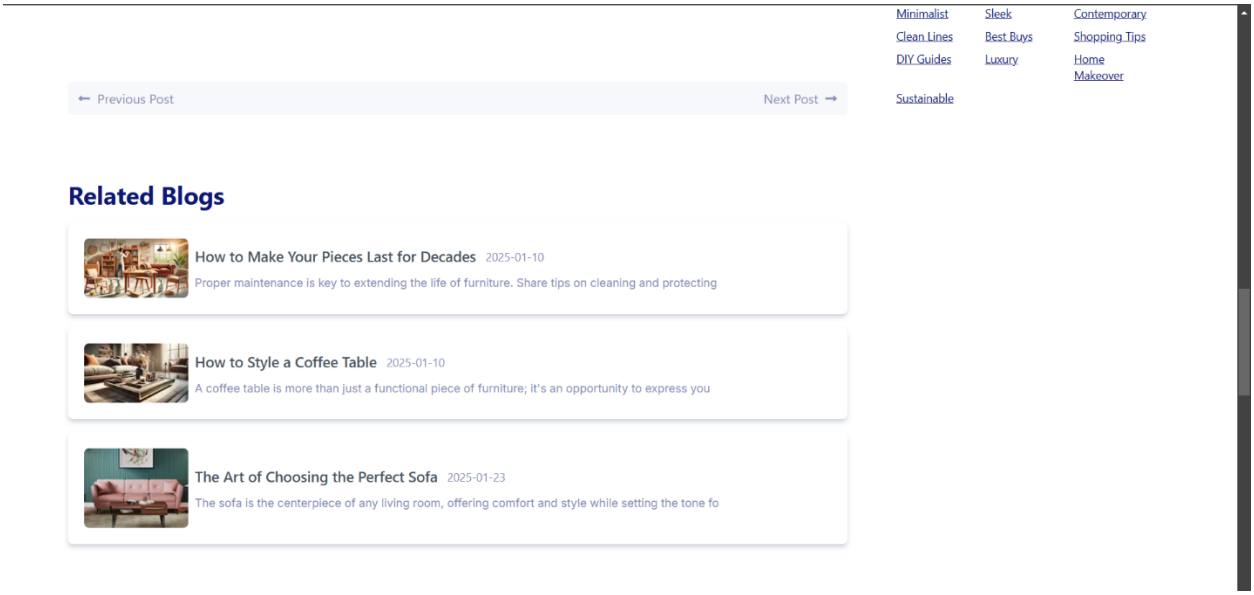


3. Blog Details Page

- **Purpose:** Displays full details for a specific blog post when clicked from the Blog Page.
- **Features:**
    - **Blog Post Title:** Displays the full title of the post.
    - **Date of Publication:** Shows when the post was published for better user context.
    - **Author Info:** Includes the name and profile of the author of the blog post.
    - **Main Content:** Contains the full text of the blog post with rich formatting (headings, images, etc.).
    - **Comments Section:** Users can leave comments on the blog post for interaction.

- **Related Posts:** Suggests other blog posts on similar topics to keep users engaged.
- **Navigation to Previous and Next Posts:** Allows users to move between blog posts easily, encouraging them to read more.





Extra Features:

Filtering and Sorting:

Purpose: Help users find products faster.

- **Filters:**
    - **Categories**
    - **Price Range**
    - **Ratings**
- **Sorting Options:**
    - **Best Sellers**
    - **Discounts**

Real-Time Notification

Purpose: Keep users informed.

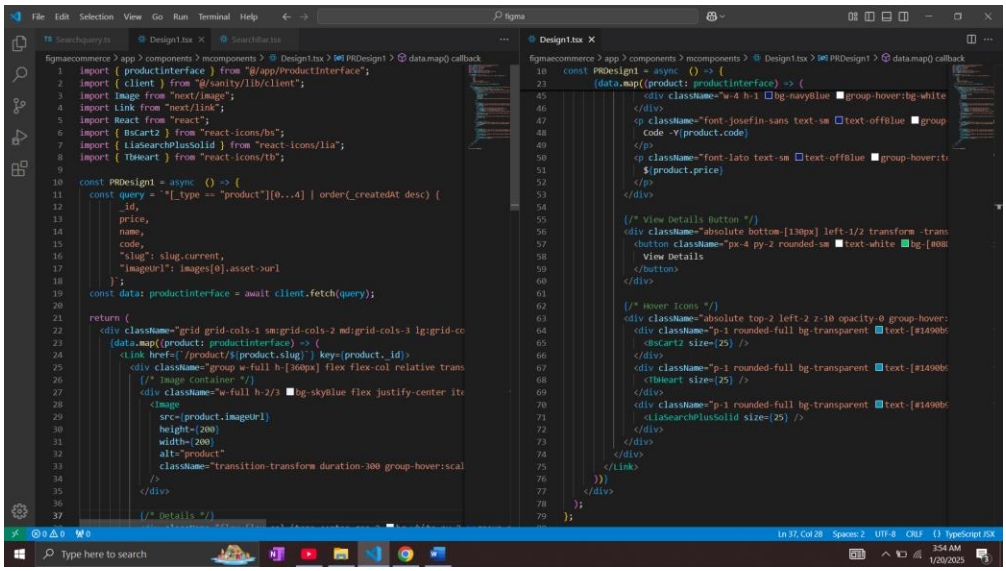
- **Alerts for adding or removing items from the cart or wishlist.**
- **Discount Section:**

Purpose: Highlight products on sale.

- **Updates automatically based on backend data.**

Code Snippets:

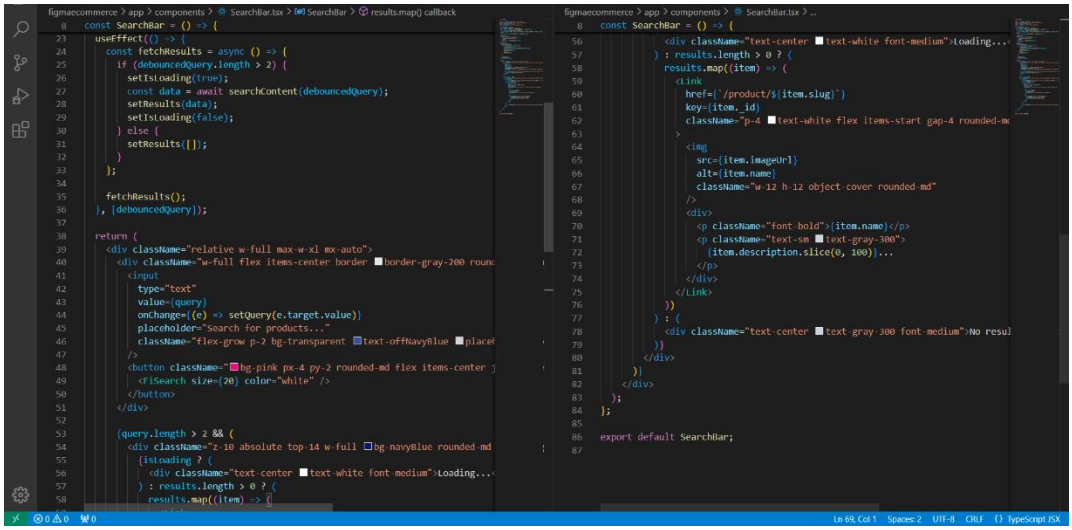
1: Code Snippets of ProductCard.



```
import { productinterface } from "@/app/Productinterface";
import { client } from "@/sanity/lib/client";
import Image from "next/image";
import Link from "next/link";
import React from "react";
import { BsCart2 } from "react-icons/bs";
import { LiaSearchPlusSolid } from "react-icons/lia";
import { TbHeart } from "react-icons/tb";

const PRDesign1 = async () => {
  const query = `*[_type == "product"][0...4] | order(_createdAt desc) {
        _id,
        price,
        name,
        code,
        "slug": slug.current,
        "imageUrl": images[0].asset->url
      }`;
  const data: productinterface = await client.fetch(query);

  return (
    <div className="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:grid-co
      {data.map((product: productinterface) => (
        <Link href={`/product/${product.slug}`} key={product._id}>
          <div className="group w-full h-[360px] flex flex-col relative trans
            {/* Image Container */}
            <div className="w-full h-2/3 bg-skyBlue flex justify-center ite
              <Image
                src={product.imageUrl}
                height={200}
                width={200}
                alt="product"
                className="transition-transform duration-300 group-hover:scal
              />
            </div>

            {/* Details */}
```

```
const PRDesign1 = async () => {
  {data.map((product: productinterface) => (
              <div className="w-4 h-1 bg-navyBlue group-hover:bg-white
            </div>
            <p className="font-josefin-sans text-sm text-offBlue group
              Code -Y{product.code}
            </p>
            <p className="font-lato text-sm text-offBlue group-hover:t
              ${product.price}
            </p>
          </div>

          {/* View Details Button */}
          <div className="absolute bottom-[130px] left-1/2 transform -trans
            <button className="px-4 py-2 rounded-sm text-white bg-[#08
              View Details
            </button>
          </div>

          {/* Hover Icons */}
          <div className="absolute top-2 left-2 z-10 opacity-0 group-hover:
            <div className="p-1 rounded-full bg-transparent text-[#1490b
              <BsCart2 size={25} />
            </div>
            <div className="p-1 rounded-full bg-transparent text-[#1490b
              <TbHeart size={25} />
            </div>
            <div className="p-1 rounded-full bg-transparent text-[#1490b
              <LiaSearchPlusSolid size={25} />
            </div>
          </div>
        </div>
      </Link>
    ))}
  </div>
);
};
```

2: Code Snippets of SearchBar.



```
const SearchBar = () => {
  useEffect(() => {
    const fetchResults = async () => {
      if (debouncedQuery.length > 2) {
        setIsLoading(true);
        const data = await searchContent(debouncedQuery);
        setResults(data);
        setIsLoading(false);
      } else {
        setResults([]);
      }
    };

    fetchResults();
  }, [debouncedQuery]);

  return (
    <div className="relative w-full max-w-xl mx-auto">
      <div className="w-full flex items-center border border-gray-200 round
        <input
          type="text"
          value={query}
          onChange={(e) => setQuery(e.target.value)}
          placeholder="Search for products..."
          className="flex-grow p-2 bg-transparent text-offNavyBlue placeh
        />
        <button className="bg-pink px-4 py-2 rounded-md flex items-center j
          <FiSearch size={20} color="white" />
        </button>
      </div>

      {query.length > 2 && (
        <div className="z-10 absolute top-14 w-full bg-navyBlue rounded-md
          {isLoading ? (
            <div className="text-center text-white font-medium">Loading...
          ) : results.length > 0 ? (
            results.map((item) => (
```

```
const SearchBar = () => {
            <div className="text-center text-white font-medium">Loading...
          ) : results.length > 0 ? (
            results.map((item) => (
              <Link
                href={`/product/${item.slug}`}
                key={item._id}
                className="p-4 text-white flex items-start gap-4 rounded-m
              >
                <img
                  src={item.imageUrl}
                  alt={item.name}
                  className="w-12 h-12 object-cover rounded-md"
                />
                <div>
                  <p className="font-bold">{item.name}</p>
                  <p className="text-sm text-gray-300">
                    {item.description.slice(0, 100)}...
                  </p>
                </div>
              </Link>
            ))
          ) : (
            <div className="text-center text-gray-300 font-medium">No resul
          )}
        </div>
      )}
    </div>
  );
};

export default SearchBar;
```

Technical report:

1. Steps to Build & Integrate Components

- **Planning & Design:**
    - **Researched marketplace components.**
    - **Used Figma for wireframes and UI design.**
- **Component Development:**
    - **Header & Navigation Bar: Used Tailwind CSS for a responsive layout with logo, search bar, and quick links.**
    - **Product Cards: Styled with Tailwind CSS for product details (name, price, image). Responsive design using flex and grid.**
    - **Product Details Page: Built two-part layout (main content + sidebar) with Tailwind CSS for responsiveness.**
    - **Shop Page: Used grid layout for product display and added filtering and sorting options in a sidebar.**
    - **Wishlist & Cart: Styled with Tailwind CSS, interactive buttons to add/remove items with real-time updates.**
- **Integration:**
    - **Combined components (e.g., product card on shop and detail pages).**
    - **Integrated real-time features (e.g., cart and wishlist notifications).**

---

2. Challenges & Solutions

- **Reusability: Ensured components like product cards could be used across multiple pages by keeping them modular with Tailwind CSS classes.**
- **Responsiveness: Used Tailwind CSS's media queries (sm:, md:, lg:) to ensure all components look good on all screen sizes.**

3. Best Practices

- **Modular Code: Used Tailwind CSS for reusable styling classes, making the code maintainable and clean.**

- **Well-Commented Code: Commented key sections for easier understanding and future updates.**
- **User-Centered Design: Focused on intuitive navigation and interactive features like filters and notifications.**

Conclusion:

On Hackathon Day 4, I focused on creating a robust and polished set of reusable frontend components designed to elevate the user experience in a dynamic marketplace. Each component was built with responsiveness, usability, and scalability in mind, ensuring they function seamlessly across devices. The core features I developed included a sleek Header and Navigation Bar, which serve as the foundation for easy navigation, and a clean, intuitive Product Card system that displays product details in a user-friendly manner.

Additionally, I created a detailed Product Details Page where users can explore individual product features, an organized Shop Page to showcase all products in a visually appealing grid, and a Wishlist and Cart functionality to enhance the shopping experience. Each component was carefully crafted to ensure smooth interactions, like real-time notifications and quick sorting/filtering options, making browsing more efficient and enjoyable.

Furthermore, I incorporated a Blog Page to engage users, featuring a detailed Blog Details Page with seamless navigation through Previous/Next posts, and a Blog Sidebar to help users easily explore related content. To improve the overall functionality, I also included features like automatic updates in the Discount Section, along with interactive sorting and filtering tools, so users can quickly find products that meet their needs.

The entire project was built with clean, well-commented code to ensure maintainability and easy future updates. The goal was to design flexible components that can be reused in various contexts, and I've ensured that all features work together to provide an exceptional, real-world-ready product.

By Wania Azam