By Muhammad Abdullah

Day 4: Building Dynamic Frontend Components for Your Marketplace

---

Objective:

Day 4 focused on designing and developing dynamic frontend components in my marketplace. The main goal was to ensure that these components fetch and display data from Sanity CMS or APIs, making my site dynamic and user-friendly. I also have learned how to structure components in a way that allows for easy reusability, scalability, and responsiveness.

Key Learning Outcomes:

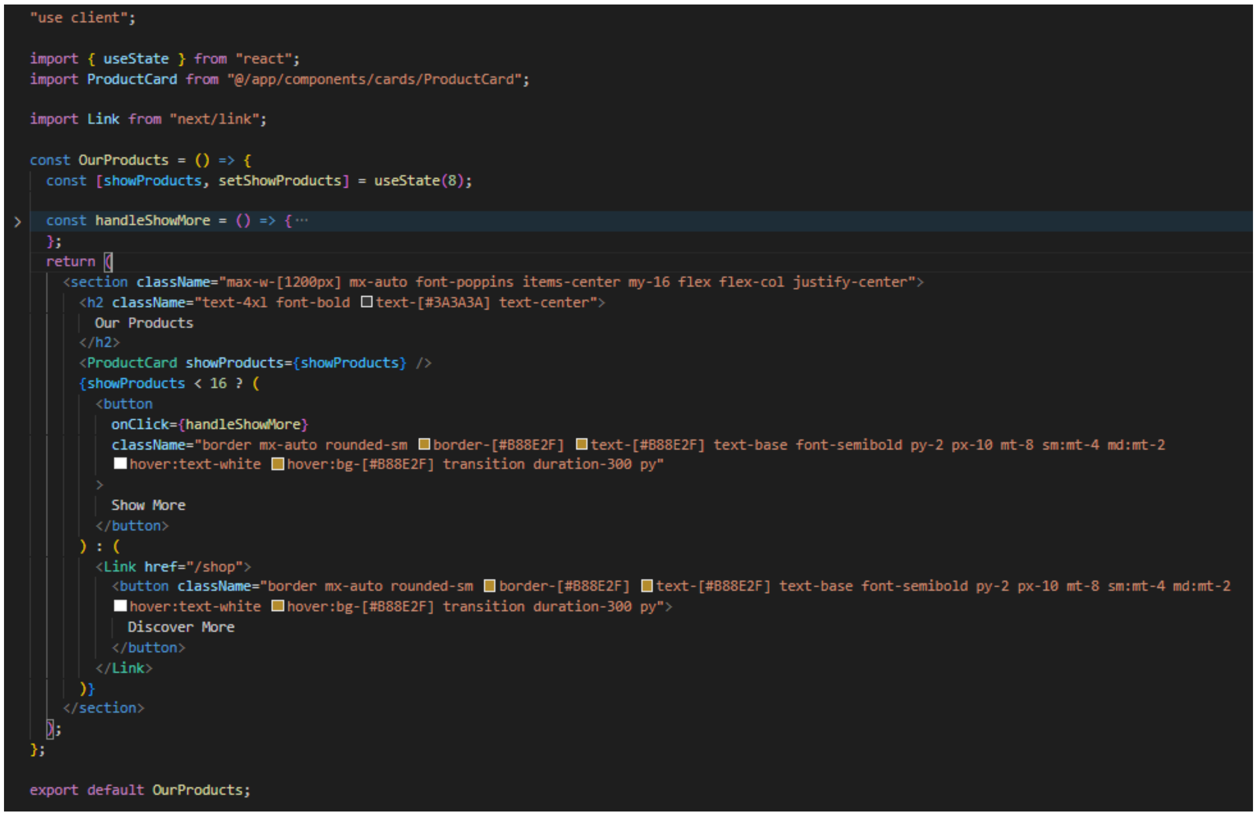
Throughout the day, I have achieved several key outcomes:

- I built **dynamic frontend components** that displayed data fetched from APIs or Sanity CMS.
- I learned how to design **reusable and modular components** to keep my code efficient and easy to maintain.
- I understood the importance of **state management**, ensuring smooth interaction between components and maintaining dynamic data.
- I applied **responsive design principles** and UX/UI best practices, ensuring that my application would look great on any device, from mobile to desktop.
- Lastly, I got hands-on experience replicating professional workflows, preparing me for real-world client projects.

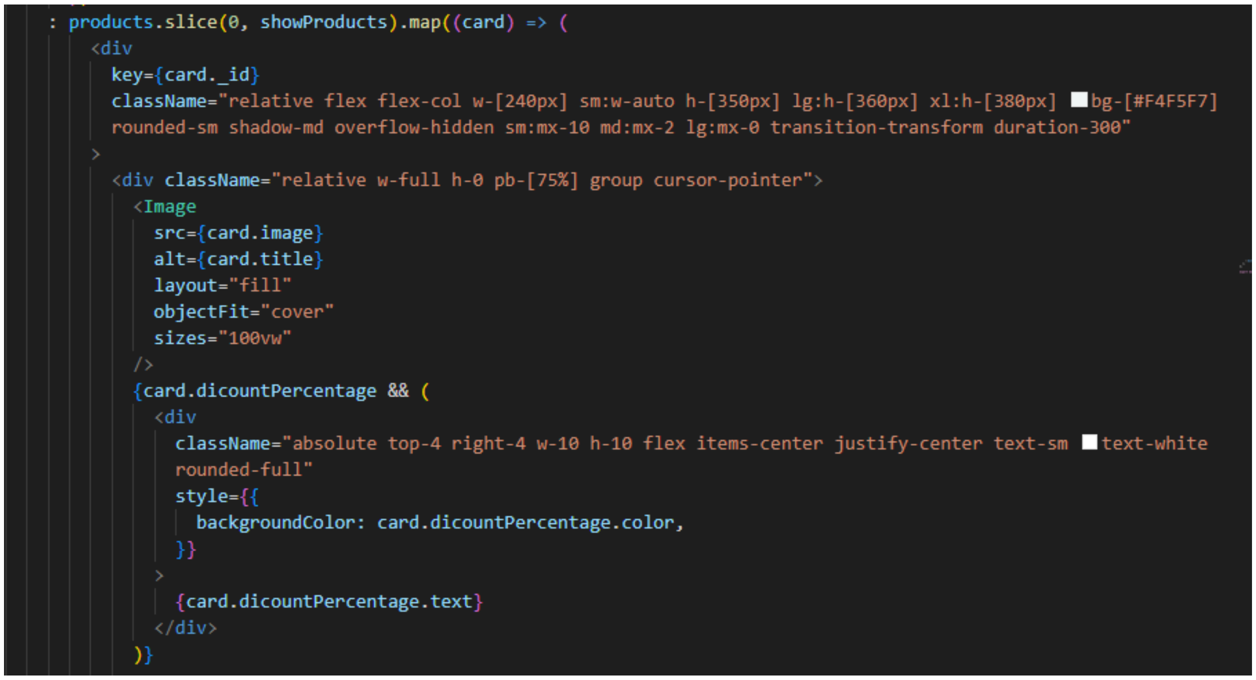
Key Components Built:

On Day 4, I constructed several key components:

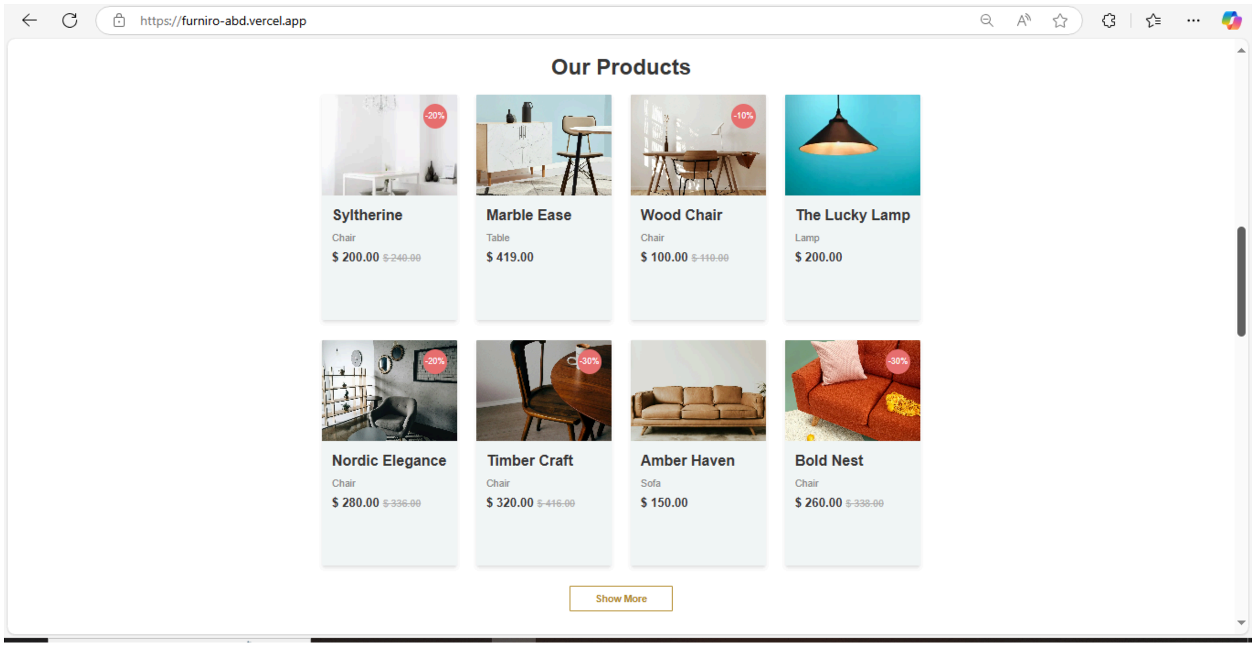
1. **Product Listing Component:** I created a grid layout to display product data, including fields such as product name, price, image, and stock status. This component helps organize products and make them easy to browse.



```
"use client";

import { useState } from "react";
import ProductCard from "@/app/components/cards/ProductCard";

import Link from "next/link";

const OurProducts = () => {
  const [showProducts, setShowProducts] = useState(8);

  const handleShowMore = () => {⋯
  };
  return (
    <section className="max-w-[1200px] mx-auto font-poppins items-center my-16 flex flex-col justify-center">
      <h2 className="text-4xl font-bold text-[#3A3A3A] text-center">
        Our Products
      </h2>
      <ProductCard showProducts={showProducts} />
      {showProducts < 16 ? (
        <button
          onClick={handleShowMore}
          className="border mx-auto rounded-sm border-[#B88E2F] text-[#B88E2F] text-base font-semibold py-2 px-10 mt-8 sm:mt-4 md:mt-2
          hover:text-white hover:bg-[#B88E2F] transition duration-300 py"
        >
          Show More
        </button>
      ) : (
        <Link href="/shop">
          <button className="border mx-auto rounded-sm border-[#B88E2F] text-[#B88E2F] text-base font-semibold py-2 px-10 mt-8 sm:mt-4 md:mt-2
          hover:text-white hover:bg-[#B88E2F] transition duration-300 py">
            Discover More
          </button>
        </Link>
      )}
    </section>
  );
};

export default OurProducts;
```

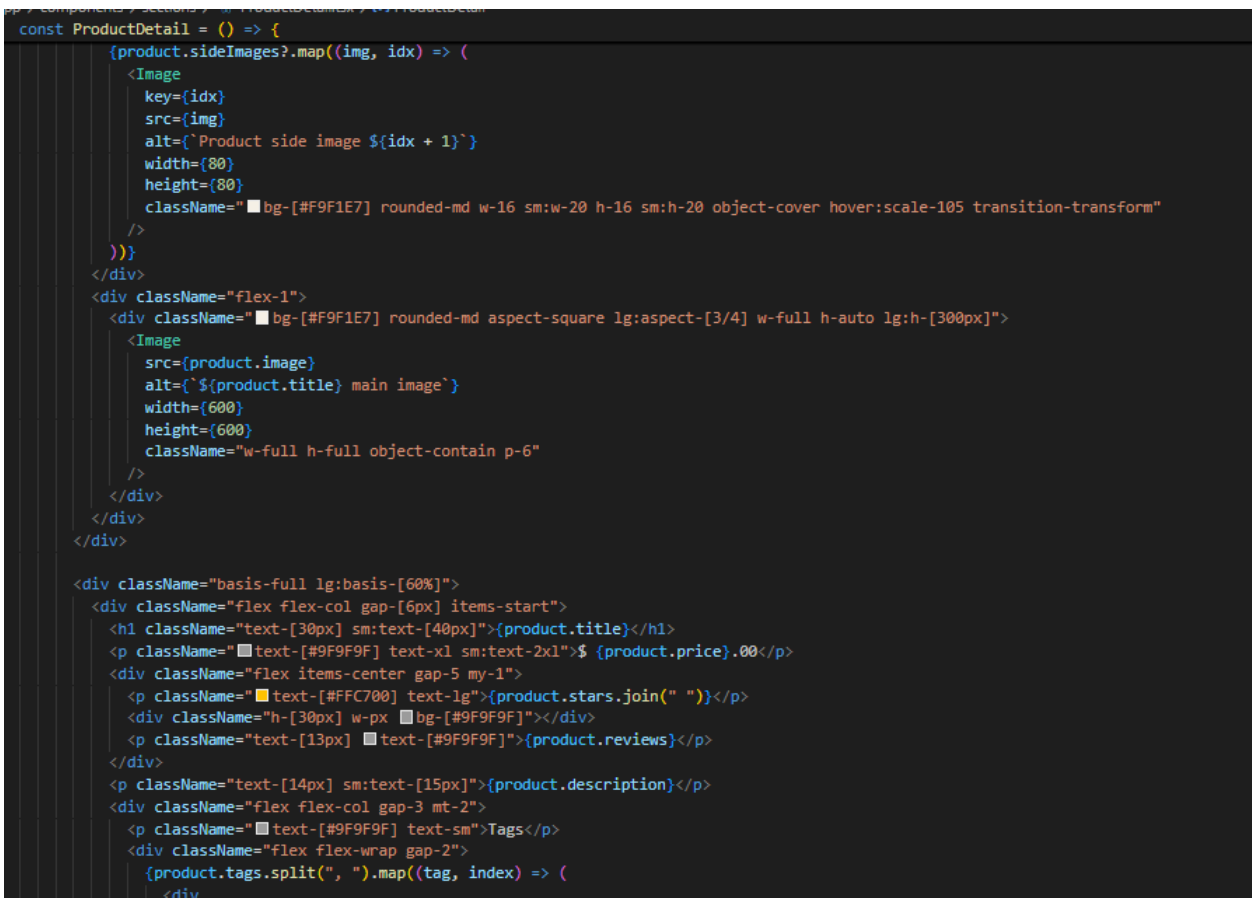


```
    : products.slice(0, showProducts).map((card) => (
        <div
          key={card._id}
          className="relative flex flex-col w-[240px] sm:w-auto h-[350px] lg:h-[360px] xl:h-[380px] bg-[#F4F5F7]
          rounded-sm shadow-md overflow-hidden sm:mx-10 md:mx-2 lg:mx-0 transition-transform duration-300"
        >
          <div className="relative w-full h-0 pb-[75%] group cursor-pointer">
            <Image
              src={card.image}
              alt={card.title}
              layout="fill"
              objectFit="cover"
              sizes="100vw"
            />
            {card.dicountPercentage && (
              <div
                className="absolute top-4 right-4 w-10 h-10 flex items-center justify-center text-sm text-white
                rounded-full"
                style={{
                  backgroundColor: card.dicountPercentage.color,
                }}
              >
                {card.dicountPercentage.text}
              </div>
            )}
```

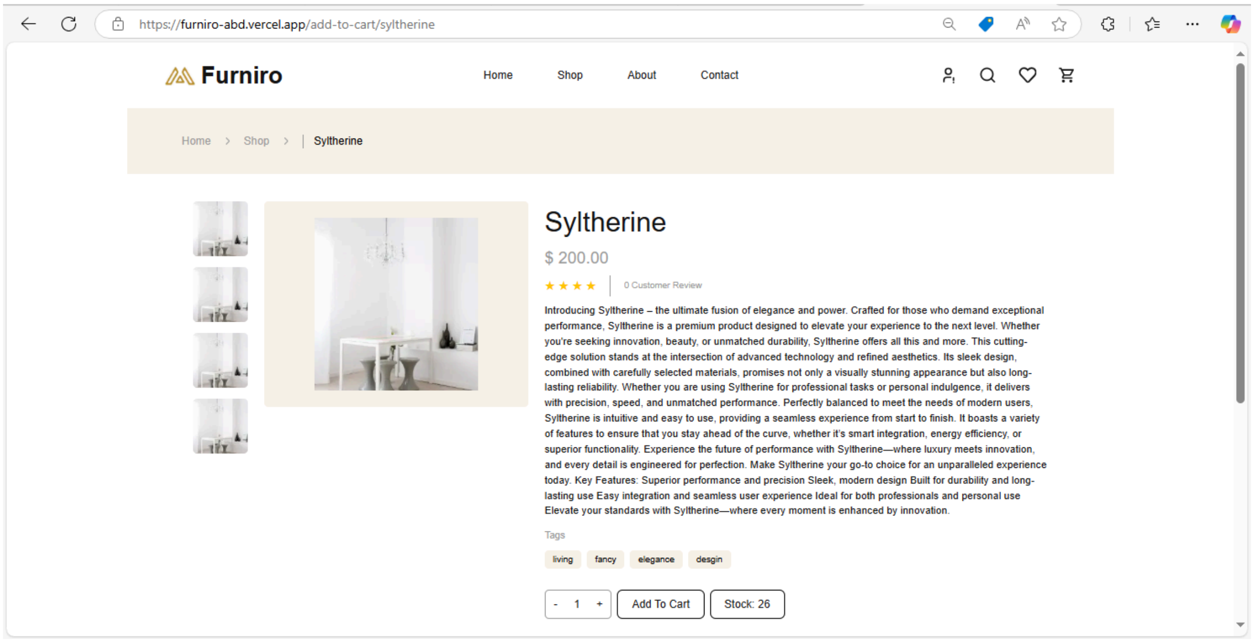


2. **Product Detail Component:** Using dynamic routing in Next.js, I built individual product detail pages. These pages include more detailed fields like product

description, price, tags, making it easy for users to learn more about each product.



```
const ProductDetail = () => {
            {product.sideImages?.map((img, idx) => (
              <Image
                key={idx}
                src={img}
                alt={`Product side image ${idx + 1}`}
                width={80}
                height={80}
                className="bg-[#F9F1E7] rounded-md w-16 sm:w-20 h-16 sm:h-20 object-cover hover:scale-105 transition-transform"
              />
            ))}
          </div>
          <div className="flex-1">
            <div className="bg-[#F9F1E7] rounded-md aspect-square lg:aspect-[3/4] w-full h-auto lg:h-[300px]">
              <Image
                src={product.image}
                alt={`${product.title} main image`}
                width={600}
                height={600}
                className="w-full h-full object-contain p-6"
              />
            </div>
          </div>
        </div>

        <div className="basis-full lg:basis-[60%]">
          <div className="flex flex-col gap-[6px] items-start">
            <h1 className="text-[30px] sm:text-[40px]">{product.title}</h1>
            <p className="text-[#9F9F9F] text-xl sm:text-2xl">$ {product.price}.00</p>
            <div className="flex items-center gap-5 my-1">
              <p className="text-[#FFC700] text-lg">{product.stars.join(" ")}</p>
              <div className="h-[30px] w-px bg-[#9F9F9F]"></div>
              <p className="text-[13px] text-[#9F9F9F]">{product.reviews}</p>
            </div>
            <p className="text-[14px] sm:text-[15px]">{product.description}</p>
            <div className="flex flex-col gap-3 mt-2">
              <p className="text-[#9F9F9F] text-sm">Tags</p>
              <div className="flex flex-wrap gap-2">
                {product.tags.split(", ").map((tag, index) => (
```



https://furniro-abd.vercel.app/add-to-cart/syltherine

Furniro

Home Shop About Contact

Home > Shop > | Syltherine

Syltherine

$ 200.00

★ ★ ★ ★ | 0 Customer Review

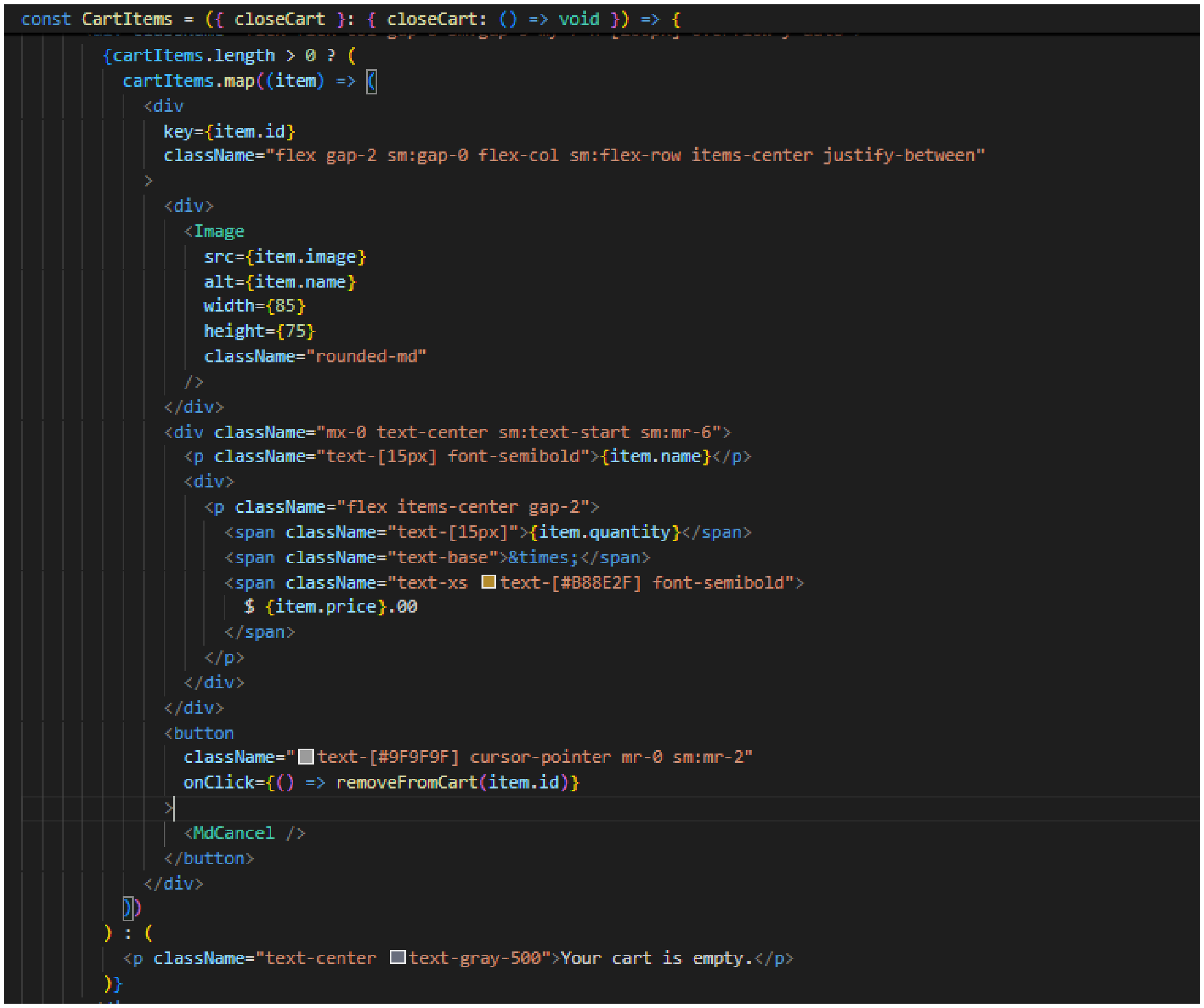
Introducing Syltherine – the ultimate fusion of elegance and power. Crafted for those who demand exceptional performance, Syltherine is a premium product designed to elevate your experience to the next level. Whether you're seeking innovation, beauty, or unmatched durability, Syltherine offers all this and more. This cutting-edge solution stands at the intersection of advanced technology and refined aesthetics. Its sleek design, combined with carefully selected materials, promises not only a visually stunning appearance but also long-lasting reliability. Whether you are using Syltherine for professional tasks or personal indulgence, it delivers with precision, speed, and unmatched performance. Perfectly balanced to meet the needs of modern users, Syltherine is intuitive and easy to use, providing a seamless experience from start to finish. It boasts a variety of features to ensure that you stay ahead of the curve, whether it's smart integration, energy efficiency, or superior functionality. Experience the future of performance with Syltherine—where luxury meets innovation, and every detail is engineered for perfection. Make Syltherine your go-to choice for an unparalleled experience today. Key Features: Superior performance and precision Sleek, modern design Built for durability and long-lasting use Easy integration and seamless user experience Ideal for both professionals and personal use Elevate your standards with Syltherine—where every moment is enhanced by innovation.

Tags

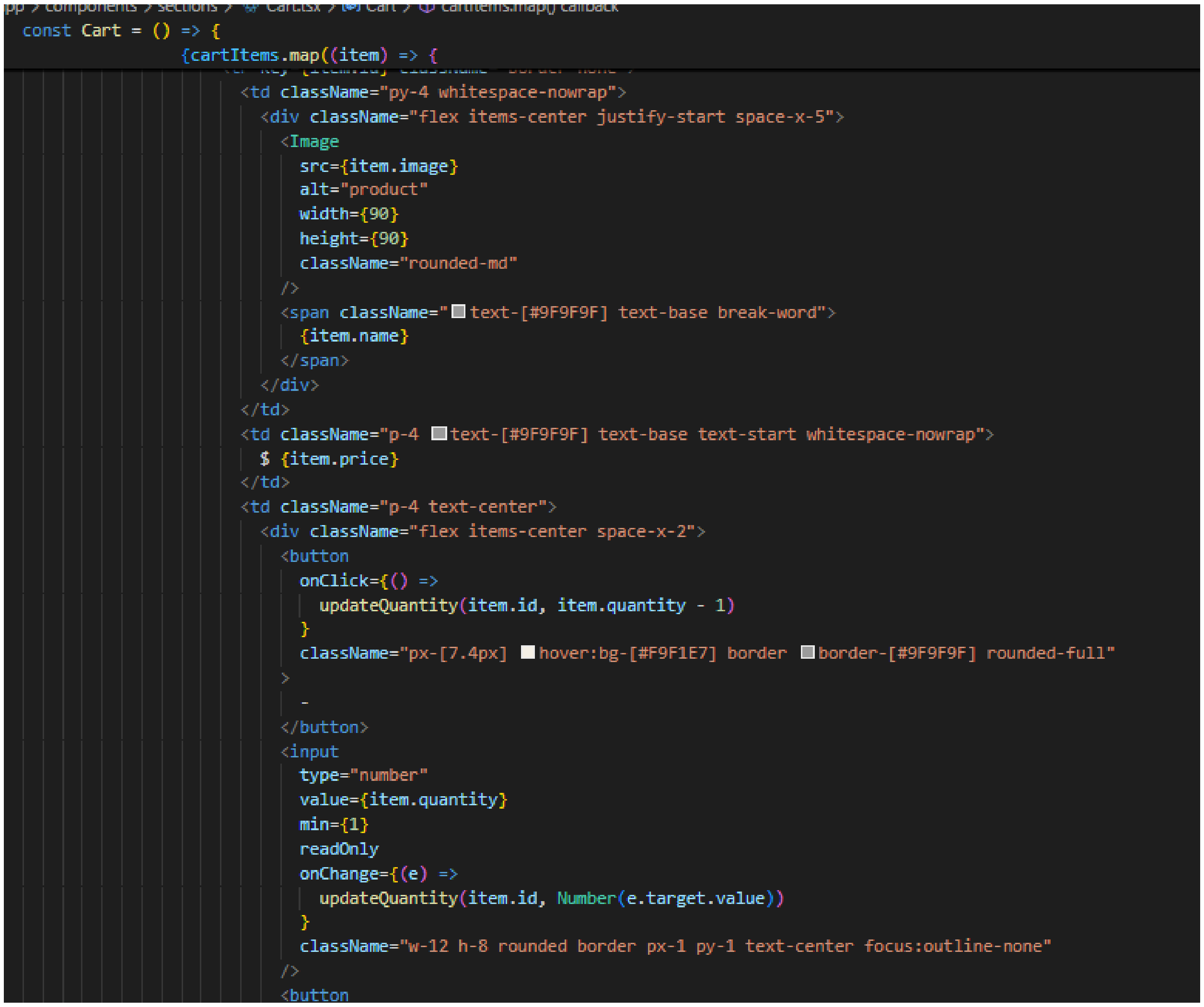
living fancy elegance desgin

- 1 + | Add To Cart | Stock: 26

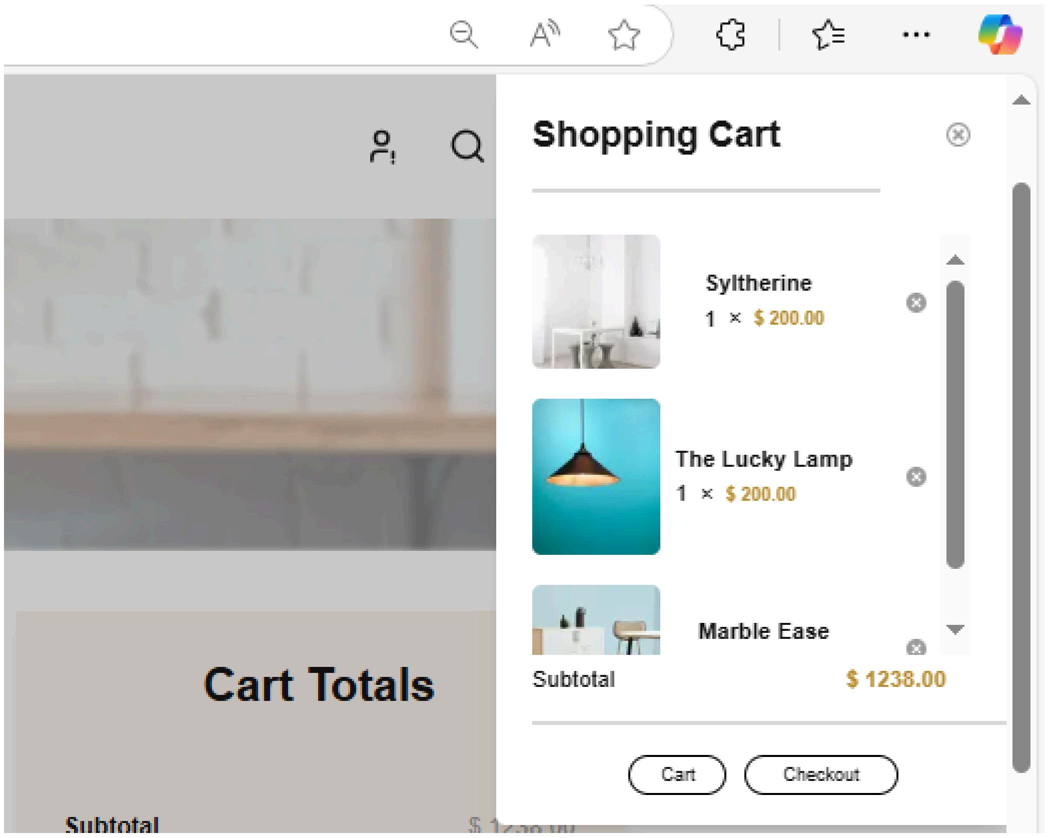
3. **Cart Component:** A cart was built to track added items, their quantities, and the total price. This component uses state management to keep track of cart contents, ensuring that it updates in real-time as users add or remove products.



```
const CartItems = ({ closeCart }: { closeCart: () => void }) => {
        {cartItems.length > 0 ? (
          cartItems.map((item) => (
            <div
              key={item.id}
              className="flex gap-2 sm:gap-0 flex-col sm:flex-row items-center justify-between"
            >
              <div>
                <Image
                  src={item.image}
                  alt={item.name}
                  width={85}
                  height={75}
                  className="rounded-md"
                />
              </div>
              <div className="mx-0 text-center sm:text-start sm:mr-6">
                <p className="text-[15px] font-semibold">{item.name}</p>
                <div>
                  <p className="flex items-center gap-2">
                    <span className="text-[15px]">{item.quantity}</span>
                    <span className="text-base">&times;</span>
                    <span className="text-xs text-[#B88E2F] font-semibold">
                      $ {item.price}.00
                    </span>
                  </p>
                </div>
              </div>
              <button
                className="text-[#9F9F9F] cursor-pointer mr-0 sm:mr-2"
                onClick={() => removeFromCart(item.id)}
              >
                <MdCancel />
              </button>
            </div>
          ))
        ) : (
          <p className="text-center text-gray-500">Your cart is empty.</p>
        )}
```



```
pp > components > sections > Cart.tsx > Cart > cartItems.map() callback
const Cart = () => {
                {cartItems.map((item) => {
                    <tr key={item.id} className="border-none">
                      <td className="py-4 whitespace-nowrap">
                        <div className="flex items-center justify-start space-x-5">
                          <Image
                            src={item.image}
                            alt="product"
                            width={90}
                            height={90}
                            className="rounded-md"
                          />
                          <span className="text-[#9F9F9F] text-base break-word">
                            {item.name}
                          </span>
                        </div>
                      </td>
                      <td className="p-4 text-[#9F9F9F] text-base text-start whitespace-nowrap">
                        $ {item.price}
                      </td>
                      <td className="p-4 text-center">
                        <div className="flex items-center space-x-2">
                          <button
                            onClick={() =>
                              updateQuantity(item.id, item.quantity - 1)
                            }
                            className="px-[7.4px] hover:bg-[#F9F1E7] border border-[#9F9F9F] rounded-full"
                          >
                            -
                          </button>
                          <input
                            type="number"
                            value={item.quantity}
                            min={1}
                            readOnly
                            onChange={(e) =>
                              updateQuantity(item.id, Number(e.target.value))
                            }
                            className="w-12 h-8 rounded border px-1 py-1 text-center focus:outline-none"
                          />
                          <button
```



Shopping Cart

Syltherine
1 × $ 200.00

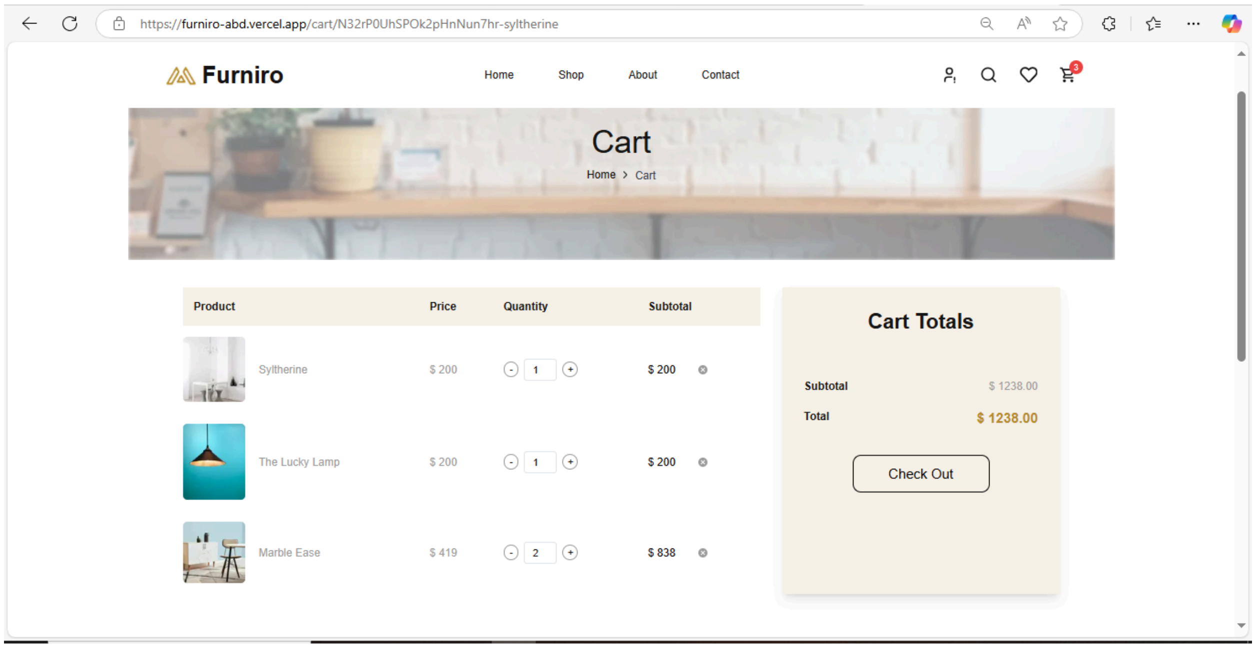
The Lucky Lamp
1 × $ 200.00

Marble Ease

Subtotal $ 1238.00

Cart Totals

Subtotal



https://furniro-abd.vercel.app/cart/N32rP0UhSPOk2pHnNun7hr-syltherine

Cart

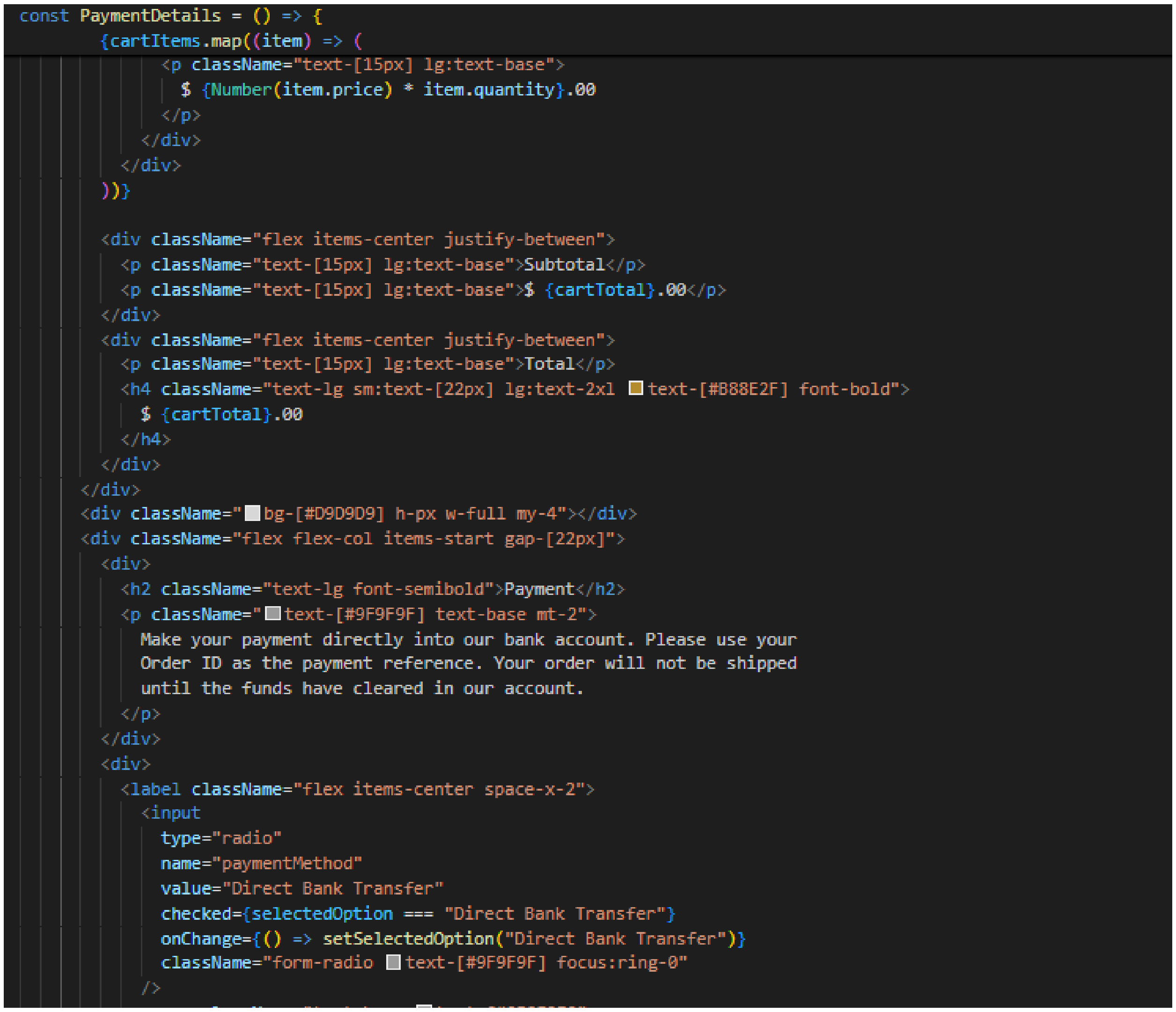
Home > Cart

Product	Price	Quantity	Subtotal
Syltherine	$ 200	1	$ 200
The Lucky Lamp	$ 200	1	$ 200
Marble Ease	$ 419	2	$ 838

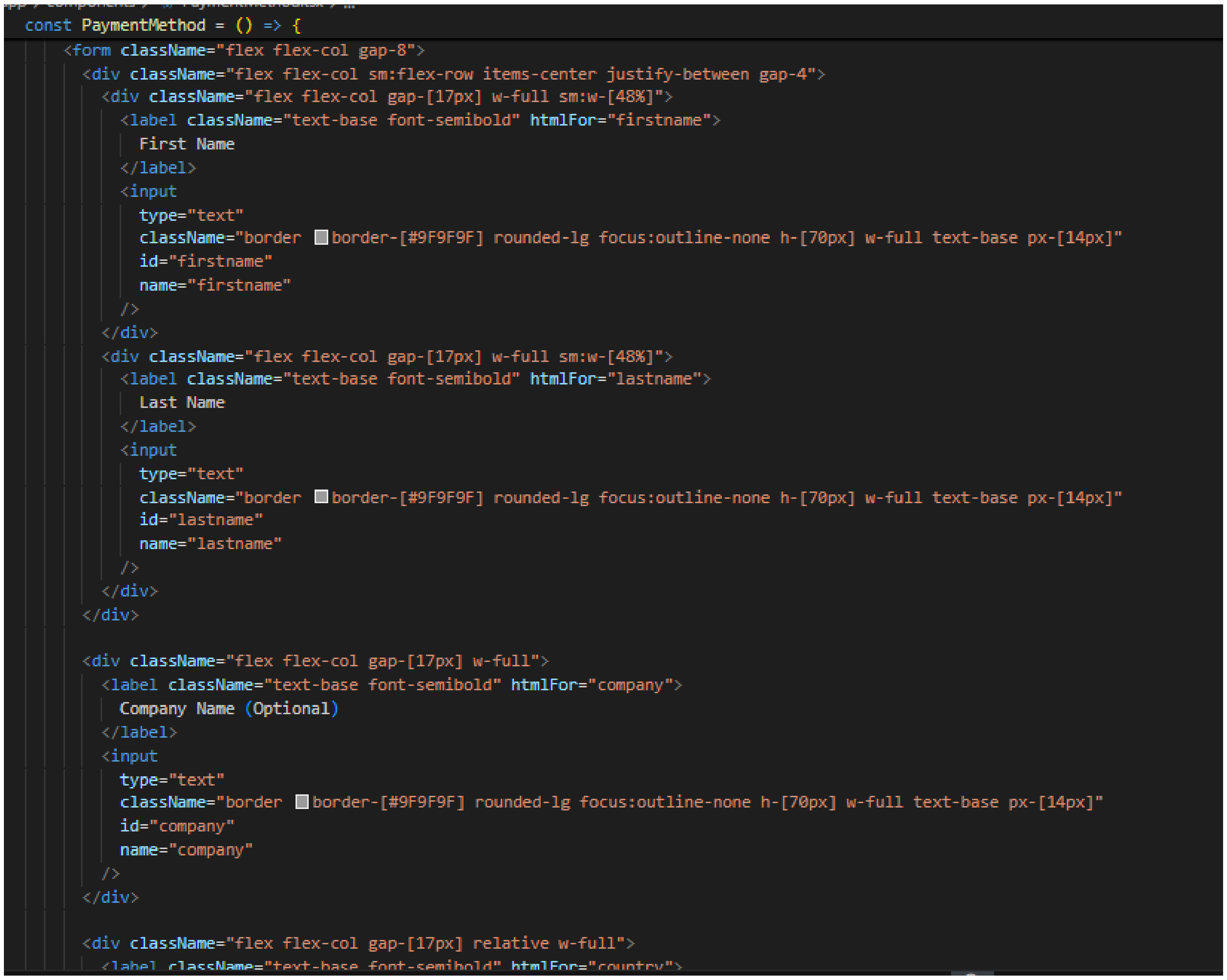
Cart Totals

Subtotal	$ 1238.00
Total	$ 1238.00

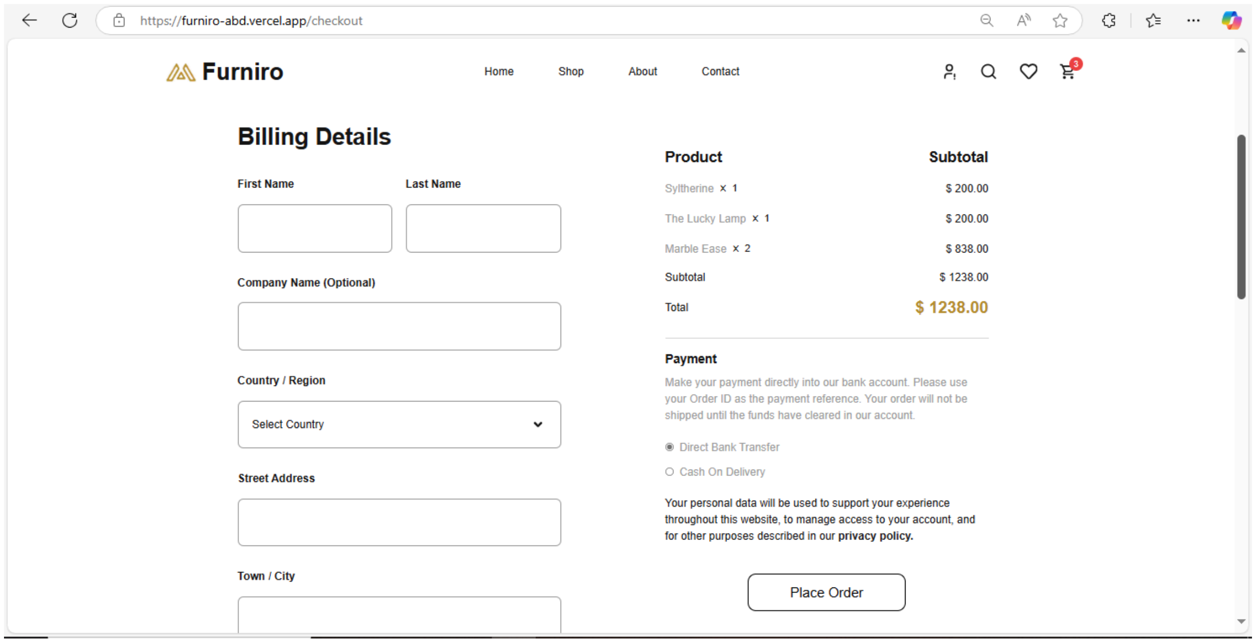
4. **Checkout Flow Component:** I have created a multi-step checkout process that collects billing and shipping details. While the payment details were mock implemented, this step is important for simulating the final purchase process in an e-commerce site.



```
const PaymentDetails = () => {
        {cartItems.map((item) => (
              <p className="text-[15px] lg:text-base">
                $ {Number(item.price) * item.quantity}.00
              </p>
            </div>
          </div>
        ))}

        <div className="flex items-center justify-between">
          <p className="text-[15px] lg:text-base">Subtotal</p>
          <p className="text-[15px] lg:text-base">$ {cartTotal}.00</p>
        </div>
        <div className="flex items-center justify-between">
          <p className="text-[15px] lg:text-base">Total</p>
          <h4 className="text-lg sm:text-[22px] lg:text-2xl text-[#B88E2F] font-bold">
            $ {cartTotal}.00
          </h4>
        </div>
      </div>
      <div className="bg-[#D9D9D9] h-px w-full my-4"></div>
      <div className="flex flex-col items-start gap-[22px]">
        <div>
          <h2 className="text-lg font-semibold">Payment</h2>
          <p className="text-[#9F9F9F] text-base mt-2">
            Make your payment directly into our bank account. Please use your
            Order ID as the payment reference. Your order will not be shipped
            until the funds have cleared in our account.
          </p>
        </div>
        <div>
          <label className="flex items-center space-x-2">
            <input
              type="radio"
              name="paymentMethod"
              value="Direct Bank Transfer"
              checked={selectedOption === "Direct Bank Transfer"}
              onChange={() => setSelectedOption("Direct Bank Transfer")}
              className="form-radio text-[#9F9F9F] focus:ring-0"
            />
```



```
const PaymentMethod = () => {
    <form className="flex flex-col gap-8">
      <div className="flex flex-col sm:flex-row items-center justify-between gap-4">
        <div className="flex flex-col gap-[17px] w-full sm:w-[48%]">
          <label className="text-base font-semibold" htmlFor="firstname">
            First Name
          </label>
          <input
            type="text"
            className="border border-[#9F9F9F] rounded-lg focus:outline-none h-[70px] w-full text-base px-[14px]"
            id="firstname"
            name="firstname"
          />
        </div>
        <div className="flex flex-col gap-[17px] w-full sm:w-[48%]">
          <label className="text-base font-semibold" htmlFor="lastname">
            Last Name
          </label>
          <input
            type="text"
            className="border border-[#9F9F9F] rounded-lg focus:outline-none h-[70px] w-full text-base px-[14px]"
            id="lastname"
            name="lastname"
          />
        </div>
      </div>

      <div className="flex flex-col gap-[17px] w-full">
        <label className="text-base font-semibold" htmlFor="company">
          Company Name (Optional)
        </label>
        <input
          type="text"
          className="border border-[#9F9F9F] rounded-lg focus:outline-none h-[70px] w-full text-base px-[14px]"
          id="company"
          name="company"
        />
      </div>

      <div className="flex flex-col gap-[17px] relative w-full">
        <label className="text-base font-semibold" htmlFor="country">
```



https://furniro-abd.vercel.app/checkout

Billing Details

First Name

Last Name

Company Name (Optional)

Country / Region

Select Country

Street Address

Town / City

Product	Subtotal
Syltherine × 1	$ 200.00
The Lucky Lamp × 1	$ 200.00
Marble Ease × 2	$ 838.00
Subtotal	$ 1238.00
Total	$ 1238.00

Payment

Make your payment directly into our bank account. Please use your Order ID as the payment reference. Your order will not be shipped until the funds have cleared in our account.

● Direct Bank Transfer

○ Cash On Delivery

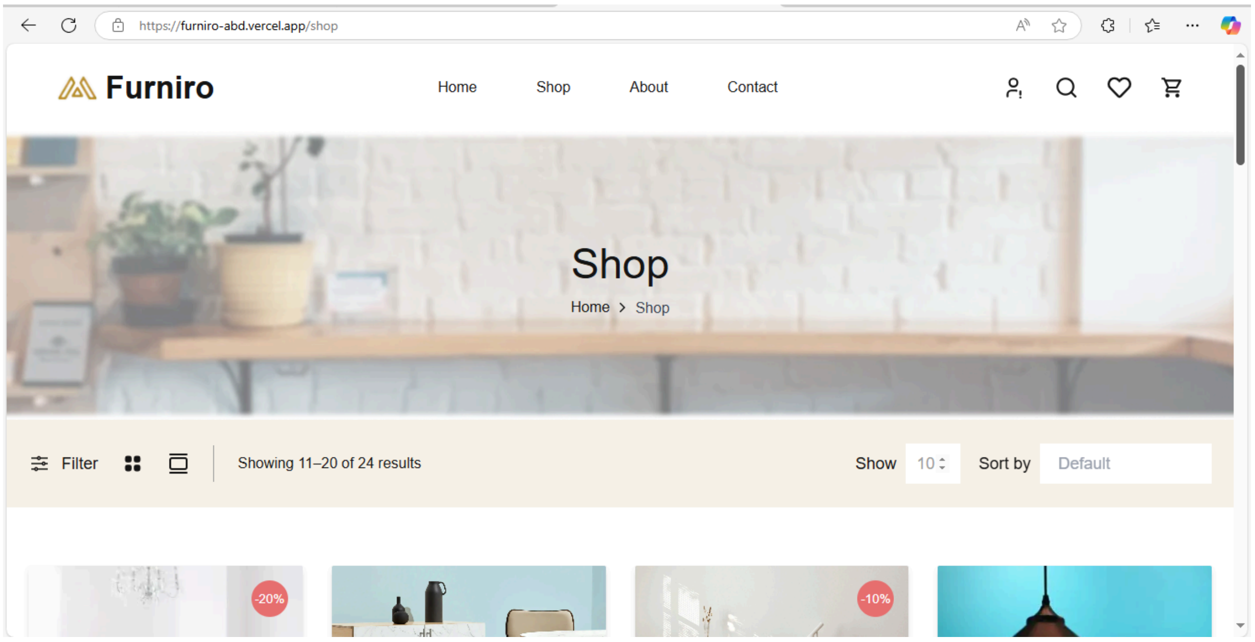
Your personal data will be used to support your experience throughout this website, to manage access to your account, and for other purposes described in our **privacy policy.**

Place Order

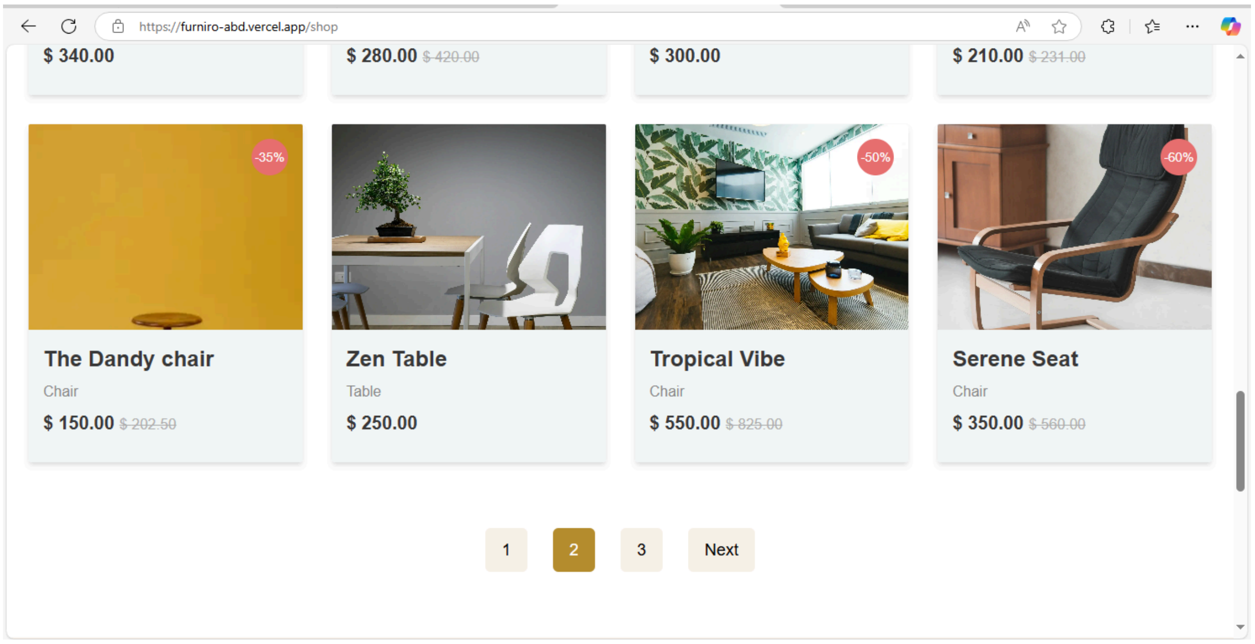
Advanced Features & Components:

Along with the core components, I had also worked on more advanced features that add significant value to the marketplace experience:

- **Pagination:** This component splits large product lists into manageable chunks, improving user experience by reducing page load times and making navigation smoother.



Showing 11–20 of 24 results



$ 340.00

$ 280.00 ~~$ 420.00~~

$ 300.00

$ 210.00 ~~$ 231.00~~

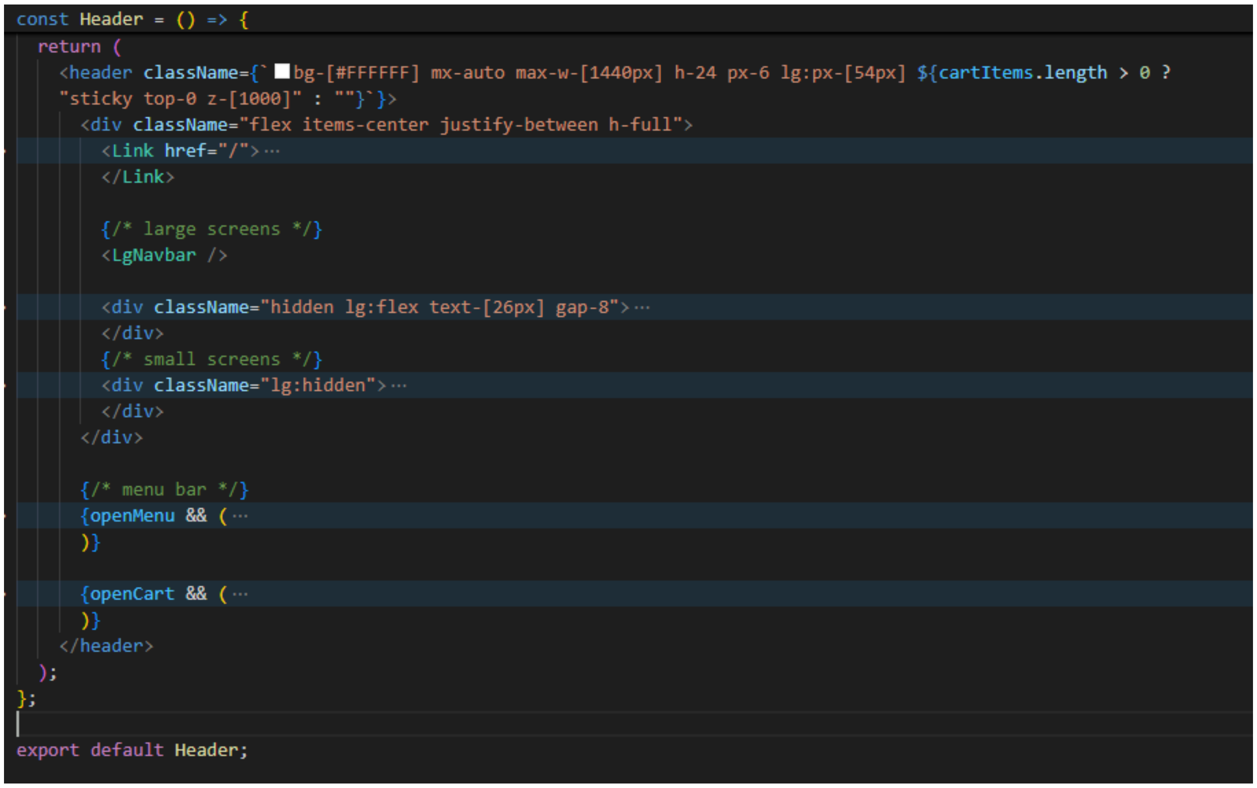
The Dandy chair
Chair
$ 150.00 ~~$ 202.50~~

Zen Table
Table
$ 250.00

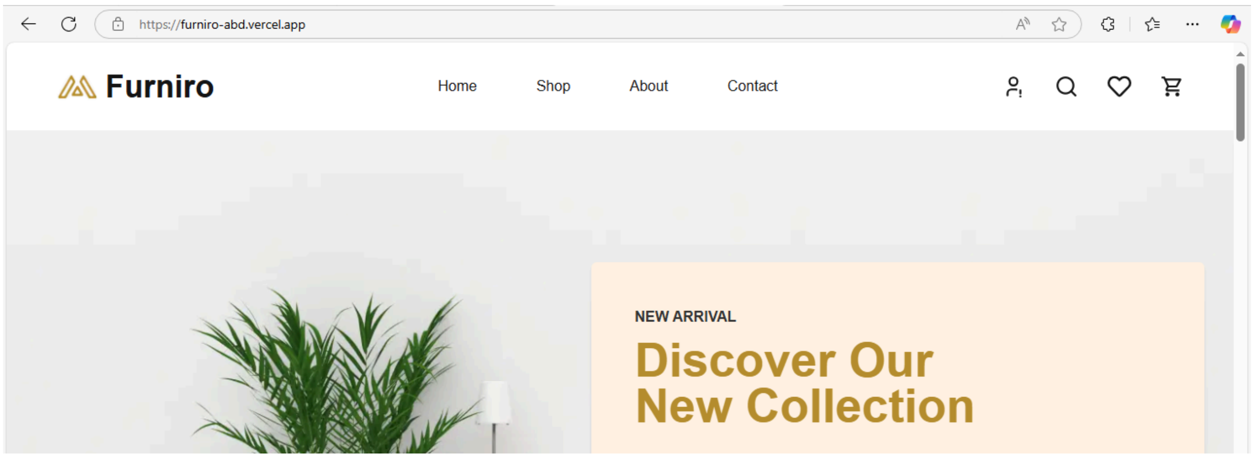
Tropical Vibe
Chair
$ 550.00 ~~$ 825.00~~

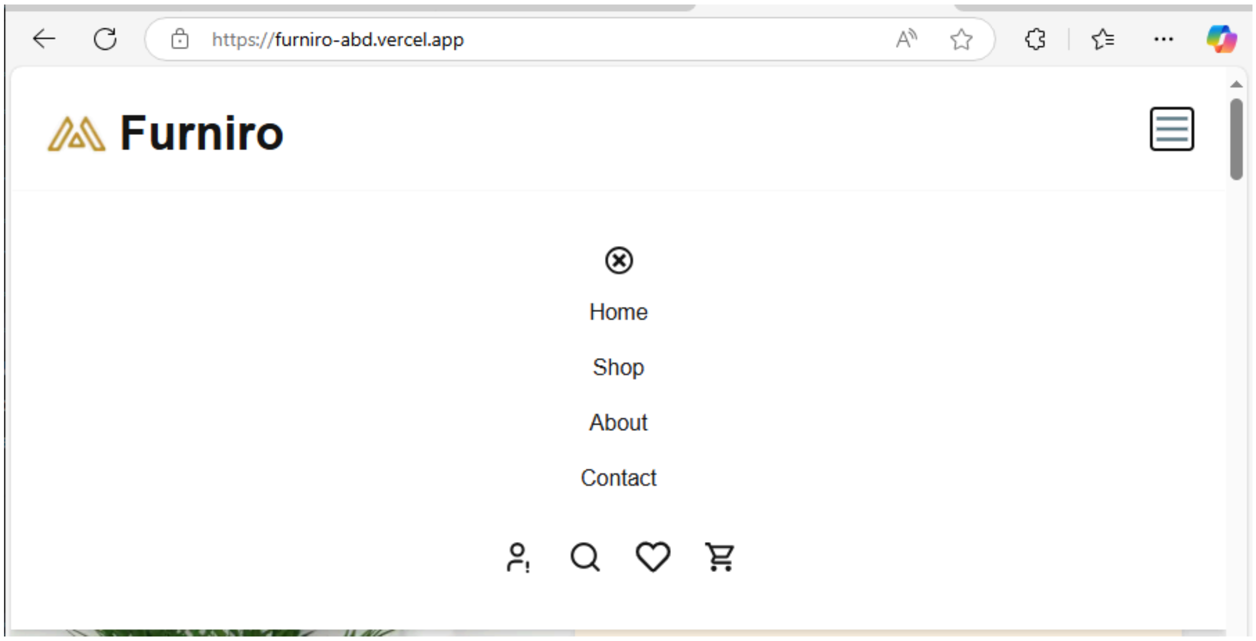
Serene Seat
Chair
$ 350.00 ~~$ 560.00~~

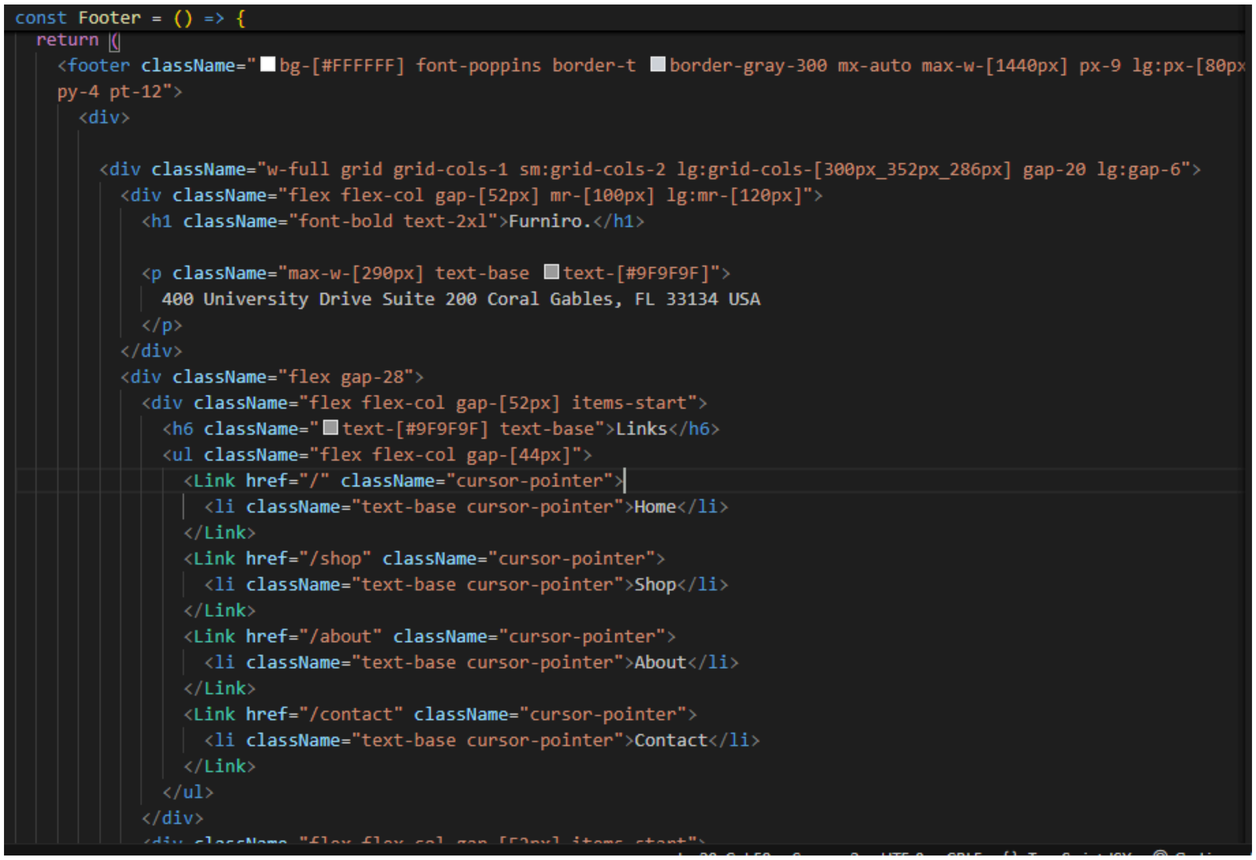
- **Footer & Header Components:** Consistent navigation across all pages, ensuring that users can easily find key sections of the marketplace, like Home, About, and Contact.



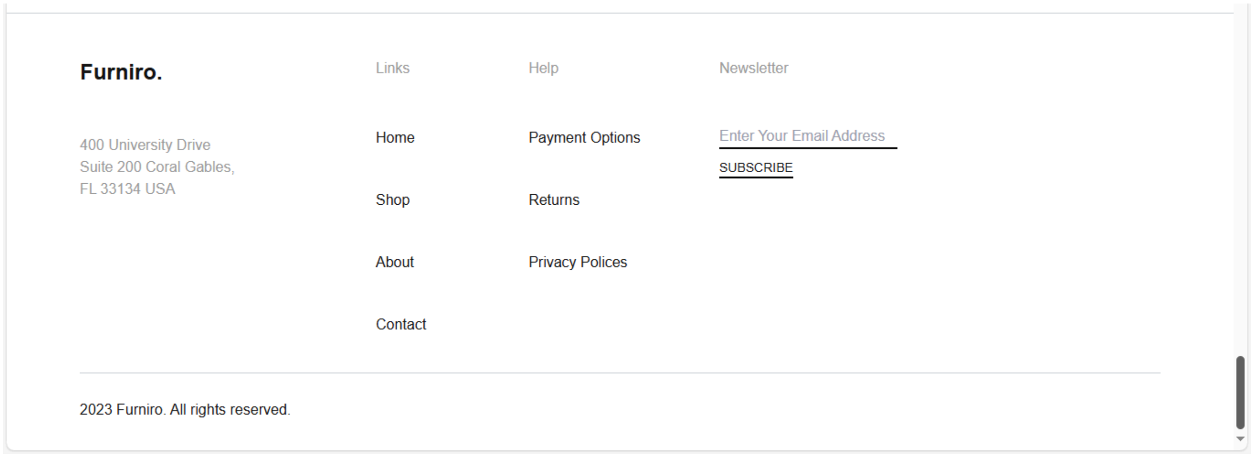
```
const Header = () => {
  return (
    <header className={` bg-[#FFFFFF] mx-auto max-w-[1440px] h-24 px-6 lg:px-[54px] ${cartItems.length > 0 ?
    "sticky top-0 z-[1000]" : ""}`}>
      <div className="flex items-center justify-between h-full">
        <Link href="/">…
        </Link>

        {/* large screens */}
        <LgNavbar />

        <div className="hidden lg:flex text-[26px] gap-8">…
        </div>
        {/* small screens */}
        <div className="lg:hidden">…
        </div>
      </div>

      {/* menu bar */}
      {openMenu && (…
      )}

      {openCart && (…
      )}
    </header>
  );
};

export default Header;
```







```
const Footer = () => {
  return (
    <footer className="bg-[#FFFFFF] font-poppins border-t border-gray-300 mx-auto max-w-[1440px] px-9 lg:px-[80px
    py-4 pt-12">
      <div>

        <div className="w-full grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-[300px_352px_286px] gap-20 lg:gap-6">
          <div className="flex flex-col gap-[52px] mr-[100px] lg:mr-[120px]">
            <h1 className="font-bold text-2xl">Furniro.</h1>

            <p className="max-w-[290px] text-base text-[#9F9F9F]">
              400 University Drive Suite 200 Coral Gables, FL 33134 USA
            </p>
          </div>
          <div className="flex gap-28">
            <div className="flex flex-col gap-[52px] items-start">
              <h6 className="text-[#9F9F9F] text-base">Links</h6>
              <ul className="flex flex-col gap-[44px]">
                <Link href="/" className="cursor-pointer">
                  <li className="text-base cursor-pointer">Home</li>
                </Link>
                <Link href="/shop" className="cursor-pointer">
                  <li className="text-base cursor-pointer">Shop</li>
                </Link>
                <Link href="/about" className="cursor-pointer">
                  <li className="text-base cursor-pointer">About</li>
                </Link>
                <Link href="/contact" className="cursor-pointer">
                  <li className="text-base cursor-pointer">Contact</li>
                </Link>
              </ul>
            </div>
            <div className="flex flex-col gap-[52px] items-start">
```



Furniro.

400 University Drive
Suite 200 Coral Gables,
FL 33134 USA

Links

Home

Shop

About

Contact

Help

Payment Options

Returns

Privacy Polices

Newsletter

Enter Your Email Address

SUBSCRIBE

2023 Furniro. All rights reserved.

Frontend Best Practices:

Throughout the development process, I had followed best practices to ensure that my components are modular, efficient, and easy to manage:

1. **Reusable Components:** By designing components like `ProductCard`, `SecondaryHeader`, `ServicesBar` `and more.` I have created reusable building blocks that can be used across different pages. This keeps my code clean and makes it easier to add new features in the future.
2. **State Management:** I had used React's state management, including `useState` for local component state and `useContext` for global state management. This ensures that data flows smoothly across components, providing a seamless user experience.
3. **Styling:** I had used modern styling libraries like Tailwind CSS to ensure design is responsive and looks great on any screen. Tailwind's utility-first approach also made it easy to create flexible layouts that adapt to different devices.

4. **Performance Optimization:** Techniques like lazy loading for images and pagination or infinite scrolling for large datasets were applied to ensure that my application performs well even as the product catalog grows.

Expected Output by End of Day 4:

By the end of Day 4, I have a fully functional marketplace with dynamic components such as:

- A product listing page displaying dynamic data from Sanity CMS or APIs.
- Individual product detail pages using dynamic routing.
- A functional search bar to filter products.
- Advanced category filters to allow users to refine product views.
- A cart component to track added items and a wishlist for saving future purchases.
- Responsive and professionally styled components, ensuring a smooth user experience across all devices.

Conclusion: Day 4 focused on bringing my marketplace to life by building dynamic and reusable frontend components. With these components, I've created a solid foundation for my project that can be expanded as I continue to work with backend data and add even more features. The skills I have developed today will be useful in any real-world web development project.