

# Day 4 - Dynamic Frontend Components - [Comforty E-commerce Store]

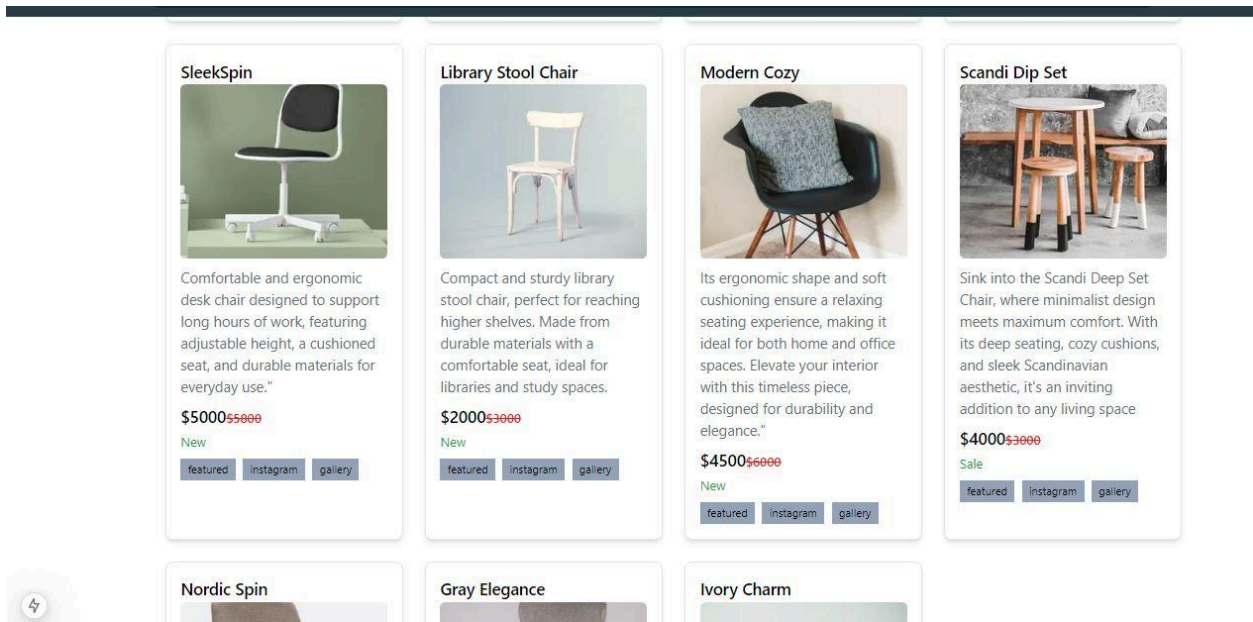
## Responsive Components

Responsive components in a website are designed to adapt their layout, style, and functionality based on the screen size and device type (desktop, tablet, or mobile). They ensure a seamless user experience by using techniques like media queries, flexible grids, and responsive images.

## Sanity Data Fetching

The website fetches data from **Sanity.io**, a headless CMS, to manage and deliver content dynamically. Using GROQ queries or APIs, the backend retrieves structured data from Sanity, which is then displayed on the frontend. This ensures real-time content updates and seamless integration between the CMS and the user interface, providing flexibility and scalability for content management.

## 1-Product Page:



The screenshot shows a VS Code editor with the Explorer panel on the left and the Source panel on the right. The Explorer panel shows the project structure with folders like 'public', 'scripts', 'src', and 'app'. The 'app' folder is expanded, showing 'Components'. The Source panel shows the code for 'page.tsx' in the 'src > app > Components > TopProducts > page.tsx' file. The code defines a function 'AllProducts' that uses 'useState' to manage a list of products and 'useEffect' to fetch products from an API. It also includes JSX elements for rendering the product list.

```
4 import Image from "next/image";
5 import { Product } from "../../types/products";
6 import { allProducts } from "@sanity/lib/queries";
7 import { urlFor } from "@sanity/lib/image";
8 import Link from "next/link";
9
10 // import {OurProducts} from "@app/OurProducts/[slug]/page"
11
12 const AllProducts = () => {
13   const [products, setProducts] = useState<Product[]>([]);
14
15   useEffect(() => {
16     async function fetchProduct() {
17       const fetchedProduct: Product[] = await client.fetch(allProducts);
18       setProducts(fetchedProduct);
19     }
20     fetchProduct();
21   }, []);
22
23   return (
24     <div className="max-w-6xl mx-auto px-4 py-8">
25       <h1 className="text-2xl font-bold mb-6 text-center">Our Products</h1>
26       <div className="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-6">
27         {products.map((product) => (
28           <div>
```


## 2-Dynamic Routing:

The screenshot shows a VS Code editor with the Explorer panel on the left and the Source panel on the right. The Explorer panel shows the project structure with folders like 'src', 'app', 'Checkout', 'contactus', 'context', 'FAQ', 'fonts', 'OurProducts', 'payment', 'privacyPolicy', 'return-policy', 'shipping-information', 'shop', 'studio', 'success', 'TermsAndConditions', 'wishlist', and 'page.tsx'. The 'OurProducts' folder is expanded, showing 'page.tsx'. The Source panel shows the code for 'page.tsx' in the 'src > app > OurProducts > [slug] > page.tsx' file. The code defines a function 'ProductPage' that uses 'getProduct' to fetch a product by slug. It also includes JSX elements for rendering the product details, including a product image and title.

```
28 export default async function ProductPage({
29   params
30 }) {
31   const product = await getProduct(params.slug);
32
33   // Check if product exists
34   if (!product) {
35     return (
36       <div className="max-w-7xl mx-auto px-4">
37         <h1 className="text-2xl font-bold">Product Not Found</h1>
38         <p className="mt-4">The product you are looking for does not exist.</p>
39       </div>
40     );
41   }
42
43   return (
44     <div className="max-w-7xl mx-auto px-4">
45       <div className="grid grid-cols-1 md:grid-cols-2 gap-12">
46         <div className="aspect-square">
47           {product.image && (
48             <Image
49               src={urlFor(product.image).url()}
50               alt={product.title}
51               width={500}
52               height={500}
53               className="rounded-md shadow-md"
54             />
55           )}
56         </div>
57       </div>
58     </div>
59   );
60 }
```

Product-1:

← → ↺ localhost:3000/OurProducts/rose-luxe-armchair 🔍 ☆ 📦 📄 🌐

A modern armchair with a tufted back and seat, upholstered in a light pink fabric. It has dark wooden legs and is set against a solid pink background.

## Rose Luxe Armchair

\$4520

~~\$6530~~

Sales


Rose Luxe Armchair offers a blend of elegance and comfort, featuring a plush design with high-quality upholstery and a sturdy frame, perfect for adding a touch of sophistication to any living space.

Category: Wing Chair

[featured](#) [instagram](#) [gallery](#)

### Product-2:

← → ↺ localhost:3000/OurProducts/ivory-charm 🔍 ☆ 📦 📄 🌐

A modern armchair with a tufted back and seat, upholstered in a light cream or ivory fabric. It has white, slightly curved legs and is set against a solid light green background.

## Ivory Charm

\$2800

~~\$3000~~


New

Elevate your space with the elegant Ivory Charm Chair. Featuring a sleek design, plush cushioning, and a soft ivory finish, this chair blends comfort with style, making it a perfect addition to any modern living room or office.

Category: Wing Chair

[featured](#) [instagram](#) [gallery](#)

### Product-3:



## SleekSpin

\$5000

~~\$5800~~

New

Comfortable and ergonomic desk chair designed to support long hours of work, featuring adjustable height, a cushioned seat, and durable materials for everyday use."

Category: Desk Chair

[featured](#) [instagram](#) [gallery](#)

Tags:

## Modern Cozy



Its ergonomic shape and soft cushioning ensure a relaxing seating experience, making it ideal for both home and office spaces. Elevate your interior with this timeless piece, designed for durability and elegance."

**\$4500** ~~\$6000~~

New

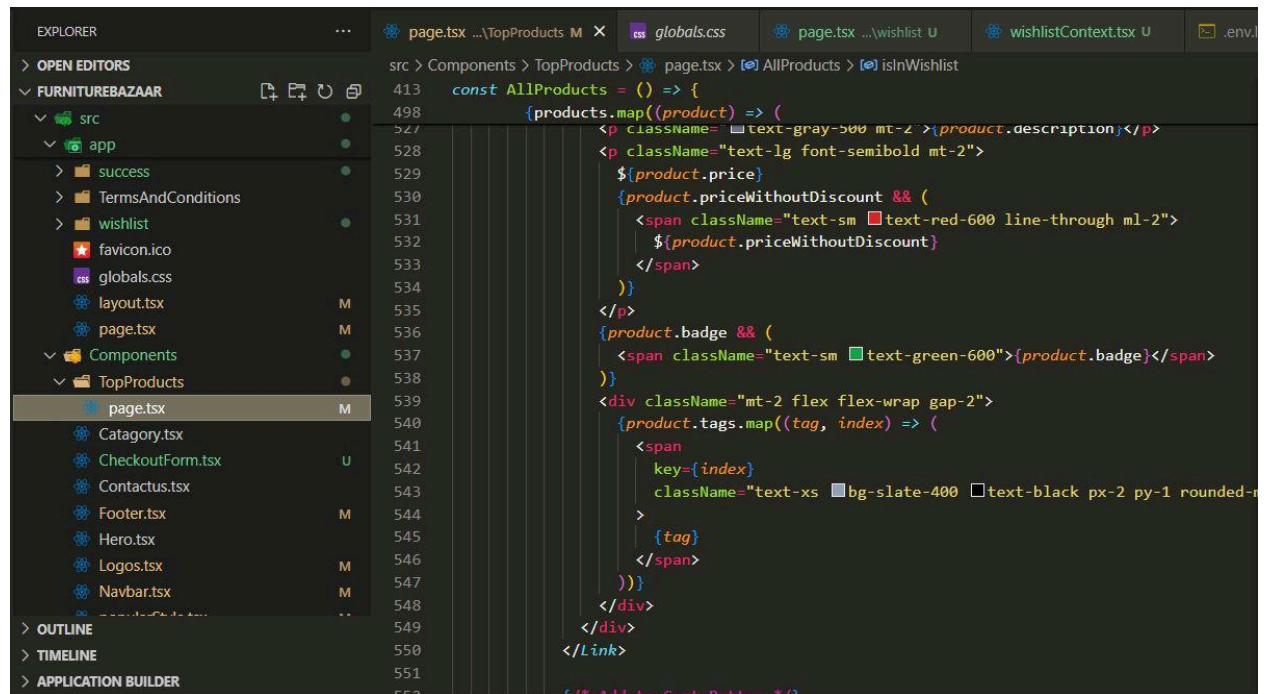
featured

instagram

gallery

Add to Cart

## Code snippet:



```
src > Components > TopProducts > page.tsx > AllProducts > isnWishlist
413 const AllProducts = () => {
498   {products.map((product) => (
527     <p className="text-gray-500 mt-2">{product.description}</p>
528     <p className="text-lg font-semibold mt-2">
529       ${product.price}
530       {product.priceWithoutDiscount && (
531         <span className="text-sm text-red-600 line-through ml-2">
532           ${product.priceWithoutDiscount}
533         </span>
534       )}
535     </p>
536     {product.badge && (
537       <span className="text-sm text-green-600">{product.badge}</span>
538     )}
539     <div className="mt-2 flex flex-wrap gap-2">
540       {product.tags.map((tag, index) => (
541         <span
542           key={index}
543           className="text-xs bg-slate-400 text-black px-2 py-1 rounded-r
544         >
545           {tag}
546         </span>
547       )}
548     </div>
549   </Link>
550   </div>
551   </div>
552   </div>
553   </div>
554   </div>
555   </div>
556   </div>
557   </div>
558   </div>
559   </div>
560   </div>
561   </div>
562   </div>
563   </div>
564   </div>
565   </div>
566   </div>
567   </div>
568   </div>
569   </div>
570   </div>
571   </div>
572   </div>
573   </div>
574   </div>
575   </div>
576   </div>
577   </div>
578   </div>
579   </div>
580   </div>
581   </div>
582   </div>
583   </div>
584   </div>
585   </div>
586   </div>
587   </div>
588   </div>
589   </div>
590   </div>
591   </div>
592   </div>
593   </div>
594   </div>
595   </div>
596   </div>
597   </div>
598   </div>
599   </div>
600   </div>
601   </div>
602   </div>
603   </div>
604   </div>
605   </div>
606   </div>
607   </div>
608   </div>
609   </div>
610   </div>
611   </div>
612   </div>
613   </div>
614   </div>
615   </div>
616   </div>
617   </div>
618   </div>
619   </div>
620   </div>
621   </div>
622   </div>
623   </div>
624   </div>
625   </div>
626   </div>
627   </div>
628   </div>
629   </div>
630   </div>
631   </div>
632   </div>
633   </div>
634   </div>
635   </div>
636   </div>
637   </div>
638   </div>
639   </div>
640   </div>
641   </div>
642   </div>
643   </div>
644   </div>
645   </div>
646   </div>
647   </div>
648   </div>
649   </div>
650   </div>
651   </div>
652   </div>
653   </div>
654   </div>
655   </div>
656   </div>
657   </div>
658   </div>
659   </div>
660   </div>
661   </div>
662   </div>
663   </div>
664   </div>
665   </div>
666   </div>
667   </div>
668   </div>
669   </div>
670   </div>
671   </div>
672   </div>
673   </div>
674   </div>
675   </div>
676   </div>
677   </div>
678   </div>
679   </div>
680   </div>
681   </div>
682   </div>
683   </div>
684   </div>
685   </div>
686   </div>
687   </div>
688   </div>
689   </div>
690   </div>
691   </div>
692   </div>
693   </div>
694   </div>
695   </div>
696   </div>
697   </div>
698   </div>
699   </div>
700   </div>
701   </div>
702   </div>
703   </div>
704   </div>
705   </div>
706   </div>
707   </div>
708   </div>
709   </div>
710   </div>
711   </div>
712   </div>
713   </div>
714   </div>
715   </div>
716   </div>
717   </div>
718   </div>
719   </div>
720   </div>
721   </div>
722   </div>
723   </div>
724   </div>
725   </div>
726   </div>
727   </div>
728   </div>
729   </div>
730   </div>
731   </div>
732   </div>
733   </div>
734   </div>
735   </div>
736   </div>
737   </div>
738   </div>
739   </div>
740   </div>
741   </div>
742   </div>
743   </div>
744   </div>
745   </div>
746   </div>
747   </div>
748   </div>
749   </div>
750   </div>
751   </div>
752   </div>
753   </div>
754   </div>
755   </div>
756   </div>
757   </div>
758   </div>
759   </div>
760   </div>
761   </div>
762   </div>
763   </div>
764   </div>
765   </div>
766   </div>
767   </div>
768   </div>
769   </div>
770   </div>
771   </div>
772   </div>
773   </div>
774   </div>
775   </div>
776   </div>
777   </div>
778   </div>
779   </div>
780   </div>
781   </div>
782   </div>
783   </div>
784   </div>
785   </div>
786   </div>
787   </div>
788   </div>
789   </div>
790   </div>
791   </div>
792   </div>
793   </div>
794   </div>
795   </div>
796   </div>
797   </div>
798   </div>
799   </div>
800   </div>
801   </div>
802   </div>
803   </div>
804   </div>
805   </div>
806   </div>
807   </div>
808   </div>
809   </div>
810   </div>
811   </div>
812   </div>
813   </div>
814   </div>
815   </div>
816   </div>
817   </div>
818   </div>
819   </div>
820   </div>
821   </div>
822   </div>
823   </div>
824   </div>
825   </div>
826   </div>
827   </div>
828   </div>
829   </div>
830   </div>
831   </div>
832   </div>
833   </div>
834   </div>
835   </div>
836   </div>
837   </div>
838   </div>
839   </div>
840   </div>
841   </div>
842   </div>
843   </div>
844   </div>
845   </div>
846   </div>
847   </div>
848   </div>
849   </div>
850   </div>
851   </div>
852   </div>
853   </div>
854   </div>
855   </div>
856   </div>
857   </div>
858   </div>
859   </div>
860   </div>
861   </div>
862   </div>
863   </div>
864   </div>
865   </div>
866   </div>
867   </div>
868   </div>
869   </div>
870   </div>
871   </div>
872   </div>
873   </div>
874   </div>
875   </div>
876   </div>
877   </div>
878   </div>
879   </div>
880   </div>
881   </div>
882   </div>
883   </div>
884   </div>
885   </div>
886   </div>
887   </div>
888   </div>
889   </div>
890   </div>
891   </div>
892   </div>
893   </div>
894   </div>
895   </div>
896   </div>
897   </div>
898   </div>
899   </div>
900   </div>
901   </div>
902   </div>
903   </div>
904   </div>
905   </div>
906   </div>
907   </div>
908   </div>
909   </div>
910   </div>
911   </div>
912   </div>
913   </div>
914   </div>
915   </div>
916   </div>
917   </div>
918   </div>
919   </div>
920   </div>
921   </div>
922   </div>
923   </div>
924   </div>
925   </div>
926   </div>
927   </div>
928   </div>
929   </div>
930   </div>
931   </div>
932   </div>
933   </div>
934   </div>
935   </div>
936   </div>
937   </div>
938   </div>
939   </div>
940   </div>
941   </div>
942   </div>
943   </div>
944   </div>
945   </div>
946   </div>
947   </div>
948   </div>
949   </div>
950   </div>
951   </div>
952   </div>
953   </div>
954   </div>
955   </div>
956   </div>
957   </div>
958   </div>
959   </div>
960   </div>
961   </div>
962   </div>
963   </div>
964   </div>
965   </div>
966   </div>
967   </div>
968   </div>
969   </div>
970   </div>
971   </div>
972   </div>
973   </div>
974   </div>
975   </div>
976   </div>
977   </div>
978   </div>
979   </div>
980   </div>
981   </div>
982   </div>
983   </div>
984   </div>
985   </div>
986   </div>
987   </div>
988   </div>
989   </div>
990   </div>
991   </div>
992   </div>
993   </div>
994   </div>
995   </div>
996   </div>
997   </div>
998   </div>
999   </div>
1000  </div>
```

## 3-Top Categories:

Top categories allow users to easily navigate through the most popular or frequently visited sections of the website. These categories are prominently displayed to highlight trending products.



### Top Categories



**Wing Chair**

3,584 Products



**Wooden Chair**

157 Products



**Desk Chair**

154 Products



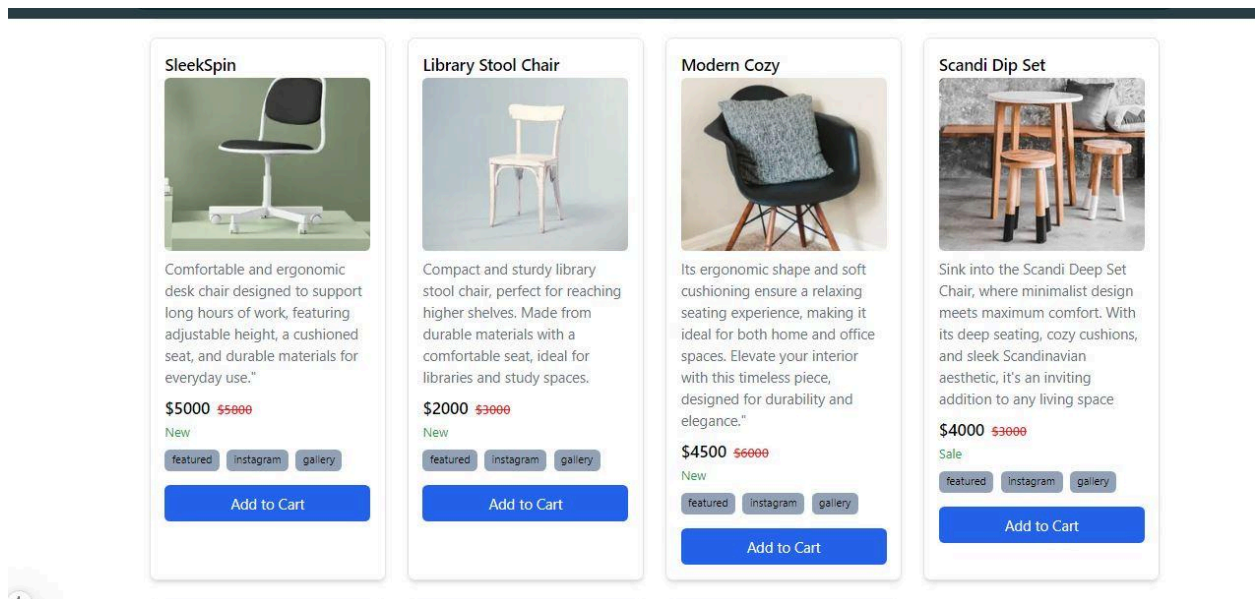
## 4-Add to Cart Functionality:

### Purpose of Add to Cart Feature:

- The "Add to Cart" functionality allows users to select products they wish to purchase and store them temporarily in a shopping cart. This provides users with the ability to review and manage their selected products before proceeding to checkout.

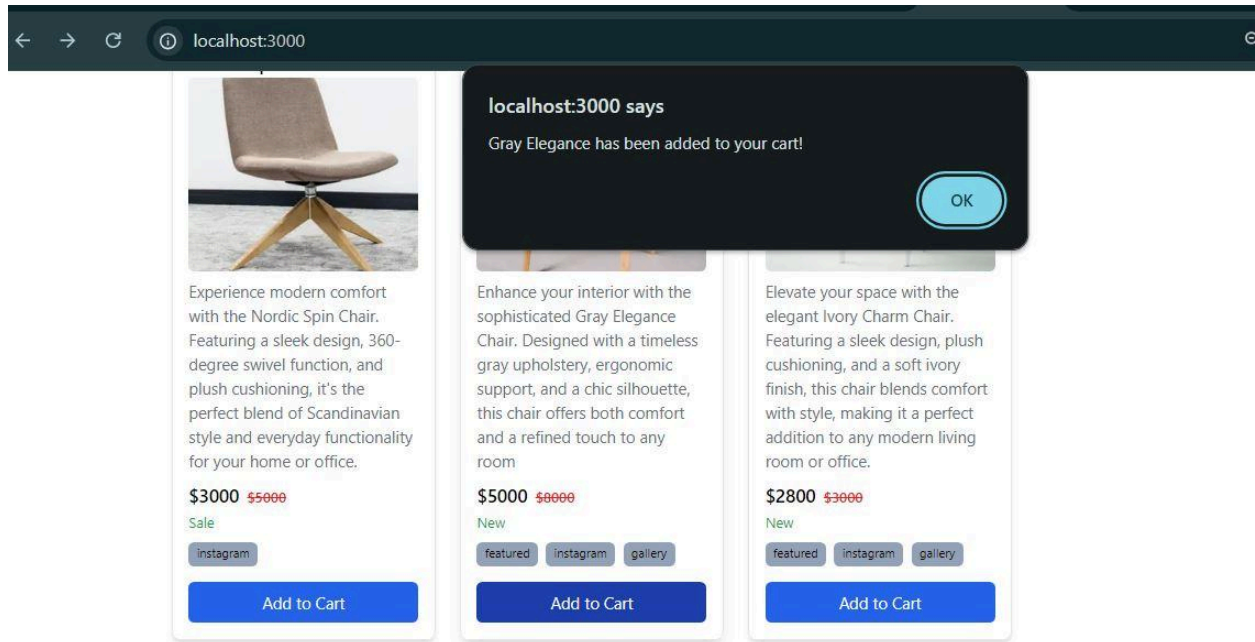
### Implementation Details:

- A state variable (cart) is used to store the list of products added by the user.
- A function named addToCart() is implemented to handle the addition of products to the cart. When a user clicks the "Add to Cart" button, the selected product is appended to the cart array.



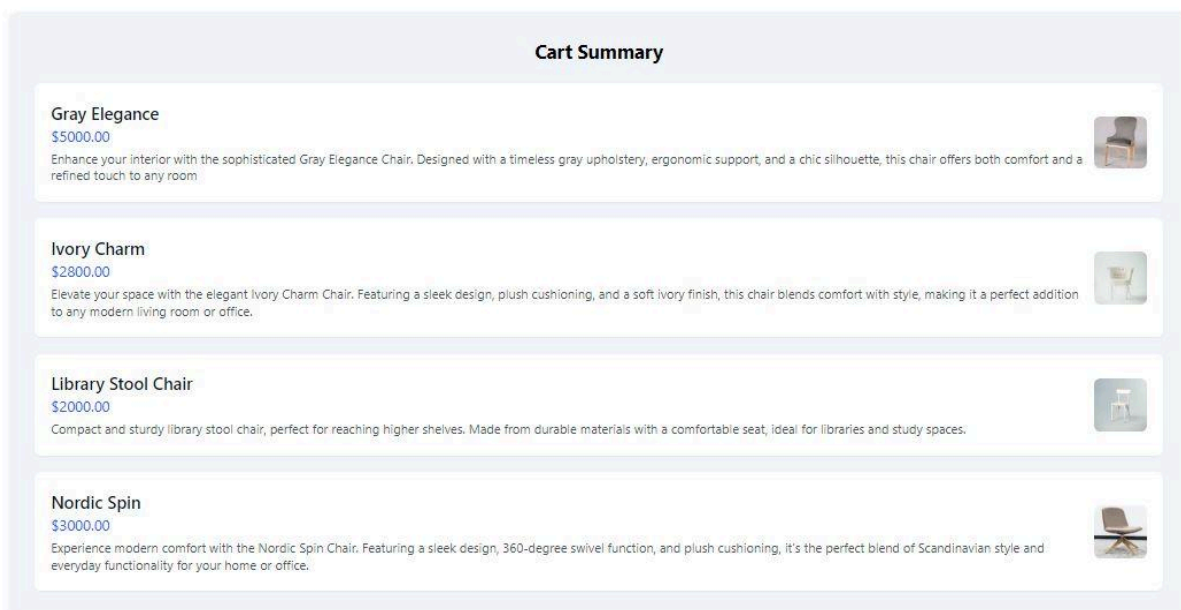
### Alert Notification:

- An alert box is shown to the user with a confirmation message such as "Product has been added to your cart!" for better feedback and interactivity.



## Dynamic Product Rendering:

- The products in the cart are dynamically rendered in the Cart Summary section. This section shows:
  - Product title, price, and image.
  - A description of the product.






## Remove from Cart Option:

- A Remove from Cart feature is also added, enabling users to delete unwanted products from their cart. This improves the overall user experience and provides flexibility to update the cart as needed.


### Cart Summary



**Ivory Charm**  
\$2800.00

Elevate your space with the elegant Ivory Charm Chair. Featuring a sleek design, plush cushioning, and a soft ivory finish, this chair blends comfort with style, making it a perfect addition to any modern living room or office.


Delete



**Gray Elegance**  
\$5000.00

Enhance your interior with the sophisticated Gray Elegance Chair. Designed with a timeless gray upholstery, ergonomic support, and a chic silhouette, this chair offers both comfort and a refined touch to any room.


Delete



**Modern Cozy**  
\$4500.00

Its ergonomic shape and soft cushioning ensure a relaxing seating experience, making it ideal for both home and office spaces. Elevate your interior with this timeless piece, designed for durability and elegance.

Delete

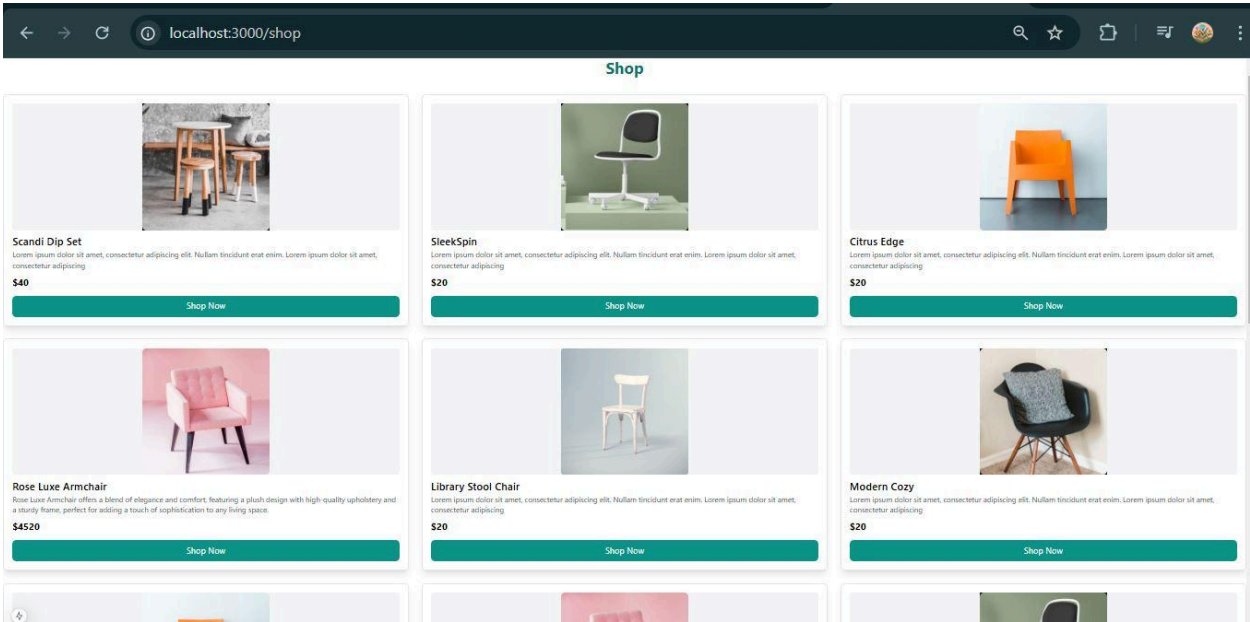


**Library Stool Chair**  
\$2000.00

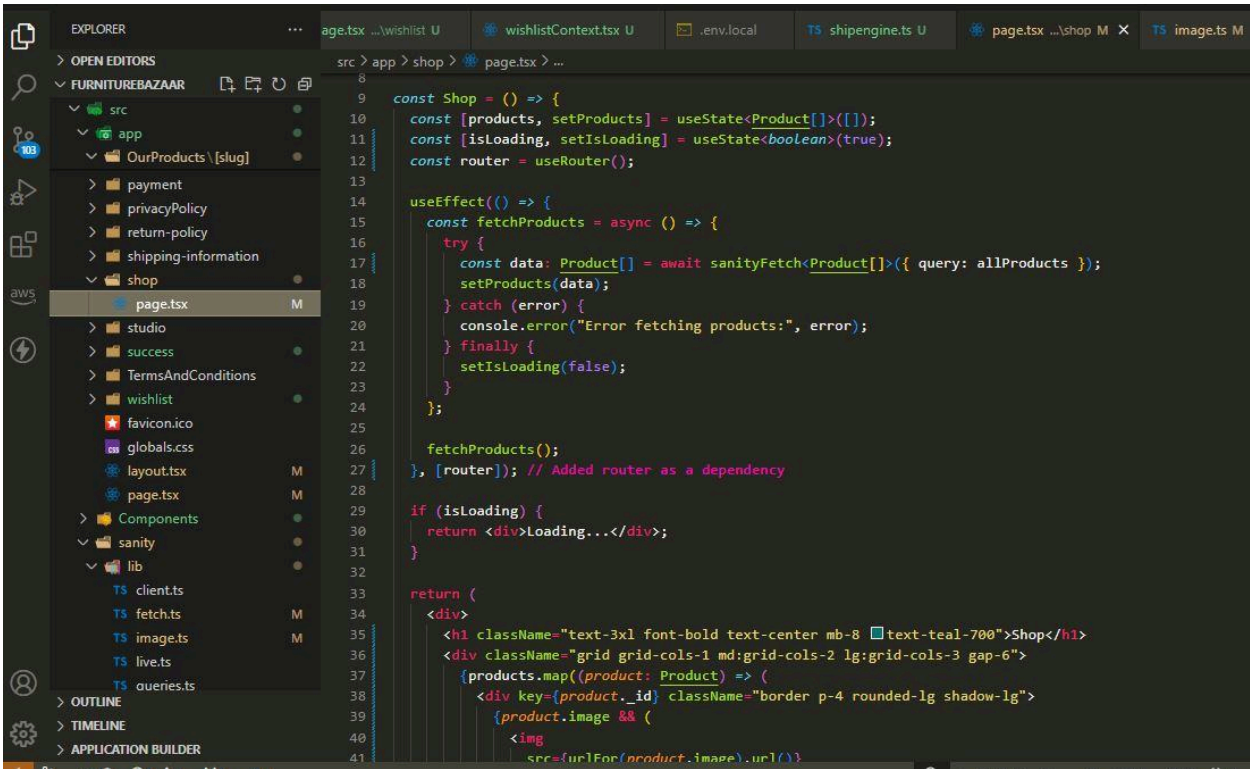
Compact and sturdy library stool chair, perfect for reaching higher shelves. Made from durable materials with a comfortable seat, ideal for libraries and study spaces.

Delete

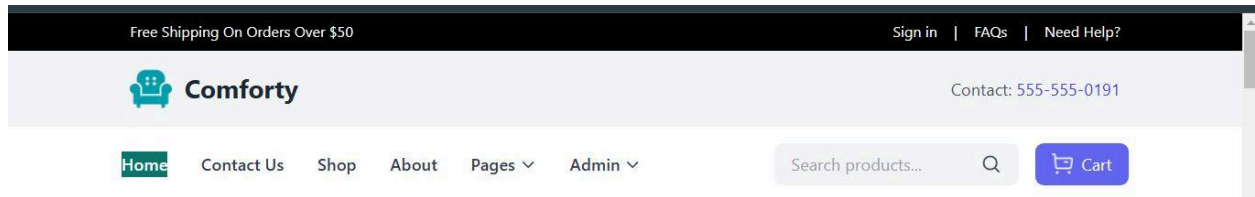
# 5-Shop Page:



# Code snippet:



## 6-SignUp/SignIn page:



**sign-up**

accounts.dev/sign-up?redirect\_url=http%3A%2F%2Flocalhost%3A3000%2F

## Create your account

Welcome! Please fill in the details to get started.



Continue with Google

or

**Username**

**Email address**

Enter your email address

**Password**

Enter your password



**Continue** ▶

Already have an account? [Sign in](#)


Secured by  Clerk

## Sign-In page

nt-29.accounts.dev/sign-in?redirect\_url=http%3A%2F%2Flocalhost%3A3000%2F

### Sign in to Comforty

Welcome back! Please sign in to continue

 Continue with Google


or

Email address or username

Enter email or username

Continue ▶

Don't have an account? [Sign up](#)

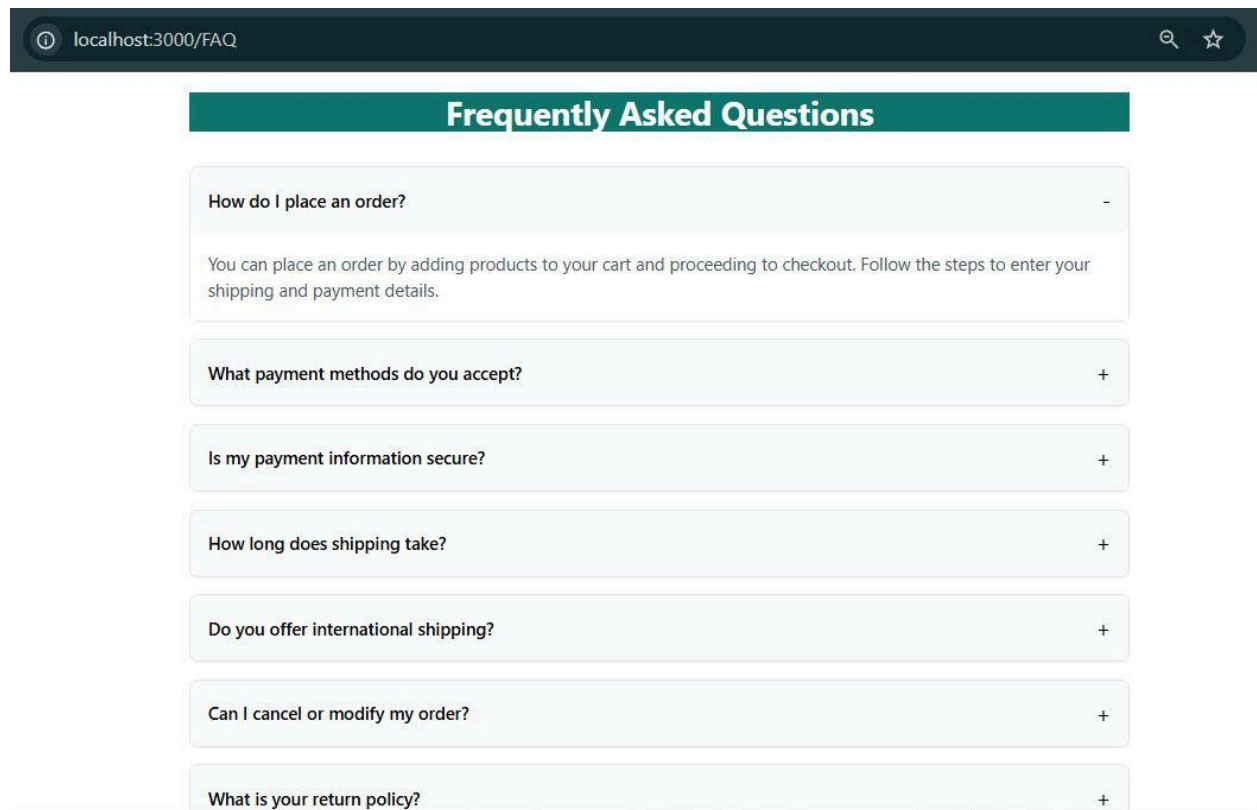
Secured by  clerk

Development mode

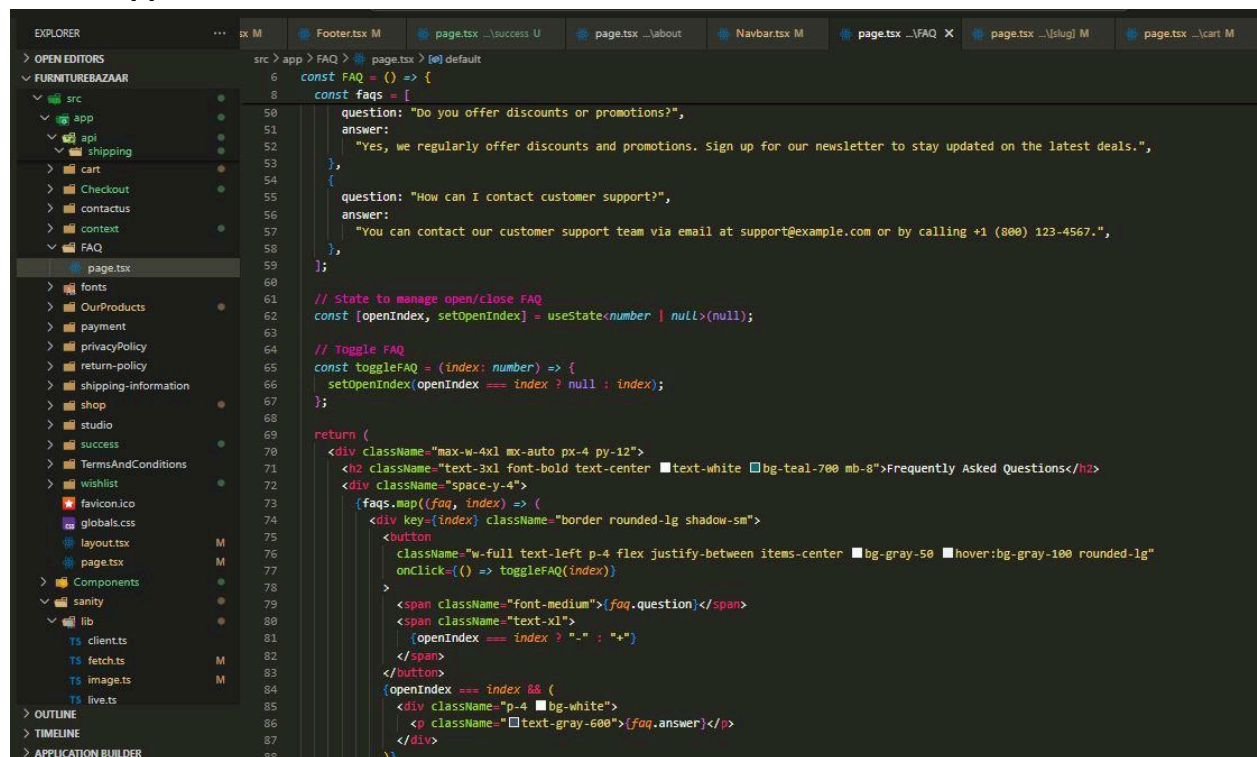
## 7-FAQs :

The FAQs (Frequently Asked Questions) section is designed to answer common user queries. It is often structured in an accordion or collapsible format, allowing users to expand and view answers easily.

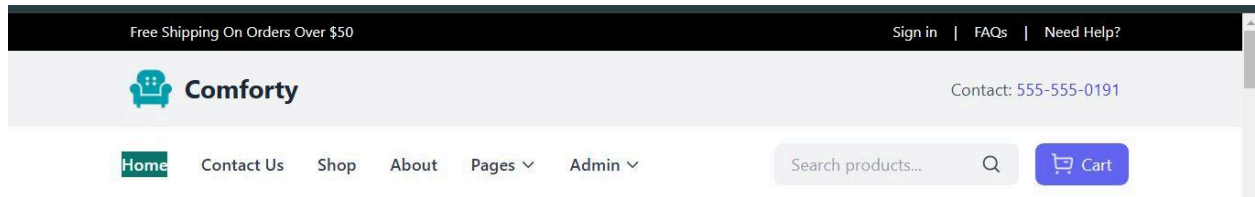




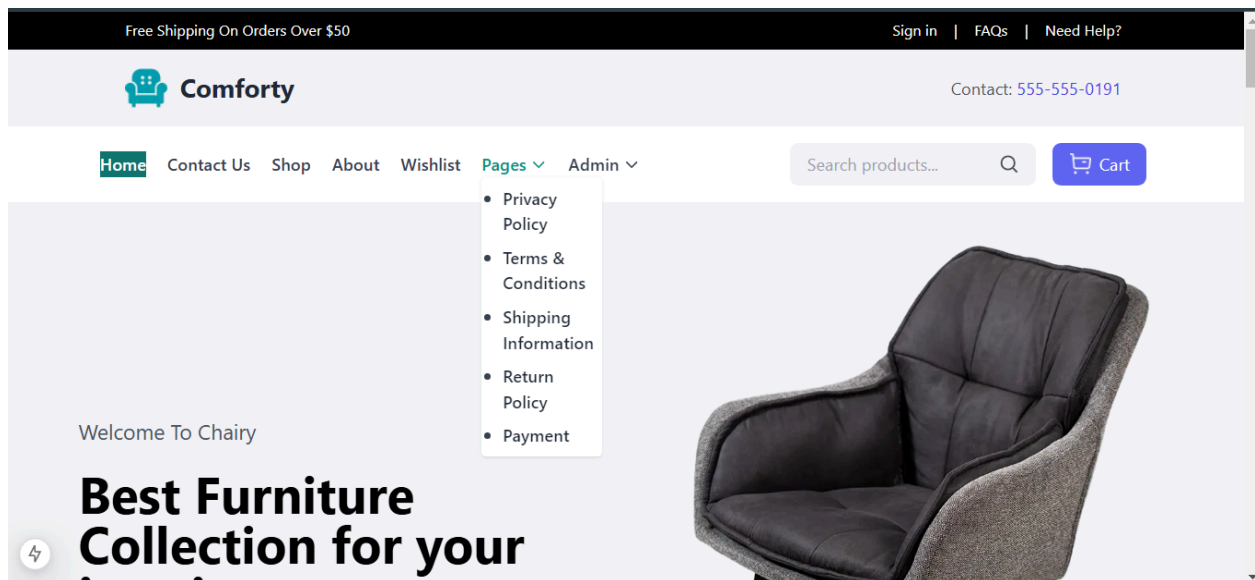
## Code snippet



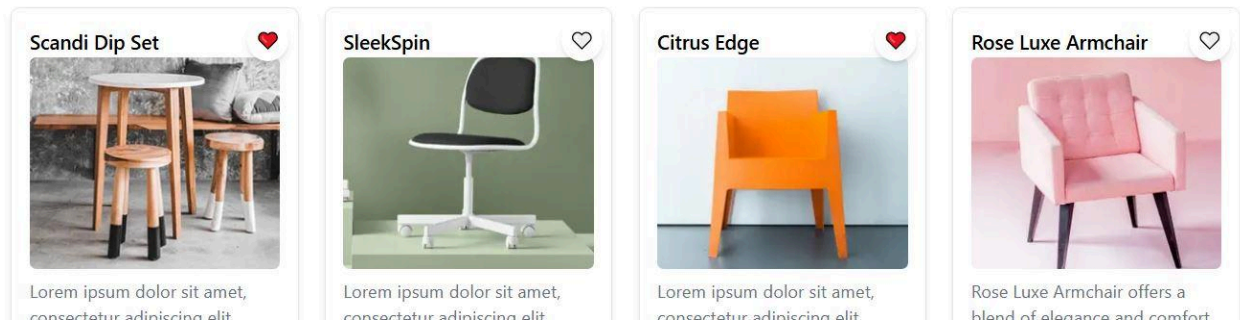
## 8: Search bar:



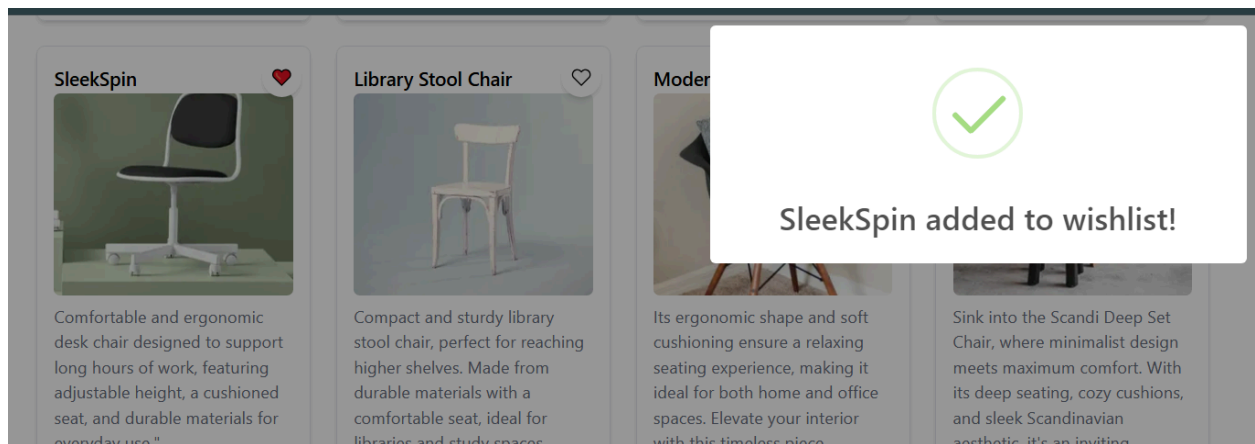
## 9- Navbar



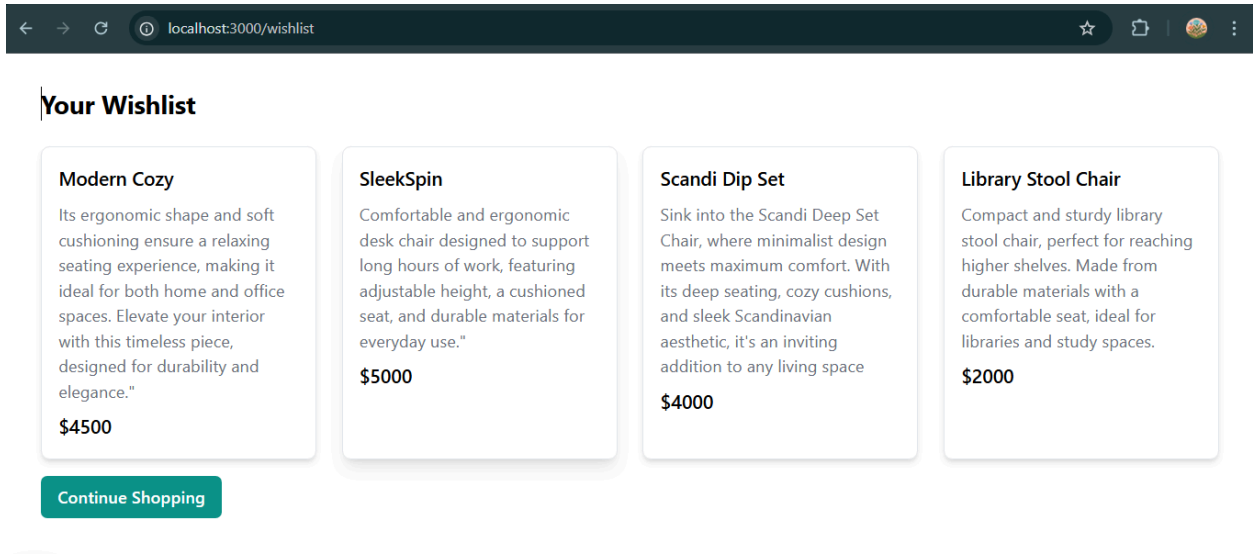
## 10-WishList:



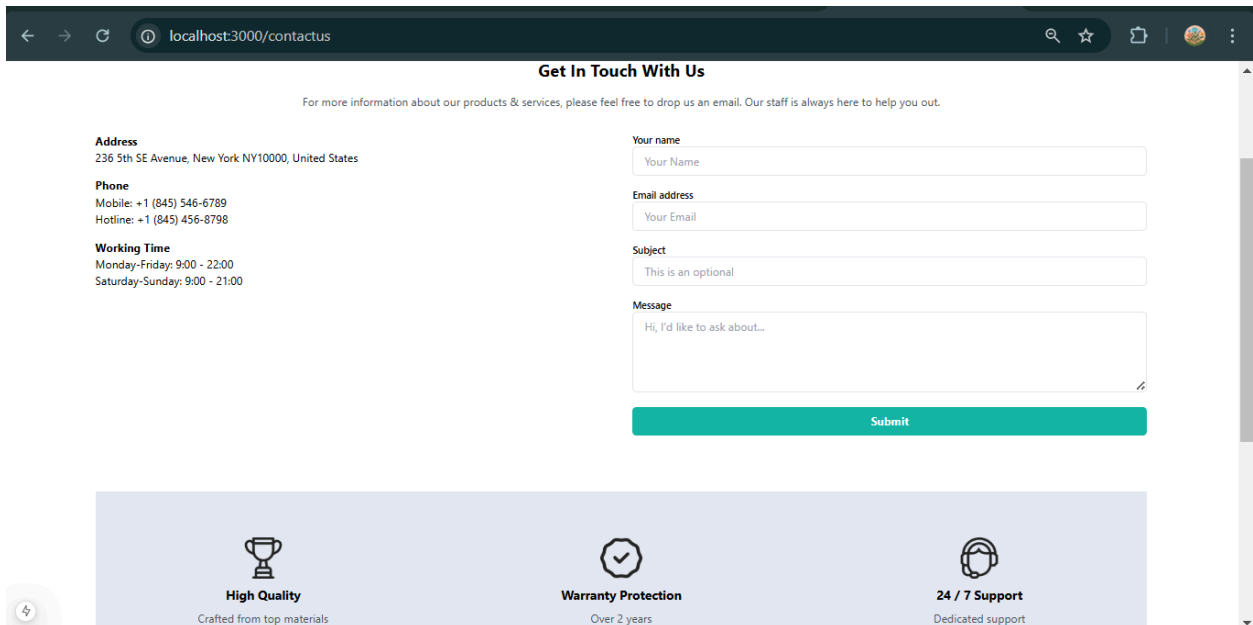
## Wishlist Notification:



## Wishlist page:



## 11-Contact-Us:



## 12-About-us:



localhost:3000/about



## About Us - Comforty

At Comforty, we believe that the right chair can transform any space. Our collection blends timeless design with innovation, crafted with precision. Discover chairs that seamlessly blend style with technology.

[View Collection](#)



## What Makes Our Brand Different



### Next day as standard

Order before 3PM and get your order the Next Day standard.



### Made by true artisans

Handmade Crafted goods made with real passion and craftsmanship.



### Unbeatable prices

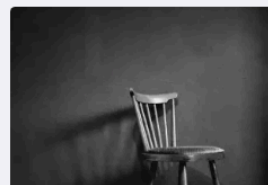
For our materials and quality you won't find better prices anywhere



### Recycled packaging

We use 100% recycled to ensure our footprint is more manageable.

## Our Popular Products





## 13-Checkout-Page:

The screenshot shows a Stripe checkout page. On the left, there's a list of items with their prices. The total amount is \$19,300.00. On the right, there's a payment form with fields for email, card information, cardholder name, and country. A green button at the top right says 'Pay with link'. Below it, there's a section for 'Or pay with card'.

checkout.stripe.com/c/pay/cs\_test\_b1CrJ5ERZpNBpx3wjS9KbQDvwUPrHWoSnPXBc2zkq0rp4qwUermkRgolgo#fidkdWxOYHwnPyd1blpxY...

Pay with link

Pay Default sandbox

**\$19,300.00**

Ivory Charm	\$2,800.00
Gray Elegance	\$5,000.00
Nordic Spin	\$3,000.00
Scandi Dip Set	\$4,000.00

Show all 5 items ▾

Or pay with card

Email

Card information

1234 1234 1234 1234

MM / YY CVC

Cardholder name

Full name on card

Country or region

Pakistan ▾

☐ Securely save my information for 1-click checkout  
Pay faster on Default sandbox and everywhere Link is accepted.

## Code Snippet:

EXPLORER

src > app > Checkout > page.tsx > ...

OPEN EDITORS

FURNITUREBAZAAR

src

app

api

shipping

cart

Checkout

page.tsx

contactus

context

FAQ

fonts

OurProducts

payment

privacyPolicy

return-policy

shipping-information

shop

studio

success

TermsAndConditions

OUTLINE

TIMELINE

APPLICATION BUILDER

```
1
2 "use client";
3 import { loadStripe } from "@stripe/stripe-js";
4 import { useEffect, useState } from "react";
5
6 const stripePromise = loadStripe(process.env.NEXT_PUBLIC_STRIPE_PUBLISHABLE_KEY!);
7
8 const Checkout = () => {
9   const [cart, setCart] = useState([]);
10
11   useEffect(() => {
12     const storedCart = localStorage.getItem("cartItems");
13     if (storedCart) {
14       setCart(JSON.parse(storedCart));
15     }
16   }, []);
17
18   const handleCheckout = async () => {
19     const stripe = await stripePromise;
20     const response = await fetch("/api/checkout_session", {
21       method: "POST",
22       headers: { "Content-Type": "application/json" },
23       body: JSON.stringify(cart),
24     });
25     const session = await response.json();
26     stripe?.redirectToCheckout({ sessionId: session.id });
27   };

```

main\* 0.0.0 AWS Ln 1 Col 1 Spaces 2 UTF-8 CRLF TypeScript JSX @ Go