Hackathon-3

Prepared by Aeeba Awan

Roll No : 00263538

Mastering Dynamic Frontend Components
ENHANCE YOUR MARKETPLACE EXPERIENCE - DAY 4

Day 4 - Building Dynamic Frontend Components for Comforty

Overview

On Day 4, I focused on developing dynamic components for the **Comforty** *website, a platform for online furniture sales. The primary goal of this day's work was to enhance the user experience by building dynamic product listings that include categories such as chairs, stools, and sofas. Key features like real-time data fetching, smooth pagination, and category filtering were implemented to make navigation effortless and intuitive. Furthermore, I ensured that the design was responsive, guaranteeing an excellent shopping experience on both desktop and mobile devices.*

Objective:

- *To create a dynamic and interactive frontend for an e-commerce marketplace.*
- *To implement pagination, category filters, and responsive design features for a seamless user experience.*
- *To develop a product listing page with real-time data fetching and smooth pagination.*
- *To enable dynamic product segmentation using category filters.*

- *To ensure the website layout adapts perfectly across various device types.*

Key Features Delivered

1. ***Dynamic Product Listing:*** *The heart of the website's shopping experience is the* ***product listing page****, which displays items like chairs, stools, and sofas in an engaging and efficient manner. This page is designed to dynamically fetch product data in real-time, ensuring that the information presented to the users is always current. No more manual updates—each time the page loads, the latest data is fetched automatically.*

    - ***Product Details:*** *The listing page dynamically pulls and displays important product details such as the title, price, and images from the backend, ensuring customers get the most accurate and up-to-date information.*

2. ***Responsive Design:*** *Recognizing the diverse range of devices used for online shopping, I focused on creating a fully* ***responsive design*** *using* ***Tailwind CSS****. The website's layout dynamically adapts based on screen size—whether it's a large desktop monitor, a tablet, or a mobile phone. This ensures that users get the best possible experience no matter what device they're using. Tailwind's utility classes made it easier to tweak and optimize the design for various screen sizes.*

3. ***Category Filters:*** *Users can now filter products based on different categories such as* ***Featured****,* ***New Arrivals****, and product types like* ***Men's Fashion*** *and* ***Electronics****. This feature improves product discovery and allows customers to narrow down their search according to specific preferences. The filters update dynamically through* ***category-based APIs****, allowing for a seamless and refined browsing experience.*
4. ***Product Gallery Grid Layout:*** *The* ***product gallery*** *is organized into a* ***grid layout****, which offers an intuitive way for customers to browse products. Each product is displayed in its own card, containing essential information like product images, titles, and prices. To ensure an appealing visual design, I used* ***CSS Grid*** *and* ***Flexbox****—two powerful layout techniques that automatically adjust product card placements based on the screen size and content requirements.*

5. ***Pagination:*** *To improve both user experience and site performance,* ***pagination*** *was implemented. Pagination limits the number of products shown per page, which makes the page load faster and reduces potential lag. Additionally, customers can easily navigate between pages to browse more products, without feeling overwhelmed by long lists. This is especially helpful for large inventories, making it possible for users to browse through the full catalog in manageable chunks.*

6. ***Dynamic Product Pages:*** *Each product has its own dedicated page that is dynamically generated. I used* ***Next.js*** *to fetch detailed information about each product, such as title, price, description, and images, from the backend (Sanity CMS in this case). The*

unique identifier (ID) for each product ensures that the right data is displayed every time a customer clicks on a product.

Technical Details and Key Insights

1. Real-Time Data Fetching:

One of the most important skills I developed during this project was working with ***real-time data fetching****. This was a critical aspect of ensuring that the product listings were always up-to-date. Instead of relying on static content, I used dynamic APIs to pull in data in real-time. This also involved setting up routing in* ***Next.js*** *to allow each product to have its own unique page, with its data fetched based on the product's ID.*

2. Improved User Experience Through Filters:

By adding category filter functionality, I learned how to significantly enhance user experience. These features allow customers to refine their product search with ease, leading to quicker results. This is particularly important for an e-commerce website, where product catalogs can be extensive, and customers may need help narrowing down their choices.

3. Optimizing Performance with Lazy Loading:

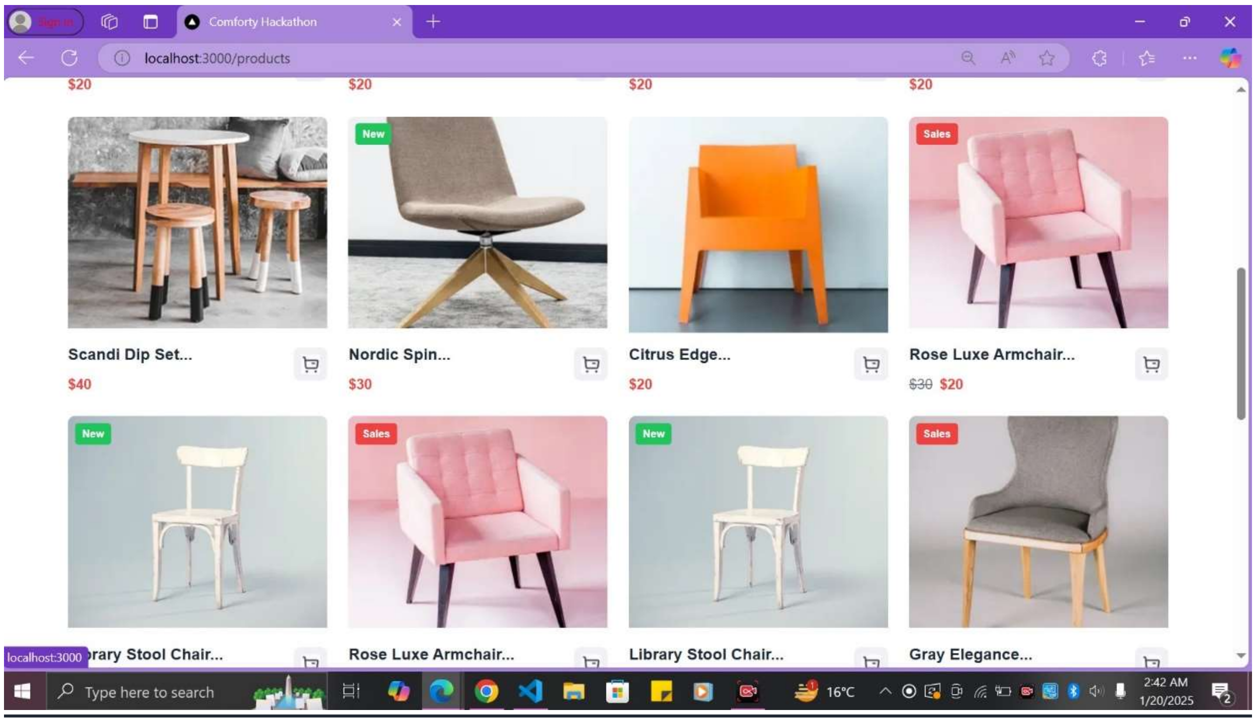
Implementing ***lazy loading*** *and* ***pagination*** *helped improve the site's performance by only loading the necessary products on each page, ensuring faster page load times. This is crucial for maintaining a high-performing site, especially as the product catalog grows over time. Lazy loading ensures that images and other content load only when they are needed, rather than all at once, which is particularly useful for mobile users on slower connections.*

4. Modular Frontend Components:

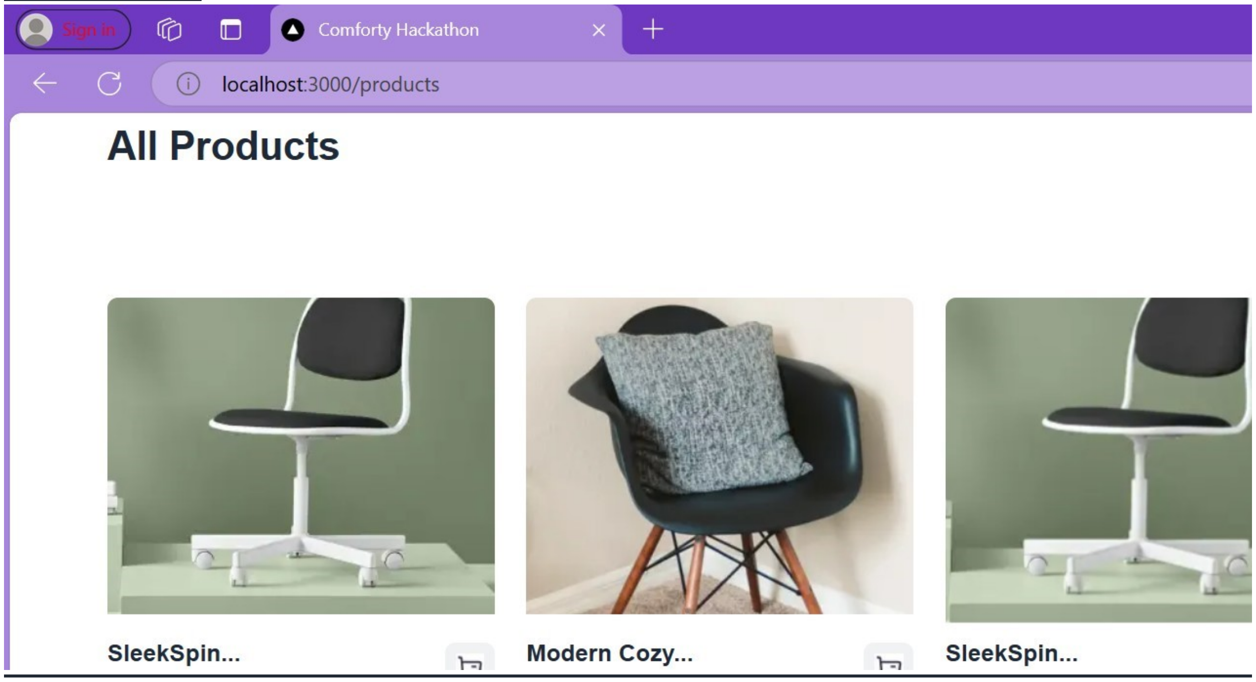
One of the primary design goals was to make the components modular and reusable. This approach not only simplifies the development process but also ensures that the code remains maintainable and scalable for future updates. The modular nature of the frontend allows for easy additions such as new categories, product types, or promotional banners without disrupting the existing structure.

Screenshots of My work Implemenation:

Display Data from Sanity CMS or APIS:



Products 2:



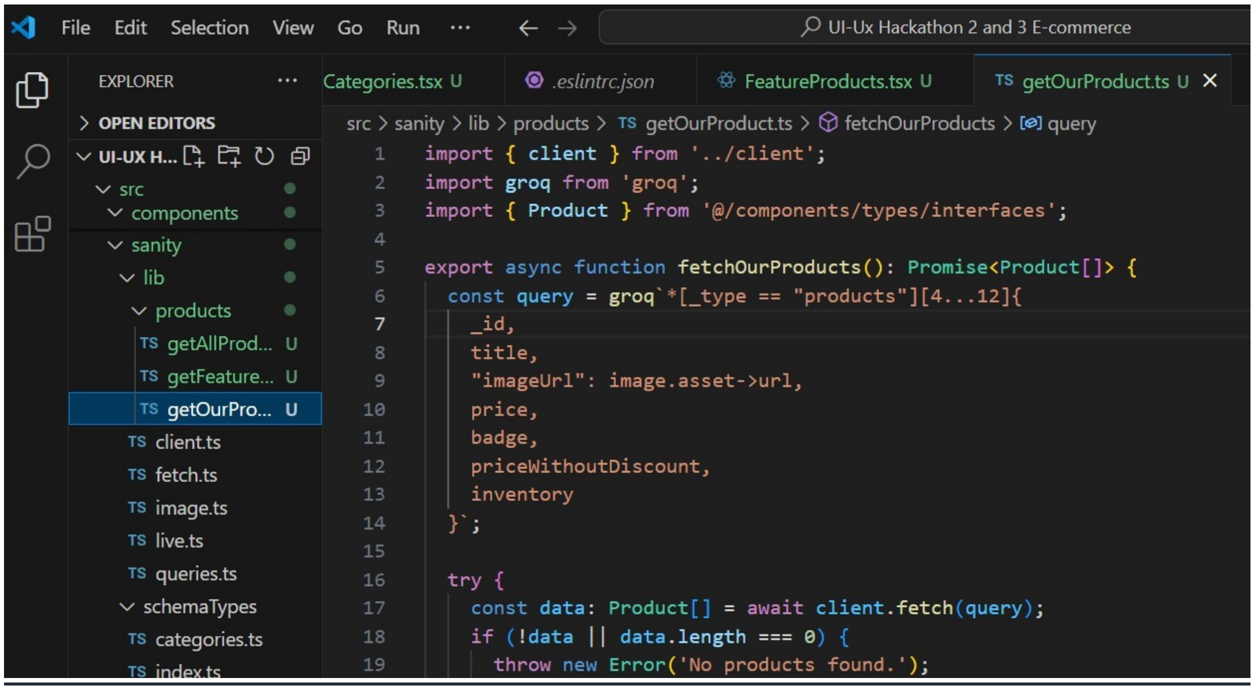
All Products

SleekSpin...

Modern Cozy...

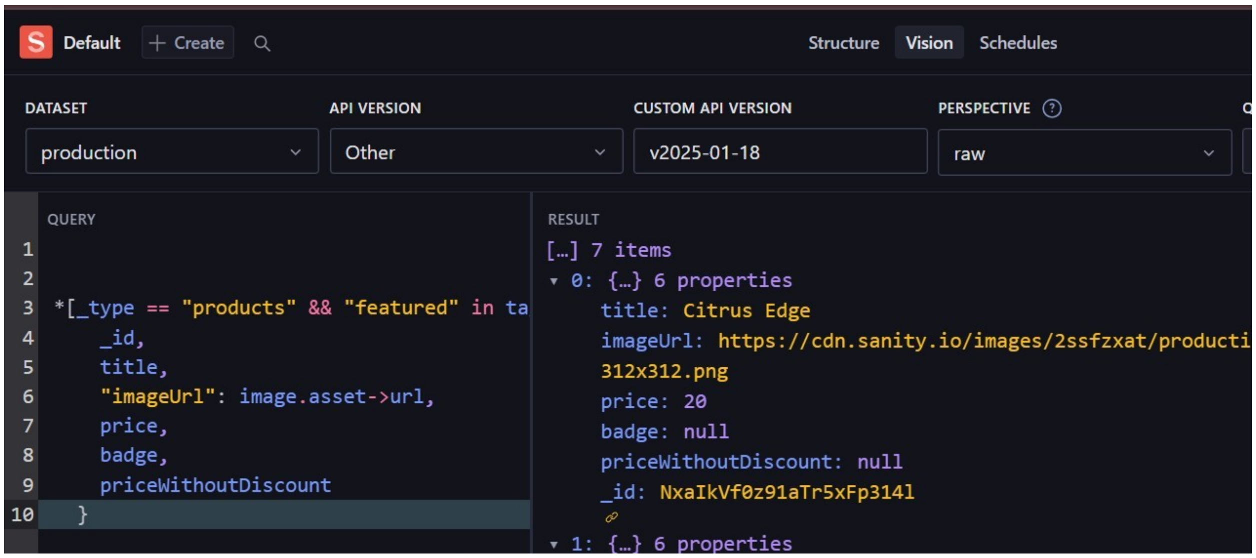
SleekSpin...

Code of fetching products:



```
import { client } from '../client';
import groq from 'groq';
import { Product } from '@/components/types/interfaces';

export async function fetchOurProducts(): Promise<Product[]> {
  const query = groq`*[_type == "products"][4...12]{
    _id,
    title,
    "imageUrl": image.asset->url,
    price,
    badge,
    priceWithoutDiscount,
    inventory
  }`;

  try {
    const data: Product[] = await client.fetch(query);
    if (!data || data.length === 0) {
      throw new Error('No products found.');
```

Data Migration on sanity:



DATASET production
API VERSION Other
CUSTOM API VERSION v2025-01-18
PERSPECTIVE raw

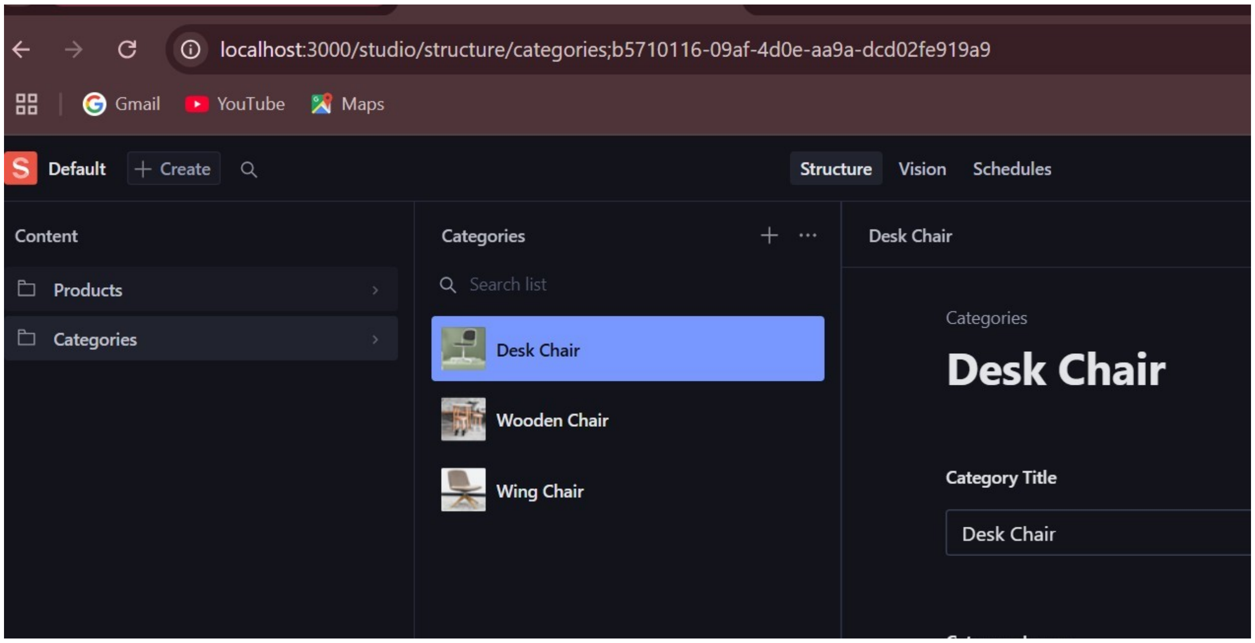
QUERY

```

*[_type == "products" && "featured" in ta
    _id,
    title,
    "imageUrl": image.asset->url,
    price,
    badge,
    priceWithoutDiscount
  }
```

RESULT

```
[…] 7 items
▾ 0: {…} 6 properties
    title: Citrus Edge
    imageUrl: https://cdn.sanity.io/images/2ssfzxat/producti
    312x312.png
    price: 20
    badge: null
    priceWithoutDiscount: null
    _id: NxaIkVf0z91aTr5xFp3141
▾ 1: {…} 6 properties
```

Products Fetch on Sanity:



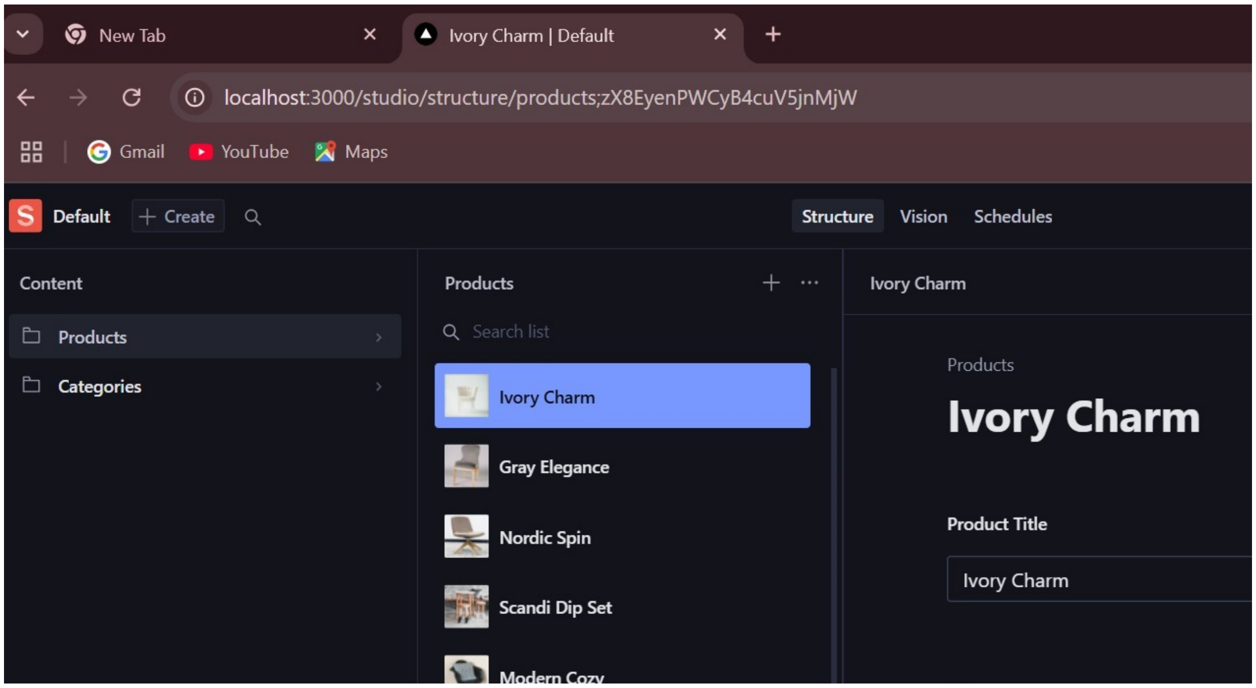
Categories

Desk Chair

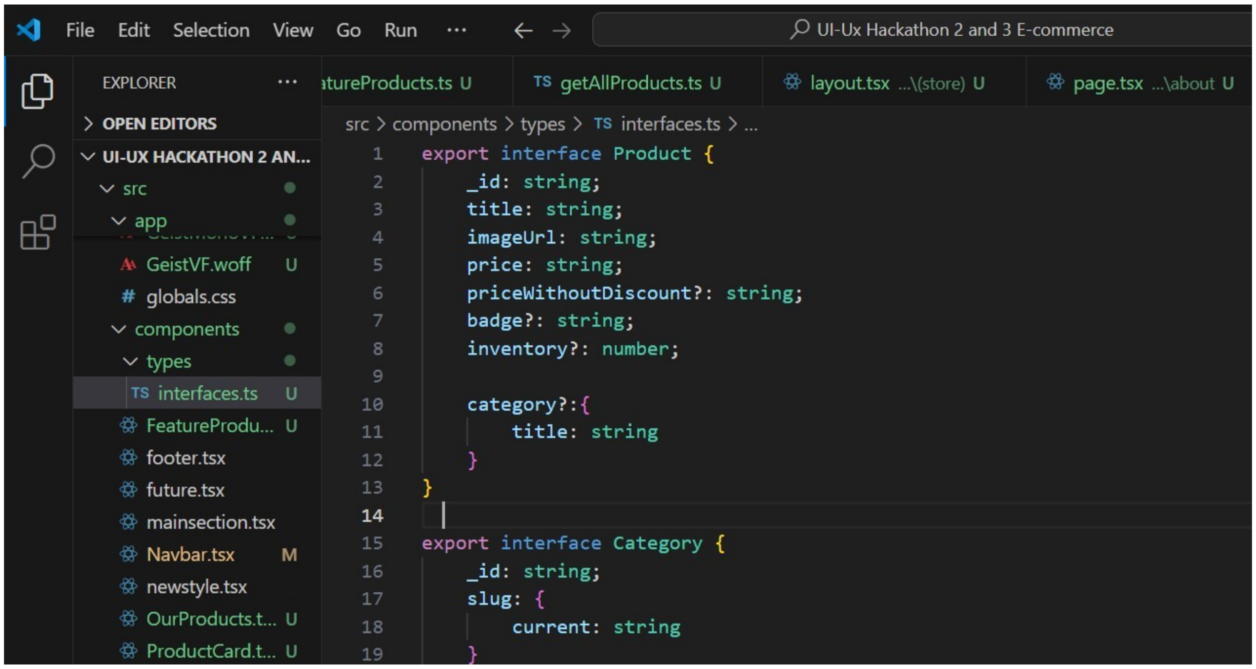
Category Title

Desk Chair

Categories Data Fetch On sanity:

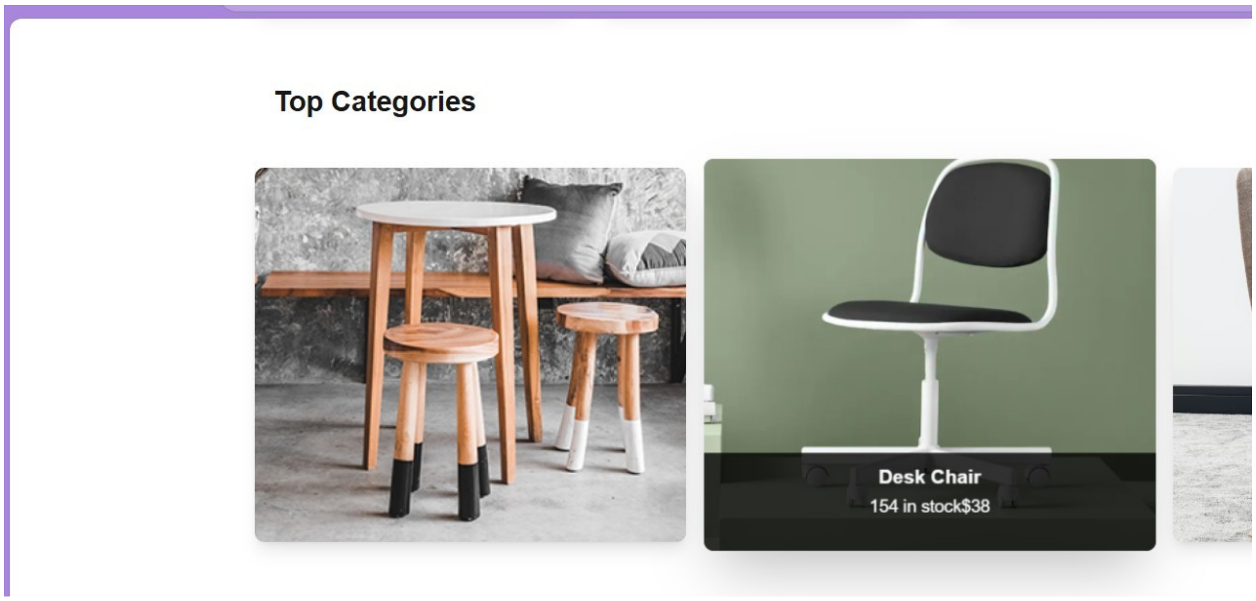


Interfaces of data :



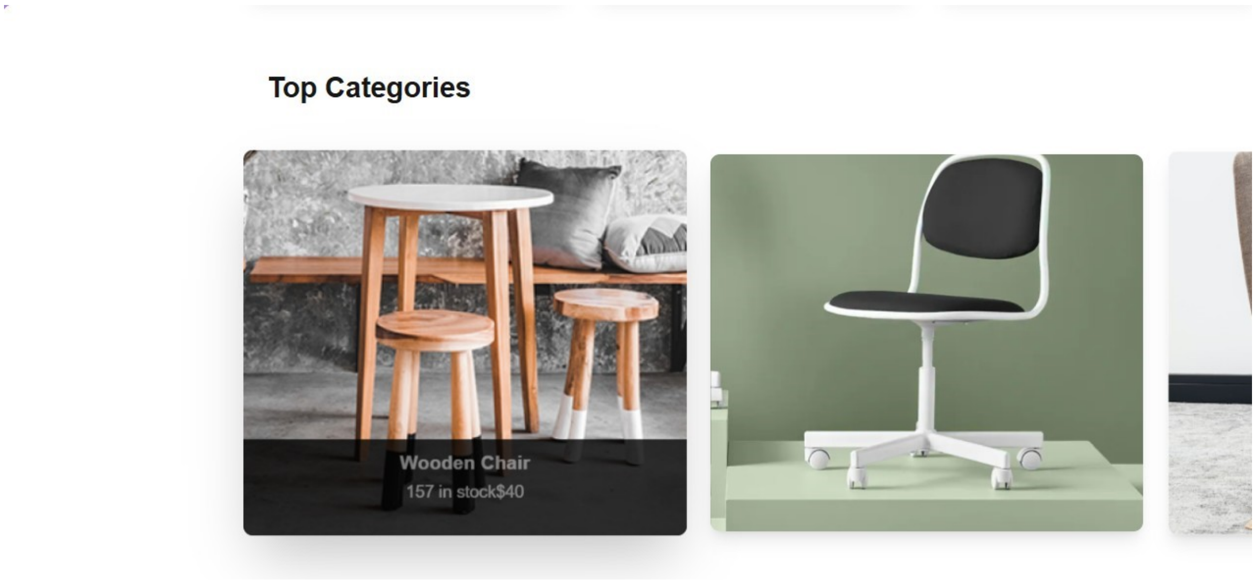
```typescript
export interface Product {
    _id: string;
    title: string;
    imageUrl: string;
    price: string;
    priceWithoutDiscount?: string;
    badge?: string;
    inventory?: number;

    category?:{
        title: string
    }
}

export interface Category {
    _id: string;
    slug: {
        current: string
    }
```

Categories Component:



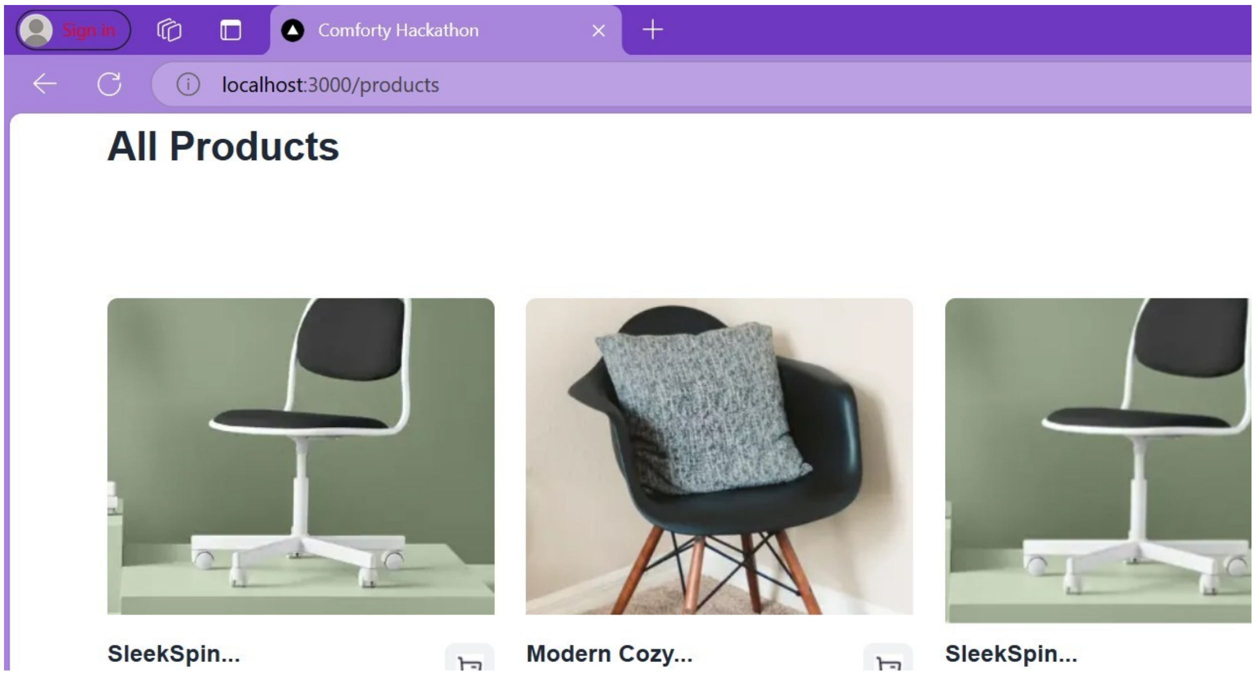
Top Categories

Categories 2:



Top Categories

All Products :



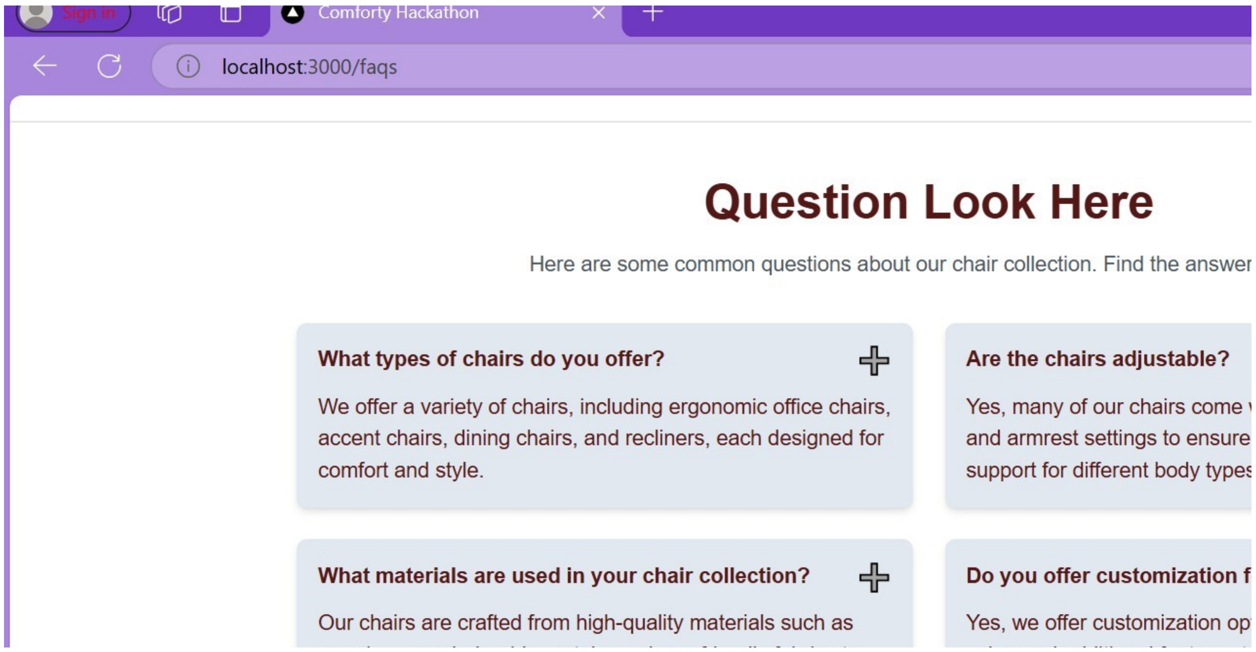
All Products

SleekSpin...

Modern Cozy...

SleekSpin...

Faqs Component :



Question Look Here

Here are some common questions about our chair collection. Find the answer

What types of chairs do you offer?

We offer a variety of chairs, including ergonomic office chairs, accent chairs, dining chairs, and recliners, each designed for comfort and style.

Are the chairs adjustable?

Yes, many of our chairs come
and armrest settings to ensure
support for different body types

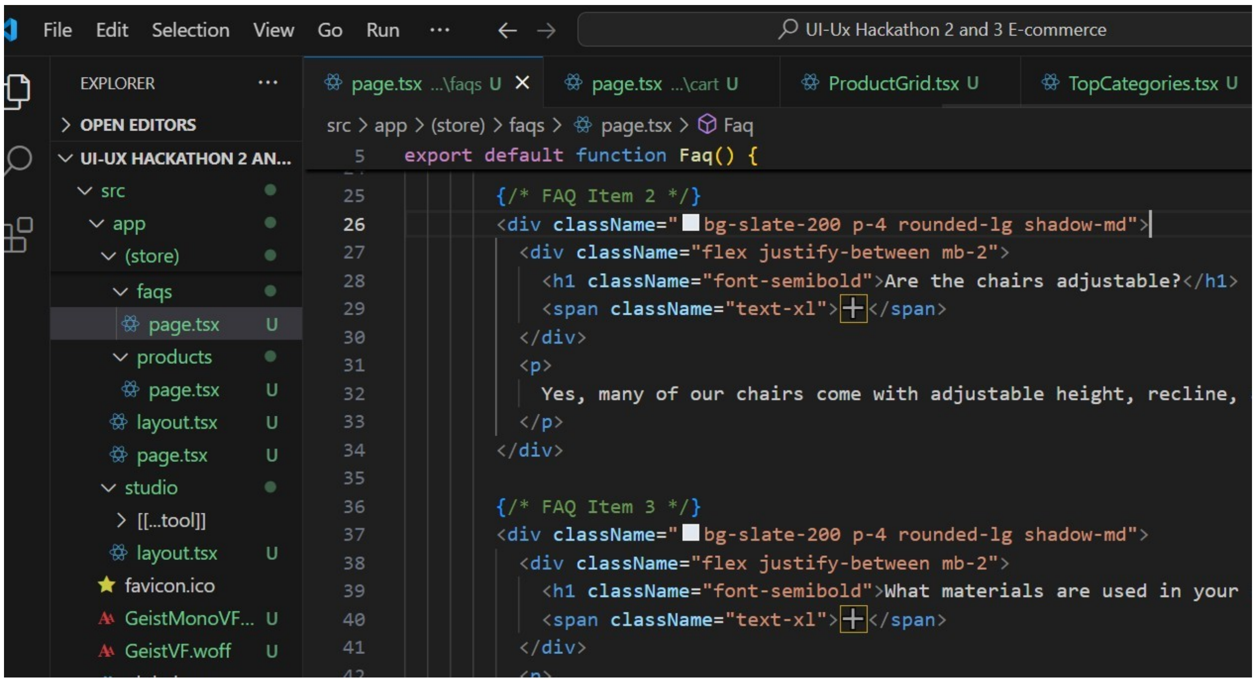
What materials are used in your chair collection?

Our chairs are crafted from high-quality materials such as

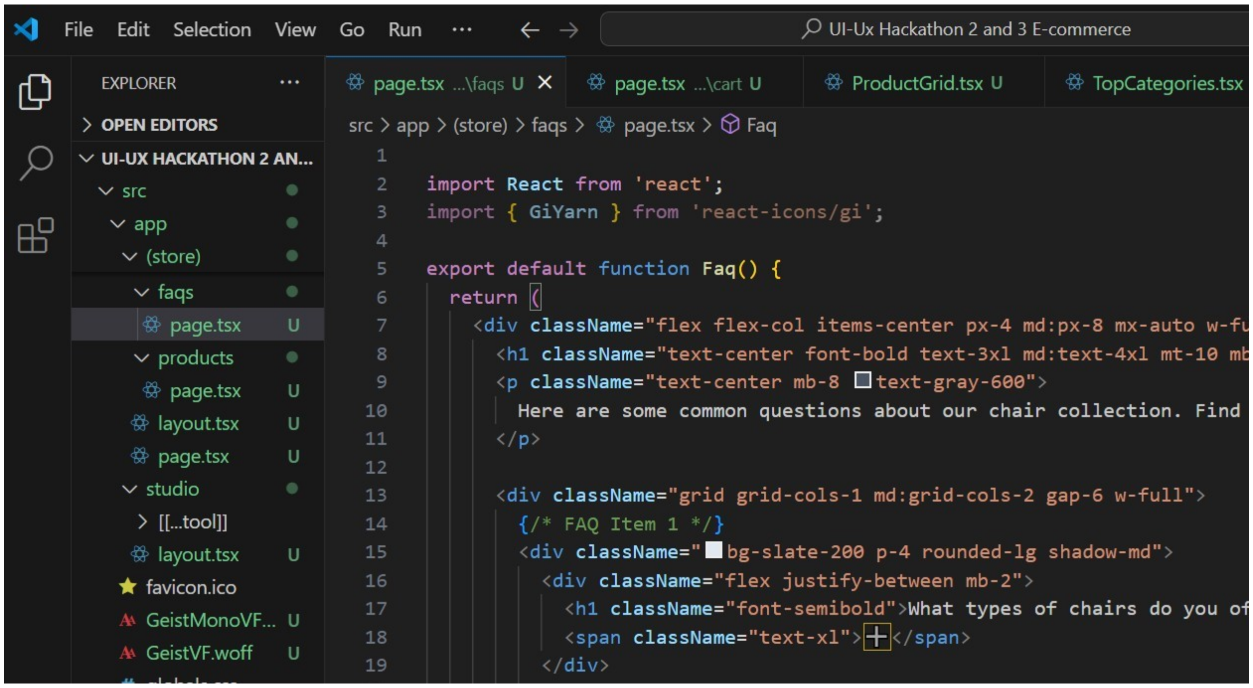
Do you offer customization f

Yes, we offer customization op

Faq's Code :

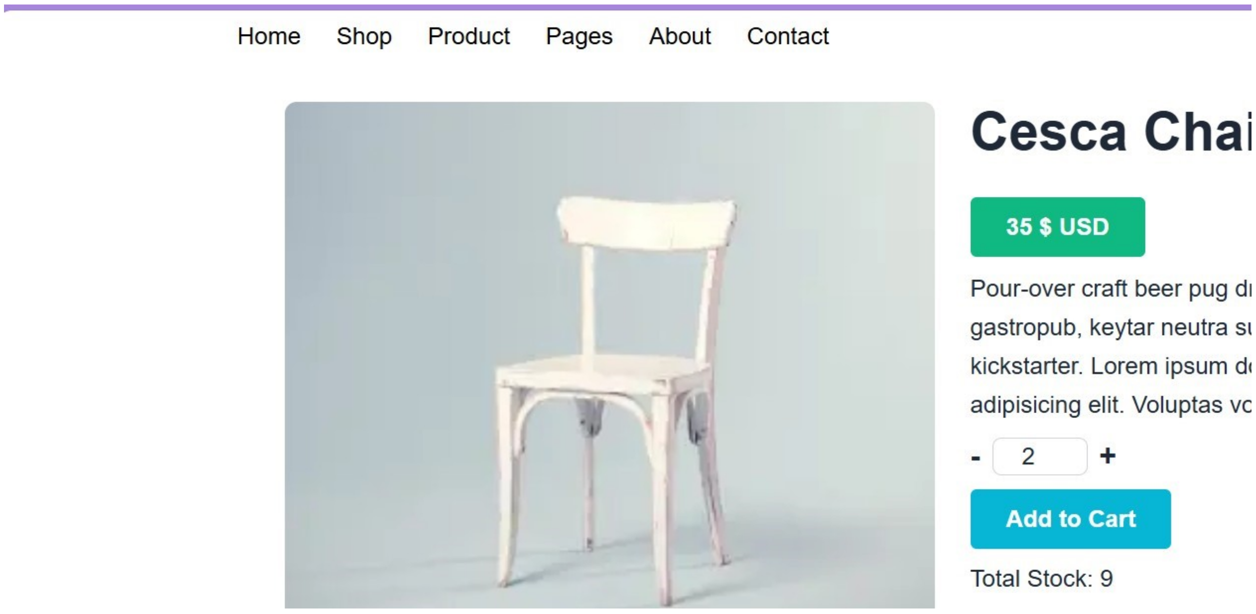


```
export default function Faq() {
        {/* FAQ Item 2 */}
        <div className=" bg-slate-200 p-4 rounded-lg shadow-md">
          <div className="flex justify-between mb-2">
            <h1 className="font-semibold">Are the chairs adjustable?</h1>
            <span className="text-xl">+</span>
          </div>
          <p>
            Yes, many of our chairs come with adjustable height, recline,
          </p>
        </div>

        {/* FAQ Item 3 */}
        <div className=" bg-slate-200 p-4 rounded-lg shadow-md">
          <div className="flex justify-between mb-2">
            <h1 className="font-semibold">What materials are used in your
            <span className="text-xl">+</span>
          </div>
```



```

import React from 'react';
import { GiYarn } from 'react-icons/gi';

export default function Faq() {
  return (
    <div className="flex flex-col items-center px-4 md:px-8 mx-auto w-fu
      <h1 className="text-center font-bold text-3xl md:text-4xl mt-10 mb
      <p className="text-center mb-8 text-gray-600">
        Here are some common questions about our chair collection. Find
      </p>

      <div className="grid grid-cols-1 md:grid-cols-2 gap-6 w-full">
        {/* FAQ Item 1 */}
        <div className=" bg-slate-200 p-4 rounded-lg shadow-md">
          <div className="flex justify-between mb-2">
            <h1 className="font-semibold">What types of chairs do you of
            <span className="text-xl">+</span>
          </div>
```

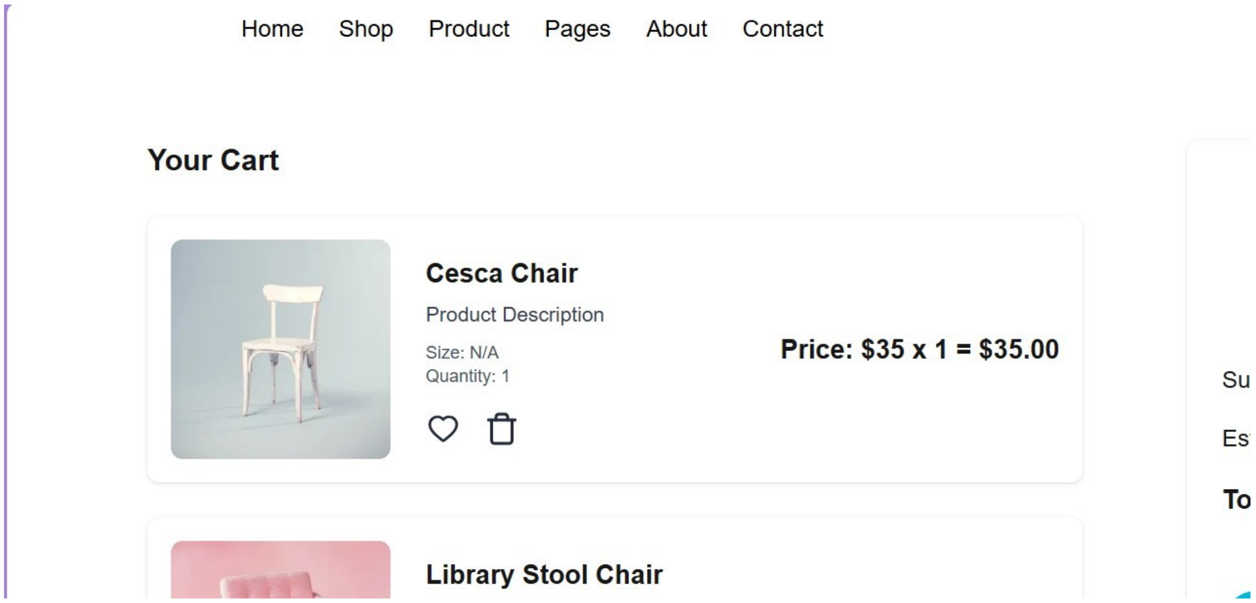
Add to Cart and view all:



Cesca Chai

Pour-over craft beer pug dı
gastropub, keytar neutra sı
kickstarter. Lorem ipsum d
adipisicing elit. Voluptas vo

Total Stock: 9



Your Cart

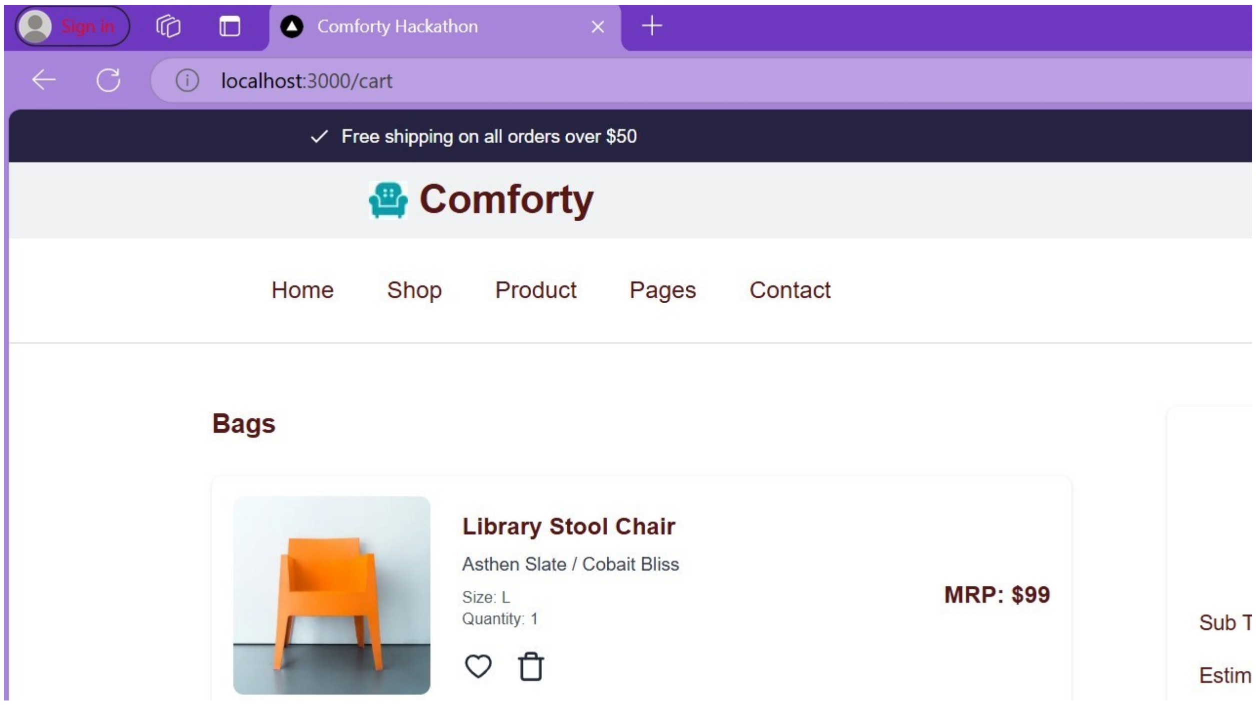
Cesca Chair

Product Description

Size: N/A
Quantity: 1

Price: $35 x 1 = $35.00

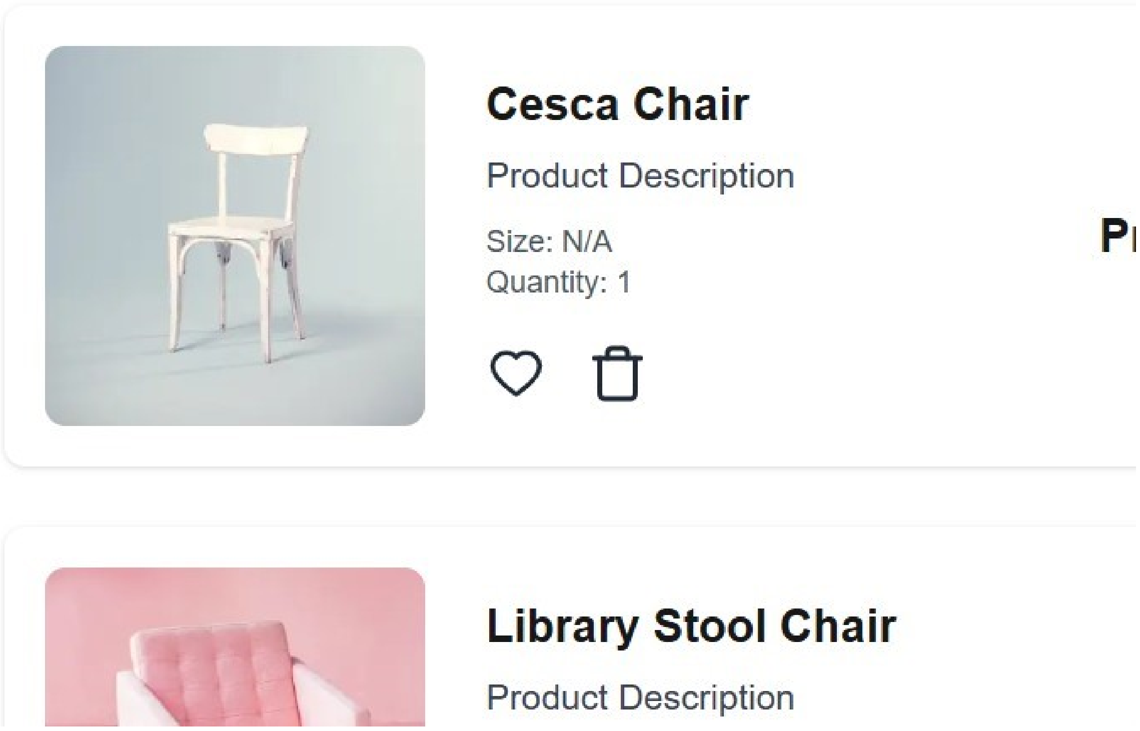
Library Stool Chair



Bags

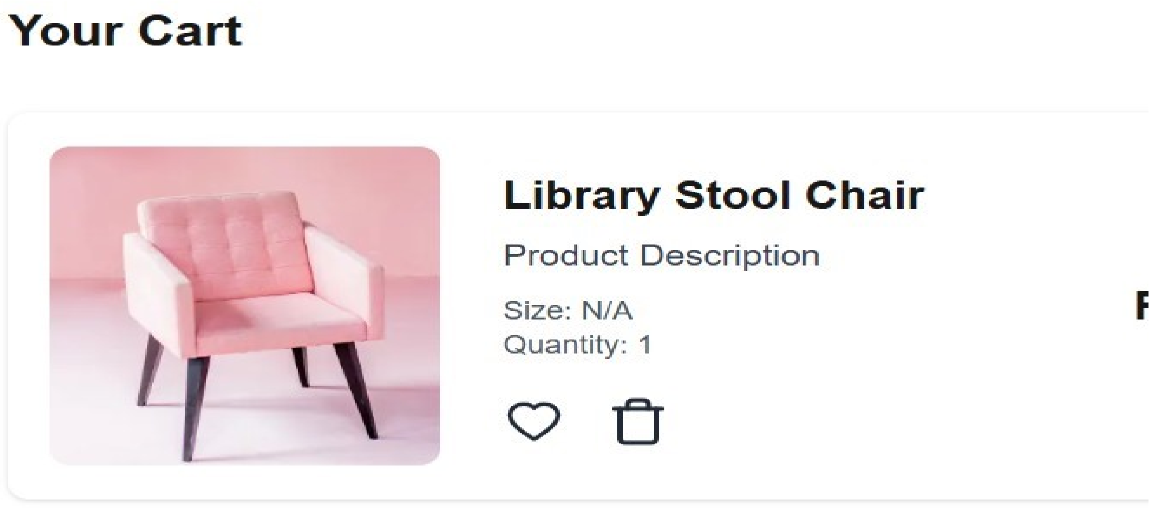
Library Stool Chair
Asthen Slate / Cobait Bliss
Size: L
Quantity: 1
MRP: $99

Saved items:


Cesca Chair
Product Description
Size: N/A
Quantity: 1

Library Stool Chair
Product Description

Deletation of one Item from Cart:


Your Cart

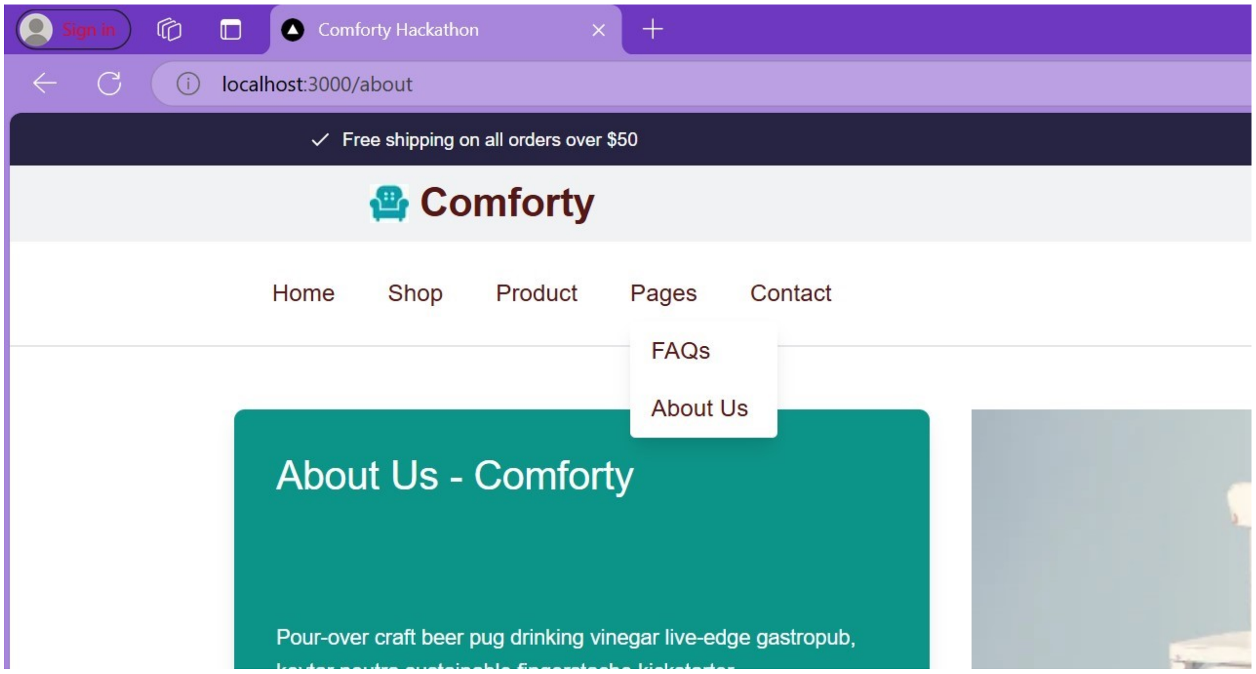
Library Stool Chair

Product Description

Size: N/A

Quantity: 1

Filtration of main components:



Free shipping on all orders over $50

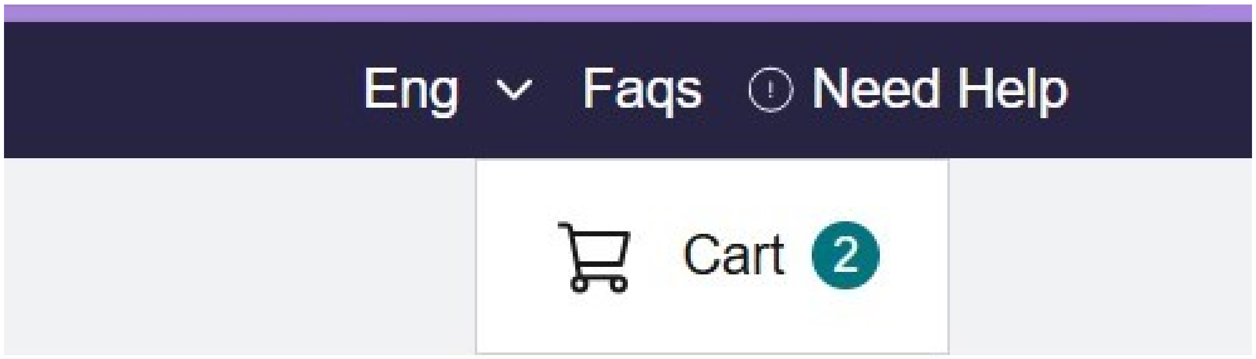
Comforty

About Us - Comforty

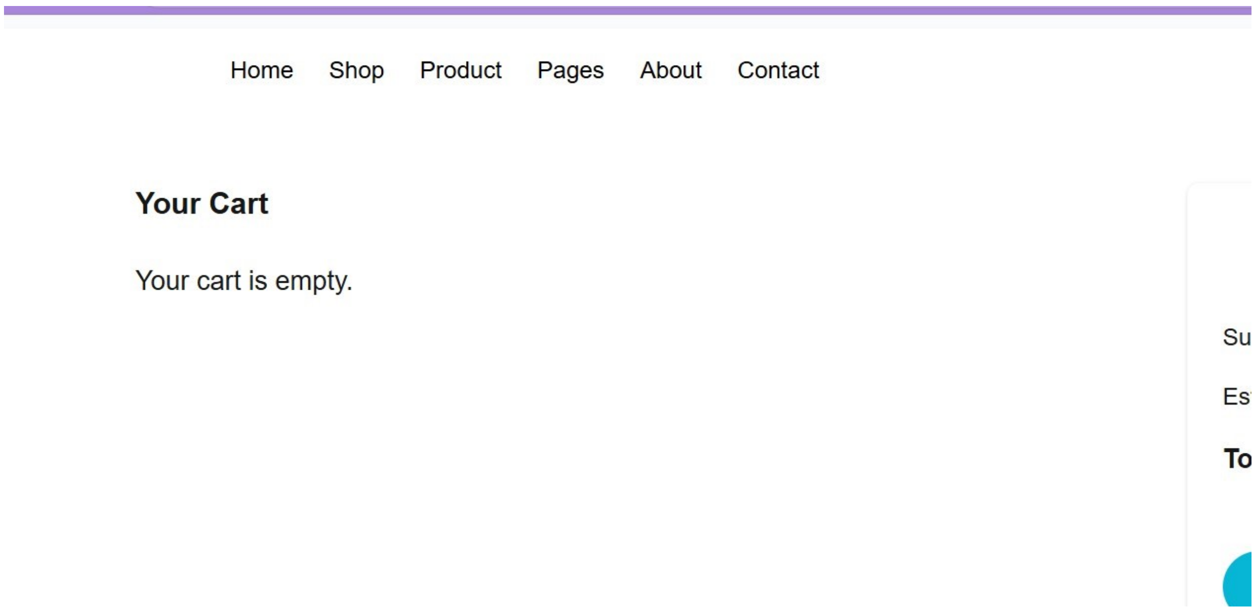
Pour-over craft beer pug drinking vinegar live-edge gastropub,

Social Media Working Icons:

Items number added to cart:



Empty Cart :



Home Shop Product Pages About Contact

Your Cart

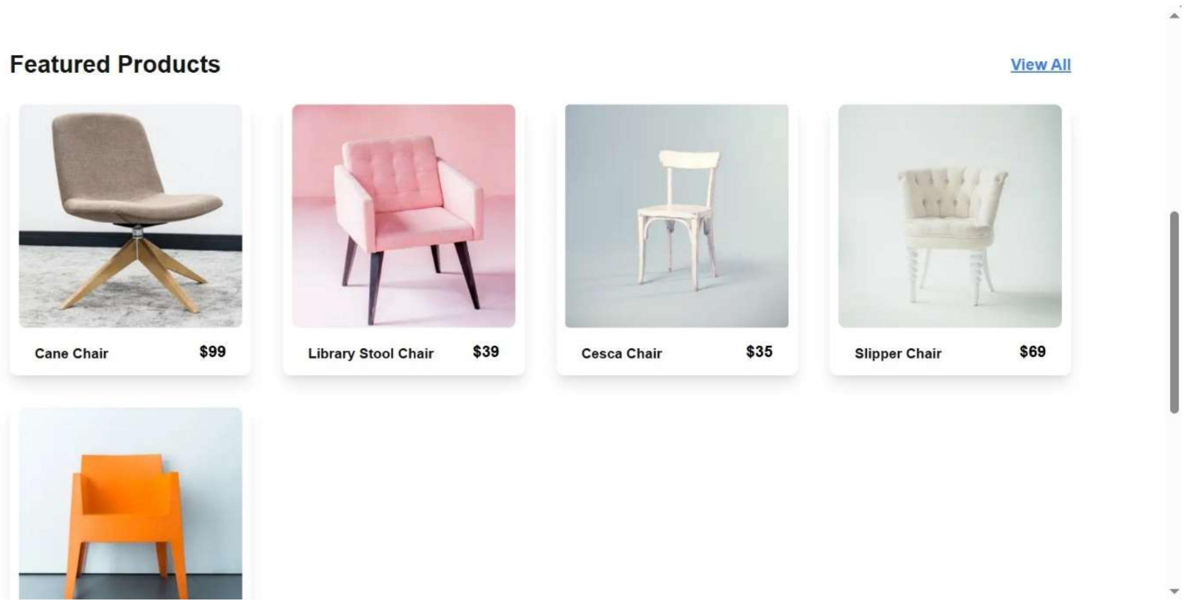
Your cart is empty.

Su

Es

To

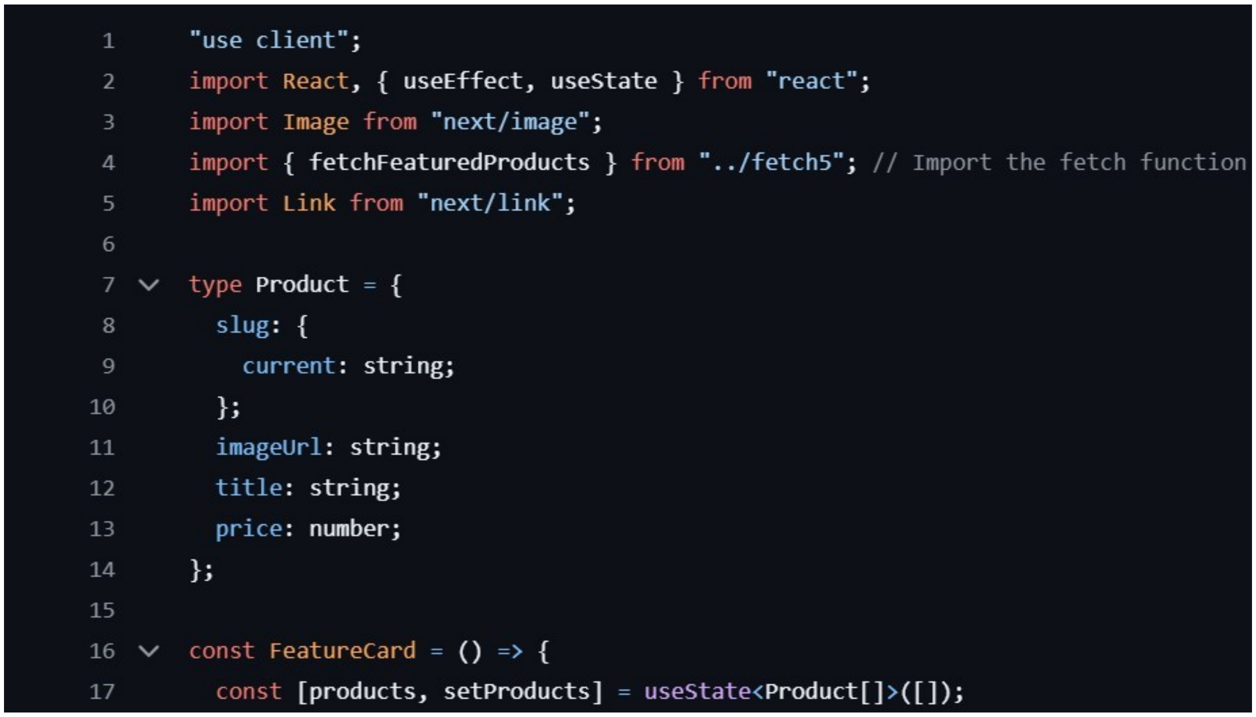
Featured Products:



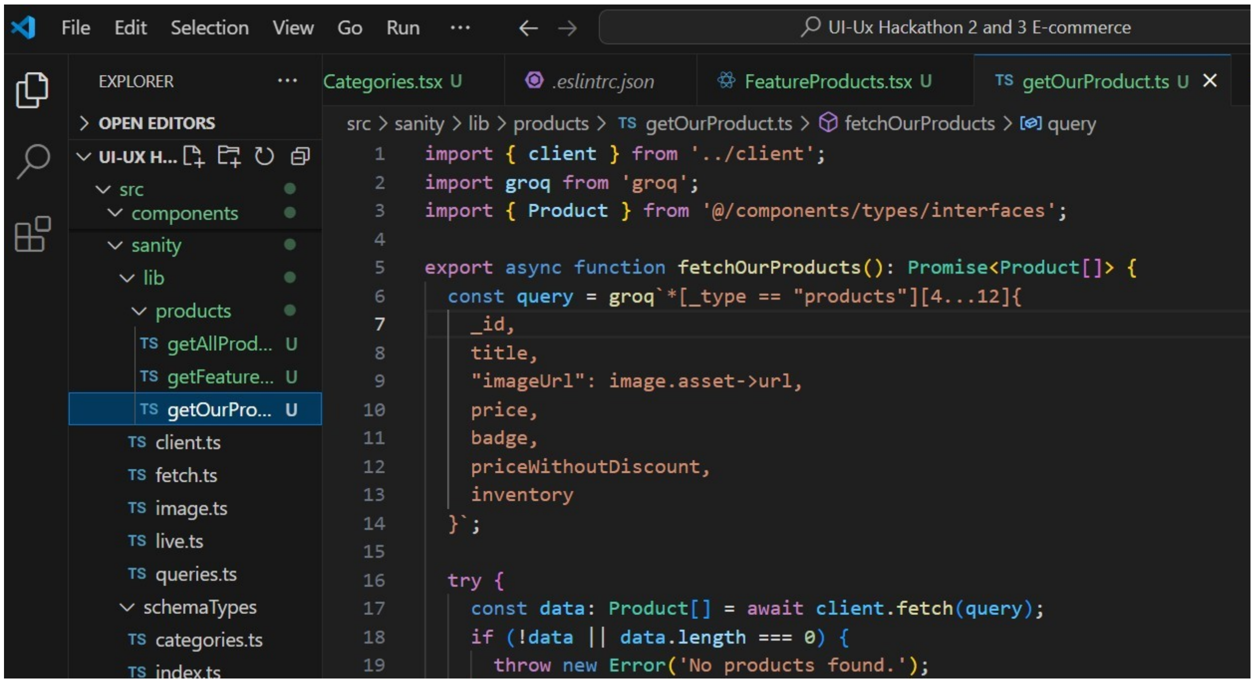
Featured Products View All

Cane Chair	$99
Library Stool Chair	$39
Cesca Chair	$35
Slipper Chair	$69

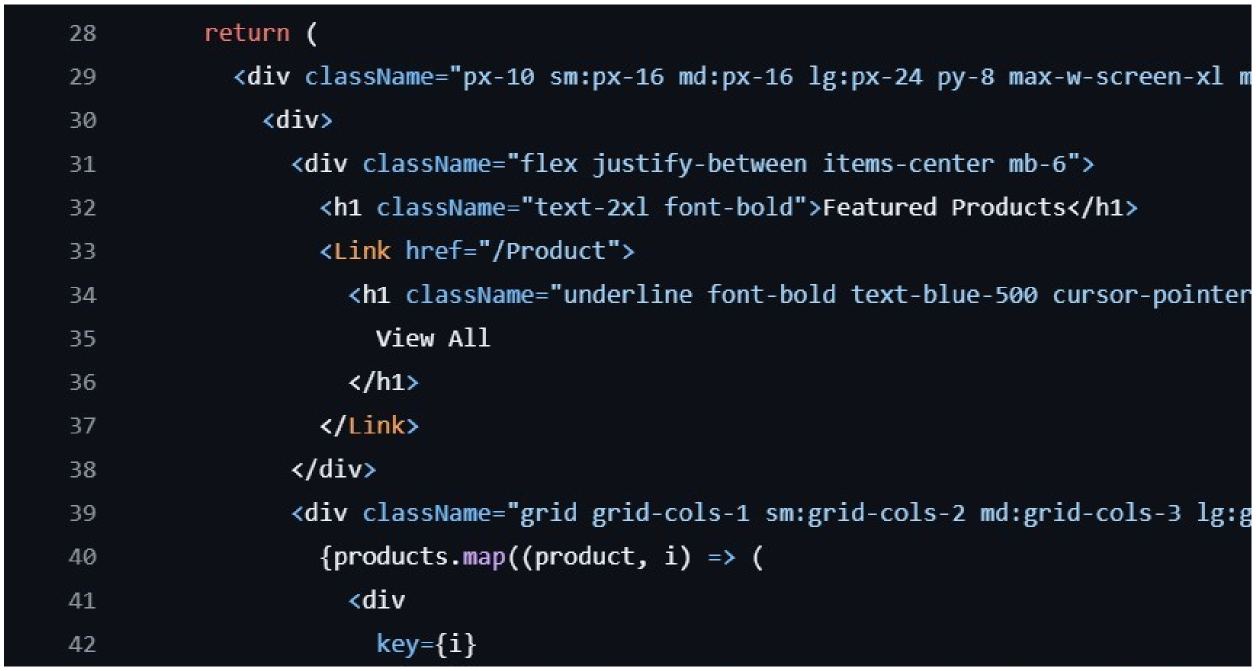
Remaining Code Snippets:



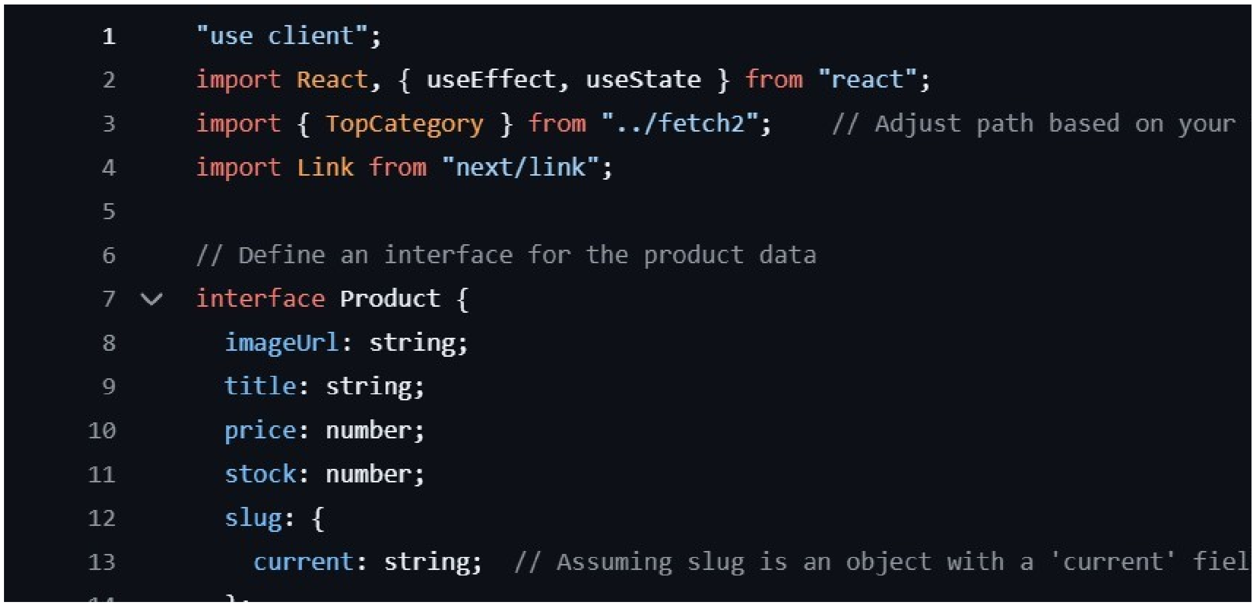
```tsx
"use client";
import React, { useEffect, useState } from "react";
import Image from "next/image";
import { fetchFeaturedProducts } from "../fetch5"; // Import the fetch function
import Link from "next/link";

type Product = {
  slug: {
    current: string;
  };
  imageUrl: string;
  title: string;
  price: number;
};

const FeatureCard = () => {
  const [products, setProducts] = useState<Product[]>([]);
```



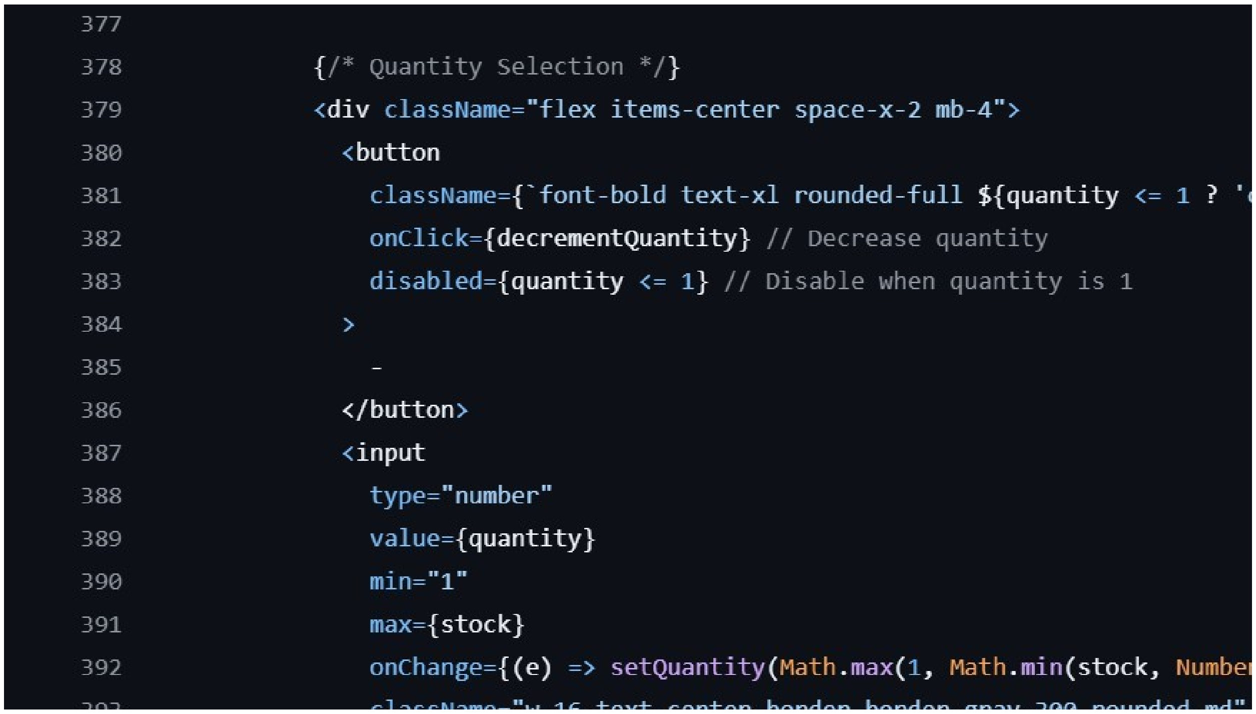
```ts
import { client } from '../client';
import groq from 'groq';
import { Product } from '@/components/types/interfaces';

export async function fetchOurProducts(): Promise<Product[]> {
  const query = groq`*[_type == "products"][4...12]{
    _id,
    title,
    "imageUrl": image.asset->url,
    price,
    badge,
    priceWithoutDiscount,
    inventory
  }`;

  try {
    const data: Product[] = await client.fetch(query);
    if (!data || data.length === 0) {
      throw new Error('No products found.');
```



```
    return (
      <div className="px-10 sm:px-16 md:px-16 lg:px-24 py-8 max-w-screen-xl m
        <div>
          <div className="flex justify-between items-center mb-6">
            <h1 className="text-2xl font-bold">Featured Products</h1>
            <Link href="/Product">
              <h1 className="underline font-bold text-blue-500 cursor-pointer
                View All
              </h1>
            </Link>
          </div>
          <div className="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:g
            {products.map((product, i) => (
              <div
                key={i}
```



```
"use client";
import React, { useEffect, useState } from "react";
import { TopCategory } from "../fetch2";    // Adjust path based on your
import Link from "next/link";

// Define an interface for the product data
interface Product {
  imageUrl: string;
  title: string;
  price: number;
  stock: number;
  slug: {
    current: string;  // Assuming slug is an object with a 'current' fiel
```



```
{/* Quantity Selection */}
<div className="flex items-center space-x-2 mb-4">
  <button
    className={`font-bold text-xl rounded-full ${quantity <= 1 ? '
    onClick={decrementQuantity} // Decrease quantity
    disabled={quantity <= 1} // Disable when quantity is 1
  >
    -
  </button>
  <input
    type="number"
    value={quantity}
    min="1"
    max={stock}
    onChange={(e) => setQuantity(Math.max(1, Math.min(stock, Numbe
    className="w-16 text-center border border-gray-300 rounded-md"
```

Conclusion

*The **Comforty** website now offers a highly dynamic and responsive shopping experience for its customers. The integration of product listings, category filters, pagination, and responsive design ensures a user-friendly interface across all devices. Each feature was designed to meet the needs of modern e-commerce users—whether they're browsing on a desktop at home or shopping on their mobile device while on the go.*

The dynamic product listings and category filters streamline the product discovery process, making it easier for customers to find exactly what they're looking for. The responsive design ensures that the website provides an optimal experience on all screen sizes, while pagination improves both site performance and usability.

*This project taught me valuable skills in **dynamic data handling**, **frontend performance optimization**, and **user experience enhancement**. It was a great opportunity to apply modern web* *development techniques such as* ***Next.js****,* ***Tailwind CSS****, and* ***API integration****.*

Moving forward, the modularity of the components allows the website to scale easily. Future updates, whether it's adding new product categories or launching a seasonal sale, can be accomplished smoothly without major disruptions. The foundation built here ensures that ***Comforty*** *will continue to provide an exceptional e-commerce*

Areeba Awan
[Roll No. 00263538]