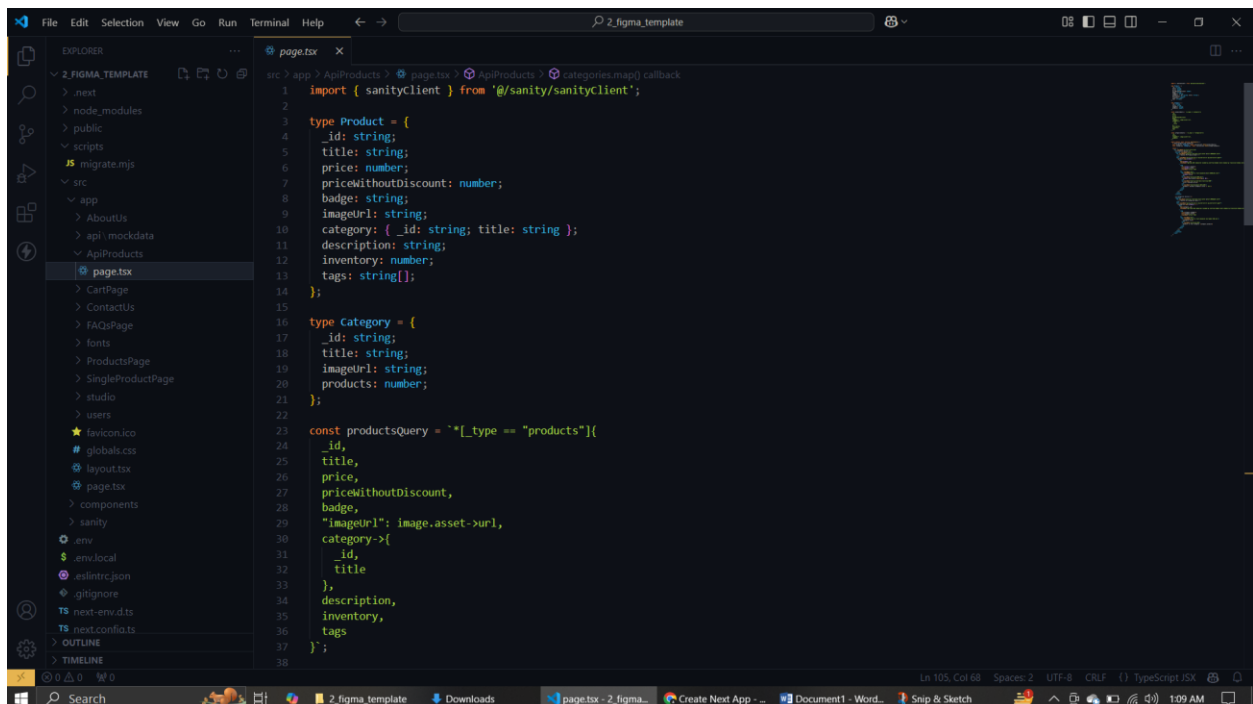
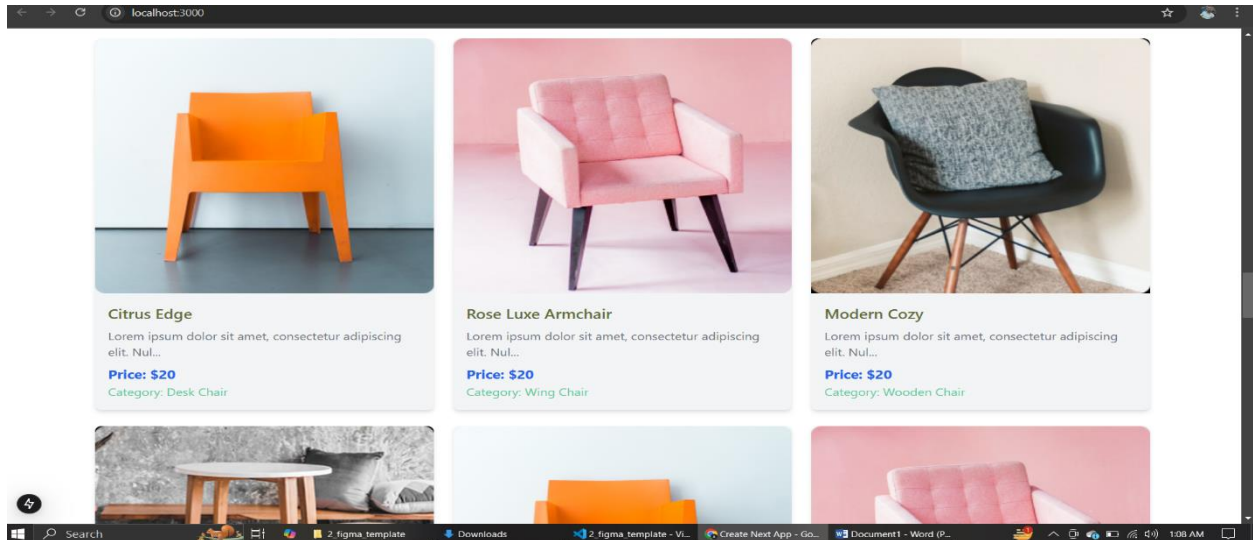


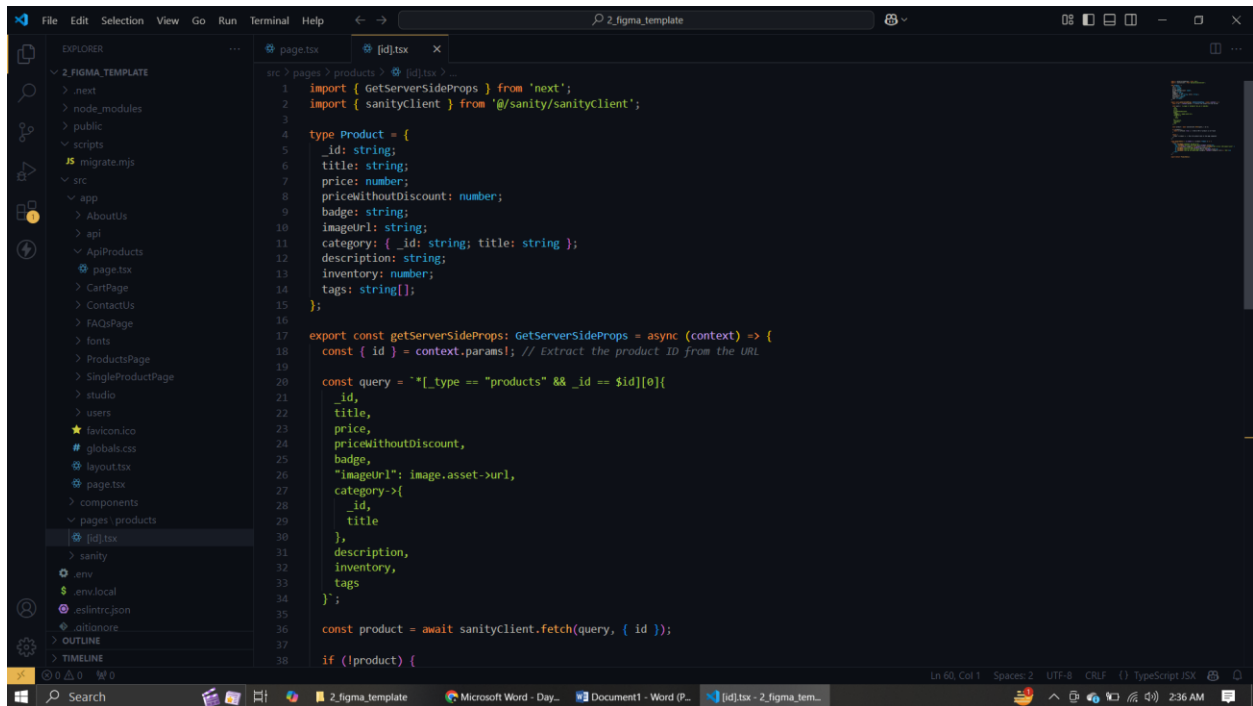
HACKATHON-3 (DAY-4)

BUILDING DYNAMIC FRONTEND COMPONENTS

1.Product Listing Components:

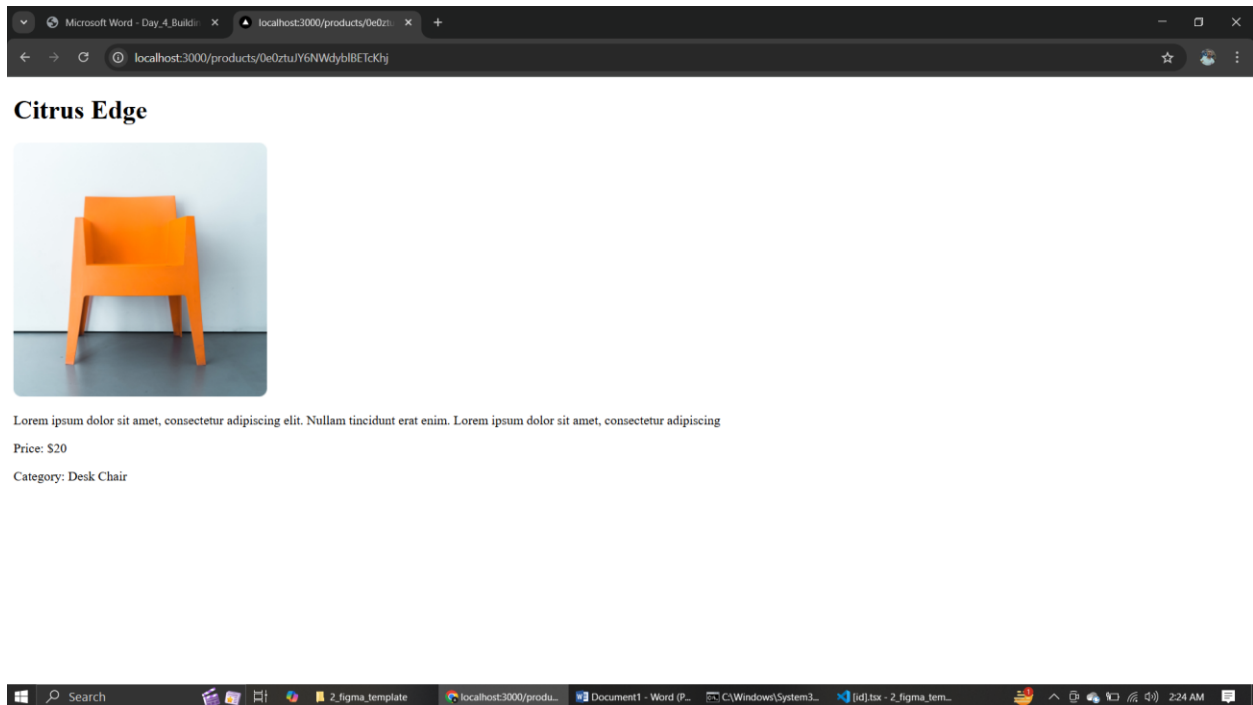


2.Product detail component using dynamic routing:



```
1 import { GetServerSideProps } from 'next';
2 import { sanityClient } from '@sanity/sanityClient';
3
4 type Product = {
5   id: string;
6   title: string;
7   price: number;
8   priceWithoutDiscount: number;
9   badge: string;
10  imageUrl: string;
11  category: { _id: string; title: string };
12  description: string;
13  inventory: number;
14  tags: string[];
15 };
16
17 export const getServerSideProps: GetServerSideProps = async (context) => {
18   const { id } = context.params!; // Extract the product ID from the URL
19
20   const query = `*[_type == "products" && _id == ${id}][0]{
21     _id,
22     title,
23     price,
24     priceWithoutDiscount,
25     badge,
26     "imageUrl": image.asset->url,
27     category->{
28       _id,
29       title
30     },
31     description,
32     inventory,
33     tags
34   }`;
35
36   const product = await sanityClient.fetch(query, { id });
37
38   if (!product) {
```

Product-1



Product-2



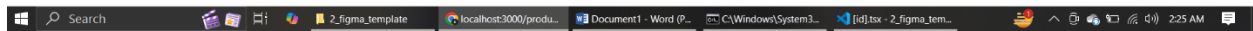
Rose Luxe Armchair



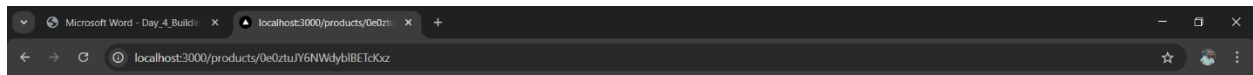
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam tincidunt erat enim. Lorem ipsum dolor sit amet, consectetur adipiscing

Price: \$20

Category: Wing Chair



Product-3



Modern Cozy



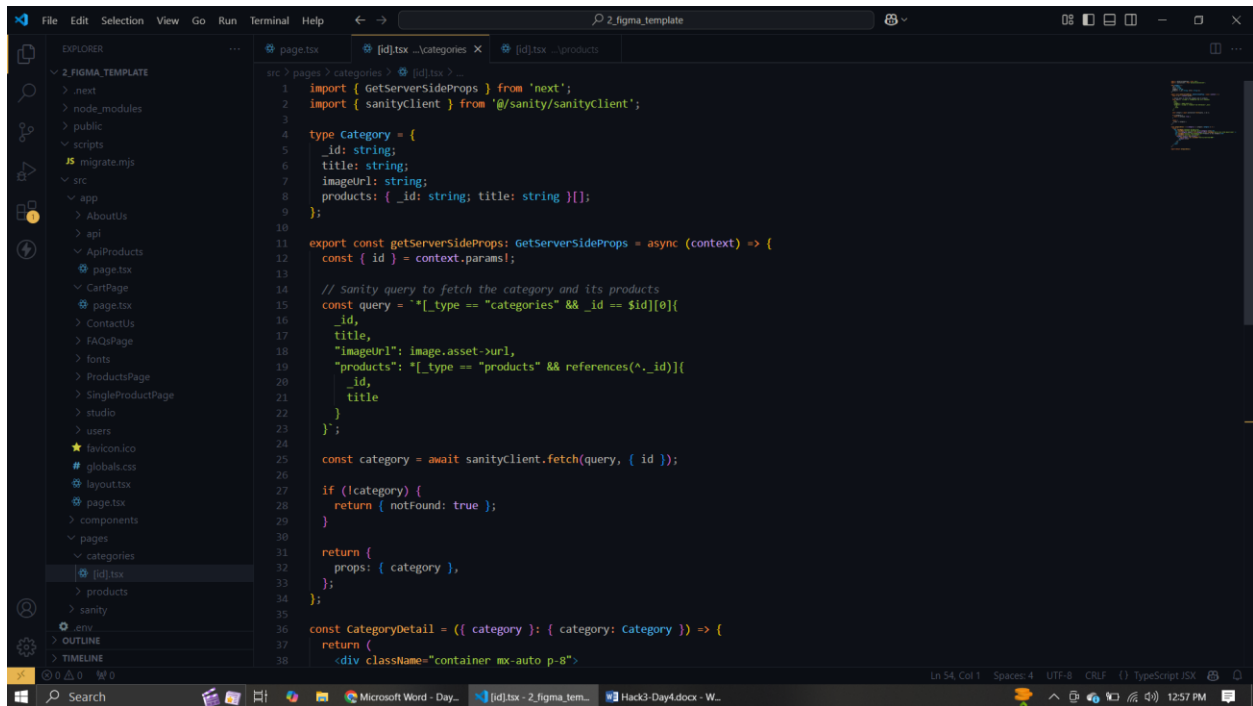
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam tincidunt erat enim. Lorem ipsum dolor sit amet, consectetur adipiscing

Price: \$20

Category: Wooden Chair



3.Category component:



```
1 import { GetServerSideProps } from 'next';
2 import { sanityClient } from '@sanity/sanityClient';
3
4 type Category = {
5   _id: string;
6   title: string;
7   imageUrl: string;
8   products: { _id: string; title: string }[];
9 };
10
11 export const getServerSideProps: GetServerSideProps = async (context) => {
12   const { id } = context.params!;
13
14   // Sanity query to fetch the category and its products
15   const query = `*[ _type == "categories" && _id == $id ][0]{
16     _id,
17     title,
18     "imageUrl": image.asset->url,
19     "products": *[ _type == "products" && references(^._id) ]{
20       _id,
21       title
22     }
23   }`;
24
25   const category = await sanityClient.fetch(query, { id });
26
27   if (!category) {
28     return { notFound: true };
29   }
30
31   return {
32     props: { category },
33   };
34 };
35
36 const CategoryDetail = ({ category }: { category: Category }) => {
37   return (
38     <div className="container mx-auto p-8">
```

-----Sanity API Categories Data-----



Wing Chair

Products in this category:
[View Category](#)



Wooden Chair

Products in this category:
[View Category](#)



Desk Chair

Products in this category:
[View Category](#)

Wooden Chair



Products in this Category:

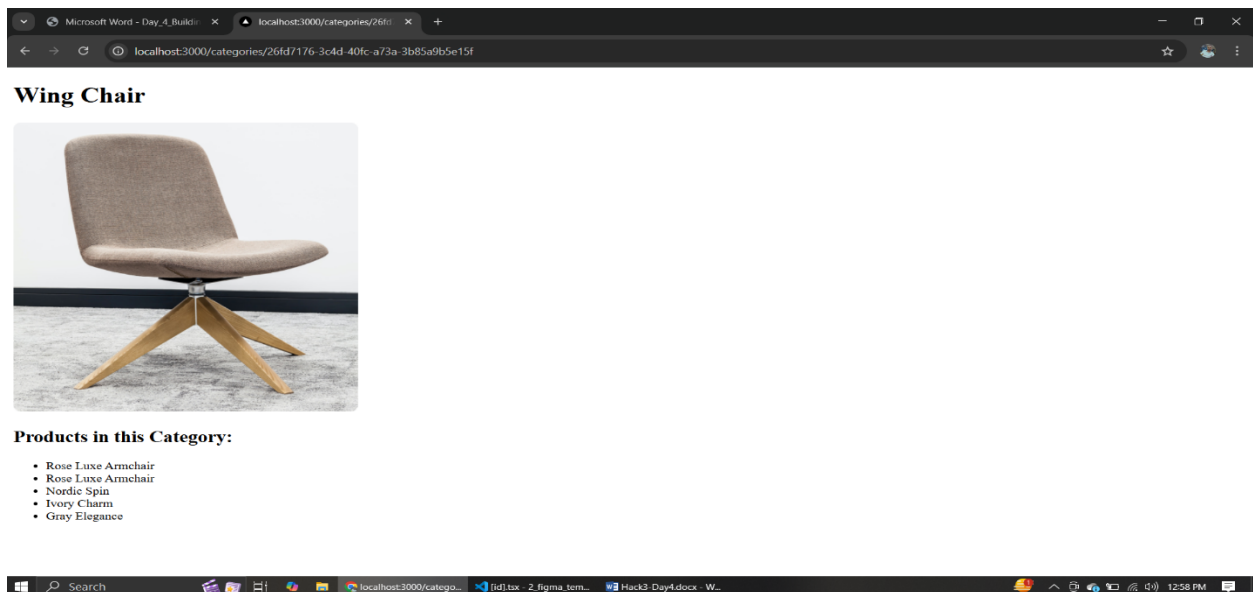
- Modern Cozy
- Scandi Dip Set
- Library Stool Chair
- Scandi Dip Set
- Modern Cozy
- Library Stool Chair

Desk Chair



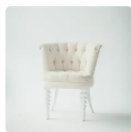
Products in this Category:

- Citrus Edge
- Citrus Edge
- SleekSpin
- SleekSpin



4.Cart Component:

Bag

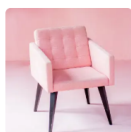


Library Stool Chair

Ashen Slate / Cobalt Bliss
Size: L Quantity: 1



MRP: \$99



Library Stool Chair

Ashen Slate / Cobalt Bliss
Size: L Quantity: 1



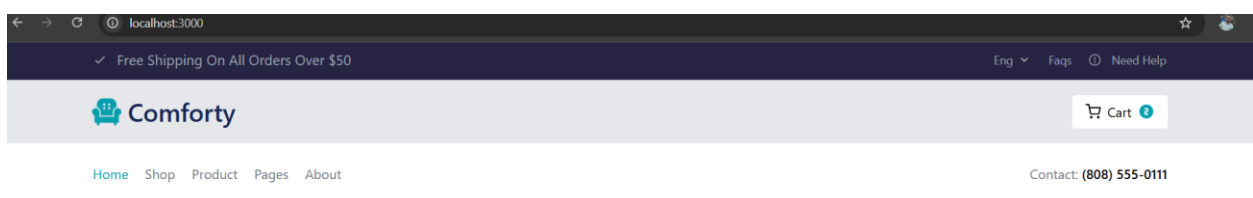
MRP: \$99

Summary

Subtotal	\$198.00
Estimated Delivery & Handling	Free
Total	\$198.00

Proceed to Checkout

5. Header & Footer Components:





Vivamus tristique odio sit amet velit
semper, eu posuere turpis interdum. Cras
egestas purus.



Categories

Sofa
Armchair
Wing Chair
Desk Chair
Wooden
Chair
Park Bench

Support

Help &
Support
Terms &
Conditions
Privacy
Policy
Claims

Newsletter

Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Nullam tincidunt erat enim.

@ 2021 - Blogy - Designed & Developed by Zakirsoft



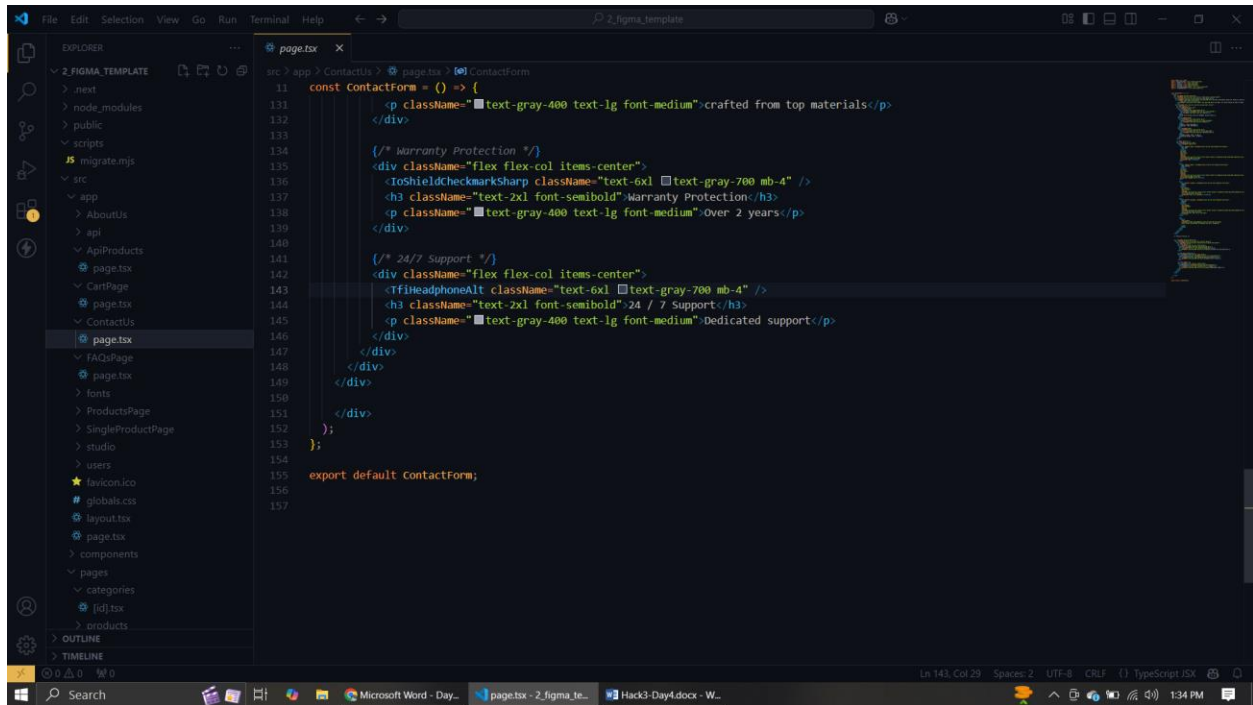
6.FAQs & Contact Us Component:

```
1 import React from "react";
2 import { FaPlus } from "react-icons/fa";
3
4 const FAQSection = () => {
5   const faqs = [
6     { question: "What types of chairs do you offer?", answer: "Lorem ipsum dolor sit amet consectetur adipiscing elit. ..." },
7     { question: "Do your chairs come with a warranty?", answer: "Lorem ipsum dolor sit amet consectetur adipiscing elit. ..." },
8     { question: "Can I try a chair before purchasing?", answer: "Lorem ipsum dolor sit amet consectetur adipiscing elit. ..." },
9     { question: "How can we get in touch with you?", answer: "Lorem ipsum dolor sit amet consectetur adipiscing elit. ..." },
10    { question: "What will be delivered? And When?", answer: "Lorem ipsum dolor sit amet consectetur adipiscing elit. ..." },
11    { question: "How do I clean and maintain my Comfortly chair?", answer: "Lorem ipsum dolor sit amet consectetur adipiscing elit. ..." }
12  ];
13
14  return (
15    <div className="py-10">
16      <div className="max-w-7xl mx-auto px-4 sm:px-6 lg:px-8">
17        <h2>Questions Looks Here</h2>
18        <div className="text-center mb-12">
19          <p>Lorem ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the</p>
20        </div>
21        <div>
22          <div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-2 gap-6">
23            {faqs.map((faq, index) => (
24              <div key={index} className="bg-gray-100 shadow-sm rounded-lg p-6">
25                <div className="flex justify-between items-center">
26                  <h3>{faq.question}</h3>
27                  <FaPlus className="text-gray-500" />
28                </div>
29                <p>{faq.answer}</p>
30              </div>
31            ))}
32          </div>
33        </div>
34      </div>
35    </div>
36  );
37}
```

Questions Looks Here

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the

<div>What types of chairs do you offer?</div> <div>+</div> <div>Lorem ipsum dolor sit amet consectetur adipiscing elit. ...</div>	<div>Do your chairs come with a warranty?</div> <div>+</div> <div>Lorem ipsum dolor sit amet consectetur adipiscing elit. ...</div>
<div>Can I try a chair before purchasing?</div> <div>+</div> <div>Lorem ipsum dolor sit amet consectetur adipiscing elit. ...</div>	<div>How can we get in touch with you?</div> <div>+</div> <div>Lorem ipsum dolor sit amet consectetur adipiscing elit. ...</div>
<div>What will be delivered? And When?</div> <div>+</div> <div>Lorem ipsum dolor sit amet consectetur adipiscing elit. ...</div>	<div>How do I clean and maintain my Comfortly chair?</div> <div>+</div> <div>Lorem ipsum dolor sit amet consectetur adipiscing elit. ...</div>



Get In Touch With Us

For More Information About Our Product & Service. Please Feel Free To Drop Us
An Email. Our Staff Always Be There To Help You Out. Do Not Hesitate!

Address

236 5th SE Avenue, New York NY10000, United States

Phone

Mobile: +(84) 546-6789

Hotline: +(84) 456-6789

Working Time

Monday-Friday: 9:00 - 22:00

Saturday-Sunday: 9:00 - 21:00

Your name

Abc

Email address

Abc@def.com

Subject

This is an optional

7.Subscription Component:

Or Subscribe To The Newsletter

Email Address...

SUBMIT

Follow Products And Discounts On Instagram



Technical Documentation

Report

1. Steps Taken to Build and Integrate Components

1.1 Building the Products and Categories Pages

- **Folder Structure:** Established a structured file hierarchy within the `src/app` directory to align with the App Router structure of Next.js.
 - `src/app/ApiProducts/page.tsx`: Main page for displaying products and categories.
 - `src/app/products/[id]/page.tsx`: Dynamic route for individual product details.
 - `src/app/categories/[id]/page.tsx`: Dynamic route for individual category details.
- **Sanity Setup:**
 - Created two GROQ queries to fetch `products` and `categories` data from Sanity CMS.
 - Implemented the Sanity client for server-side data fetching.
- **Component Integration:**
 - Designed reusable components for displaying product cards and category cards.
 - Integrated dynamic `Link` components for seamless navigation to product and category detail pages.

1.2 Adding Dynamic Routing

- Utilized the `[id]` dynamic segment for creating product and category detail pages.
- Implemented logic to fetch the specific product or category based on the `id` from the URL parameters.
- Used `sanityClient.fetch()` within these dynamic routes to query data dynamically during runtime.

1.3 Filtering Products by Category

- Added a category filter section in `ApiProducts` page:
 - Created a `useState` hook to manage the selected category.
 - Applied filtering logic to display products based on the selected category.
 - Integrated buttons for "All Categories" and individual categories.
-

2. Challenges Faced and Solutions Implemented

2.1 404 Error on Dynamic Routes

- **Issue:** Clicking on product or category links resulted in a 404 error.
- **Solution:** Ensured that the `[id]` dynamic route folders and `page.tsx` files were correctly structured under `src/app`. Verified the `Link` components pointed to valid paths.

2.2 Sanity Query Errors

- **Issue:** Errors like `param $id referenced, but not provided` were encountered while querying specific products/categories.
- **Solution:**
 - Correctly passed the `id` parameter in the GROQ query.
 - Used the Next.js `params` object to extract the dynamic `id` for use in queries.

2.3 Link Component Warning

- **Issue:** Received an error about invalid usage of `<Link>` with a child `<a>` tag.
- **Solution:** Updated all `Link` components to use the `legacyBehavior` flag or removed the `<a>` tag as appropriate.

2.4 State Management for Filtering

- **Issue:** Filtering products by category did not update the displayed products correctly.
- **Solution:** Used the `useState` hook to manage the selected category and applied filtering logic to update the product list dynamically.

3. Best Practices Followed During Development

3.1 Code Organization

- Followed the App Router conventions of Next.js for file and folder structure.
- Segregated concerns by creating dedicated components and utility functions for reusable logic.

3.2 Readable and Maintainable Code

- Used descriptive variable and function names.
- Added comments to explain key sections of the code.

3.3 Performance Optimization

- Leveraged Next.js server-side rendering to fetch and display data efficiently.
- Minimized the number of state updates by batching operations wherever possible.

3.4 Error Handling

- Added error boundaries for fetching data from Sanity CMS.
- Validated the presence of data before rendering components to prevent runtime errors.

3.5 User Experience Enhancements

- Ensured responsive design using Tailwind CSS.
- Added hover and transition effects to product and category cards for improved interactivity.