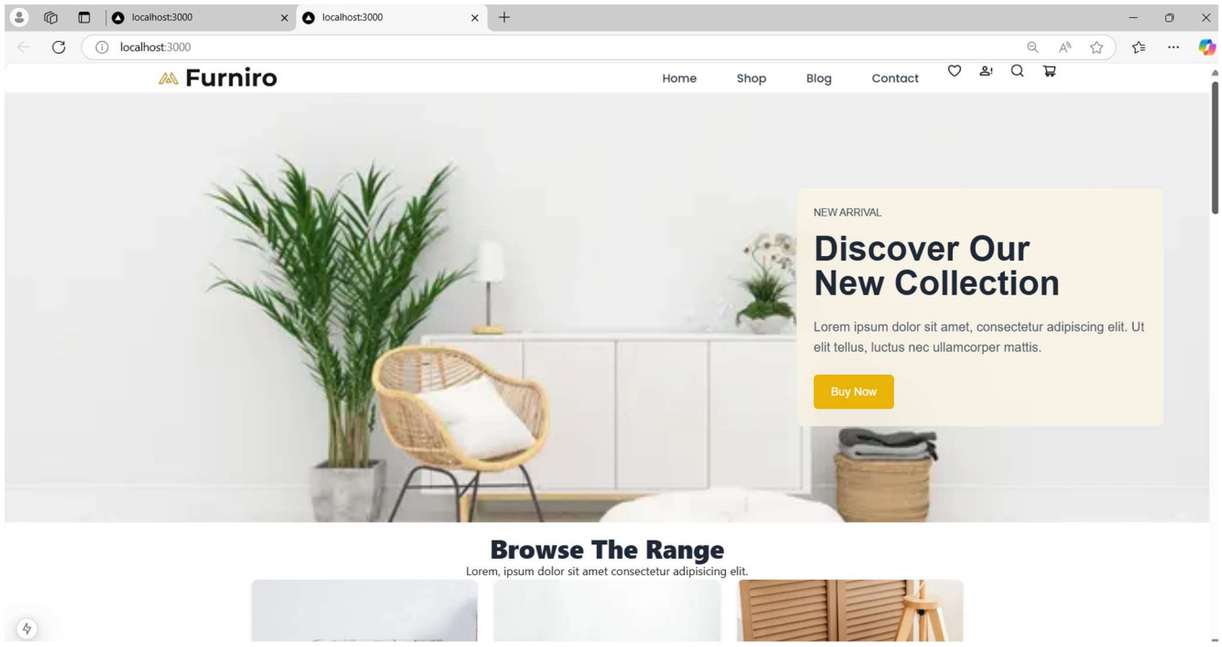
# DAY 4 - BUILDING DYNAMIC FRONTEND COMPONENTS FOR MARKETPLACE

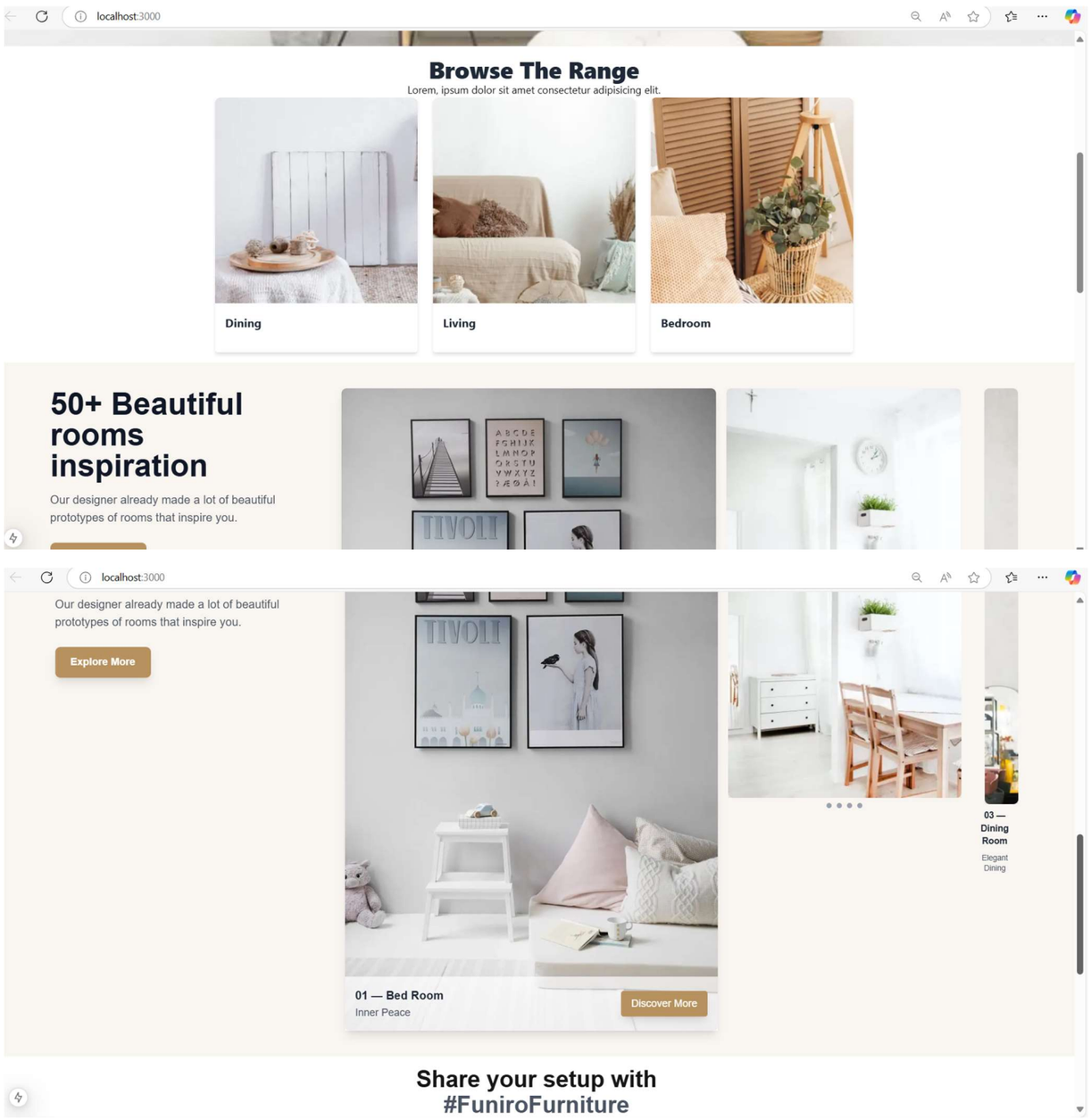
Zahida Raees

I have made copy version of template 6 and started working on it because my own marketplace require more time to make its UI and collection of product images and other details.

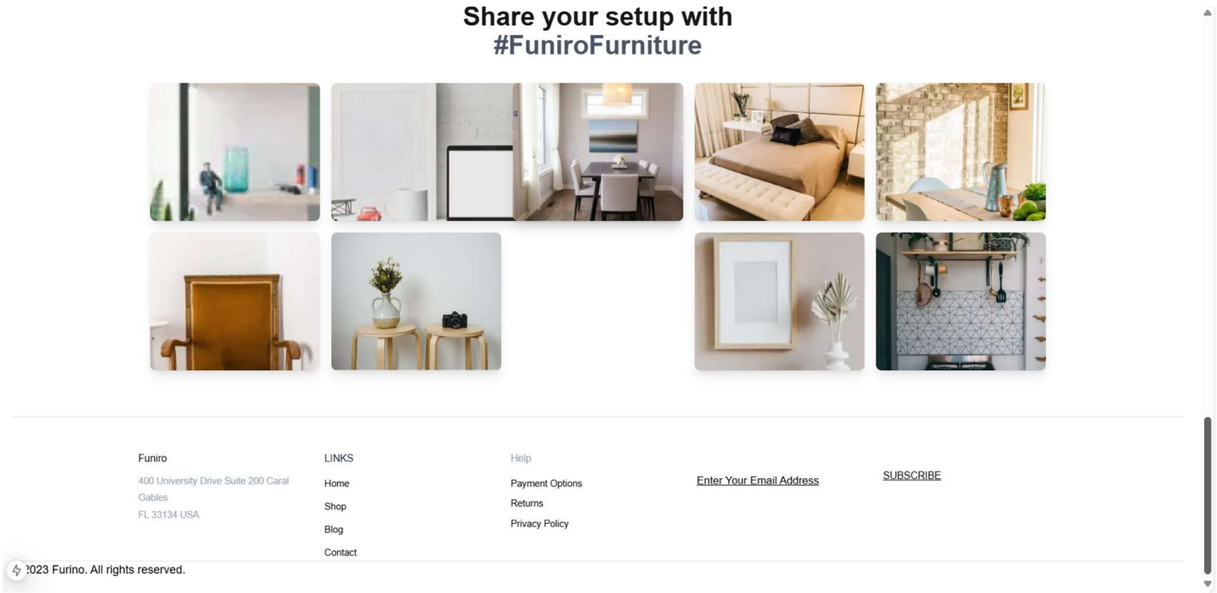
Here is my website’s landing page which is modular and responsive:



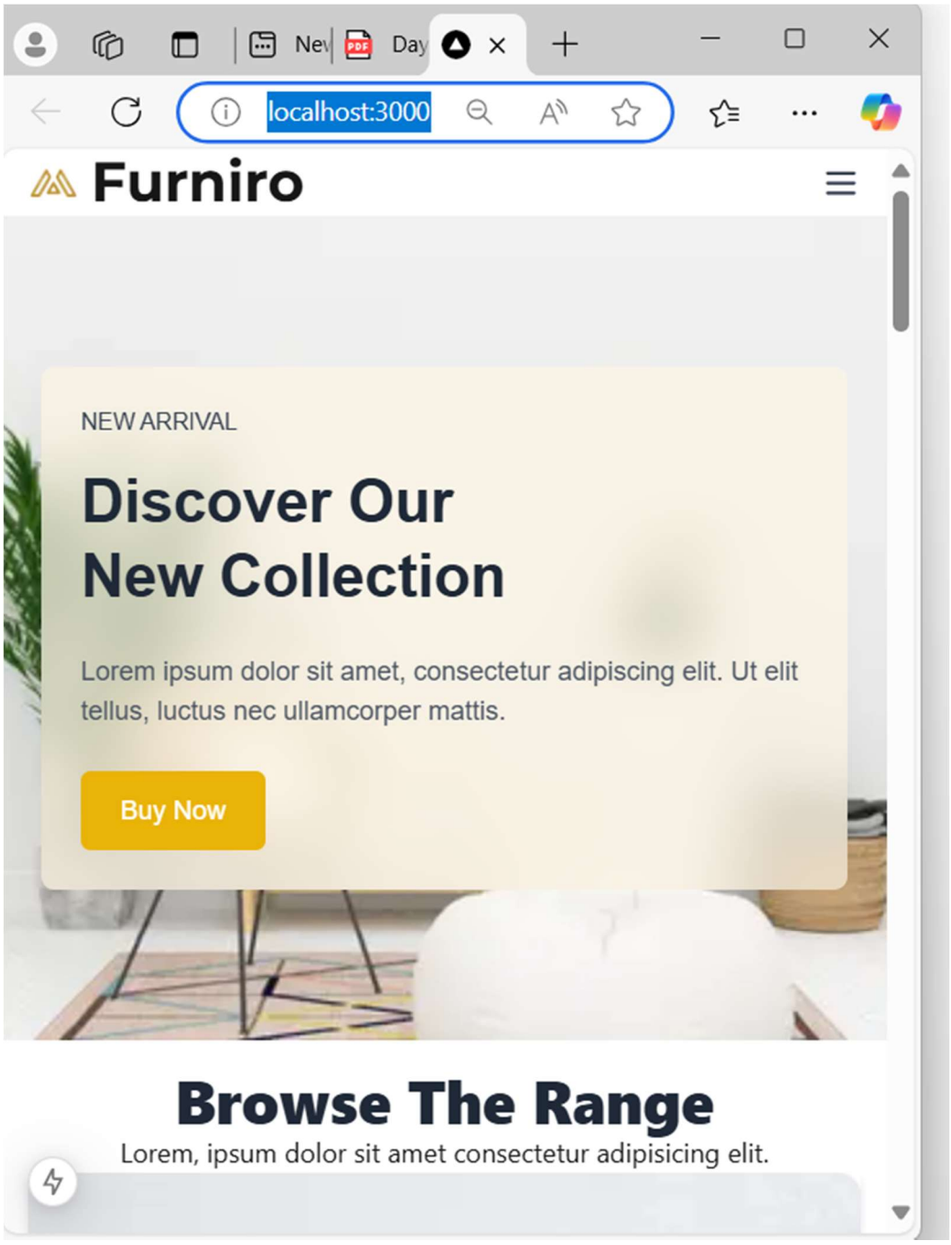

**Here is landing page’s second section but its data is coming from local drive**

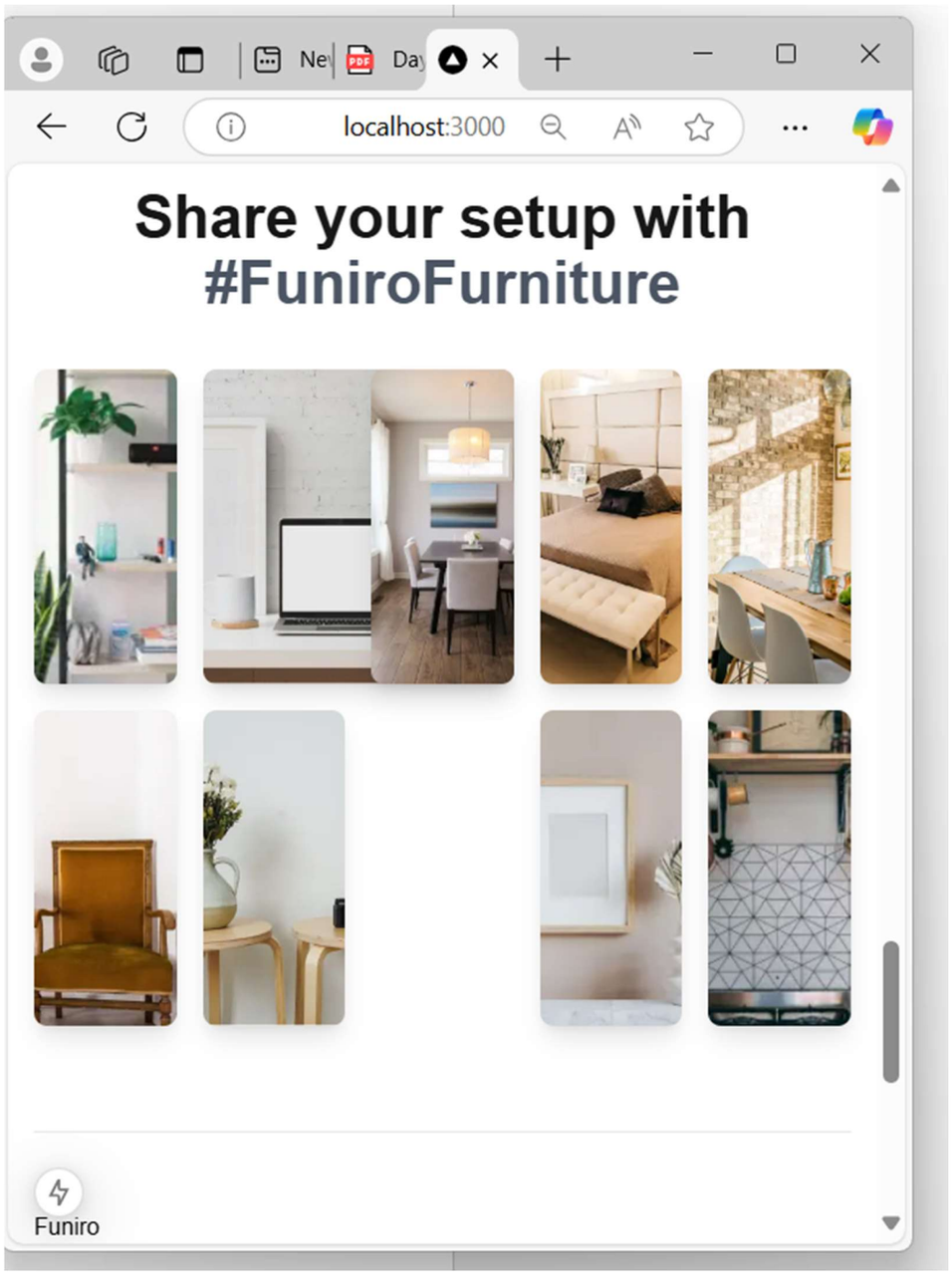


I have removed its 3rd section because I have to

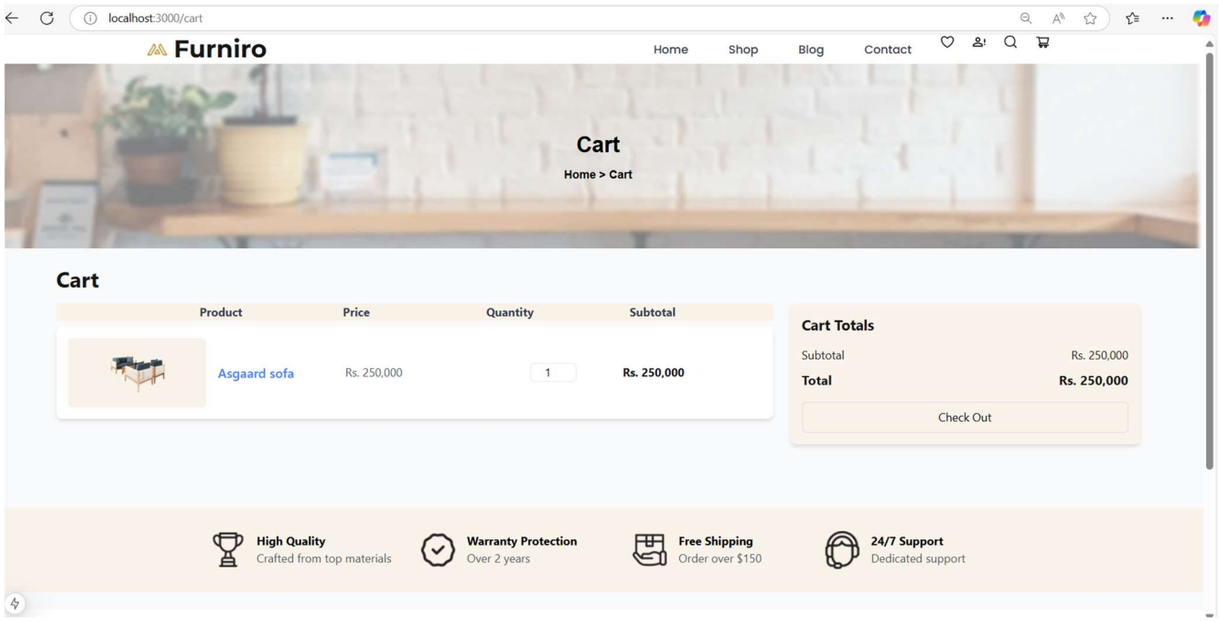



**Here is its responsiveness:**

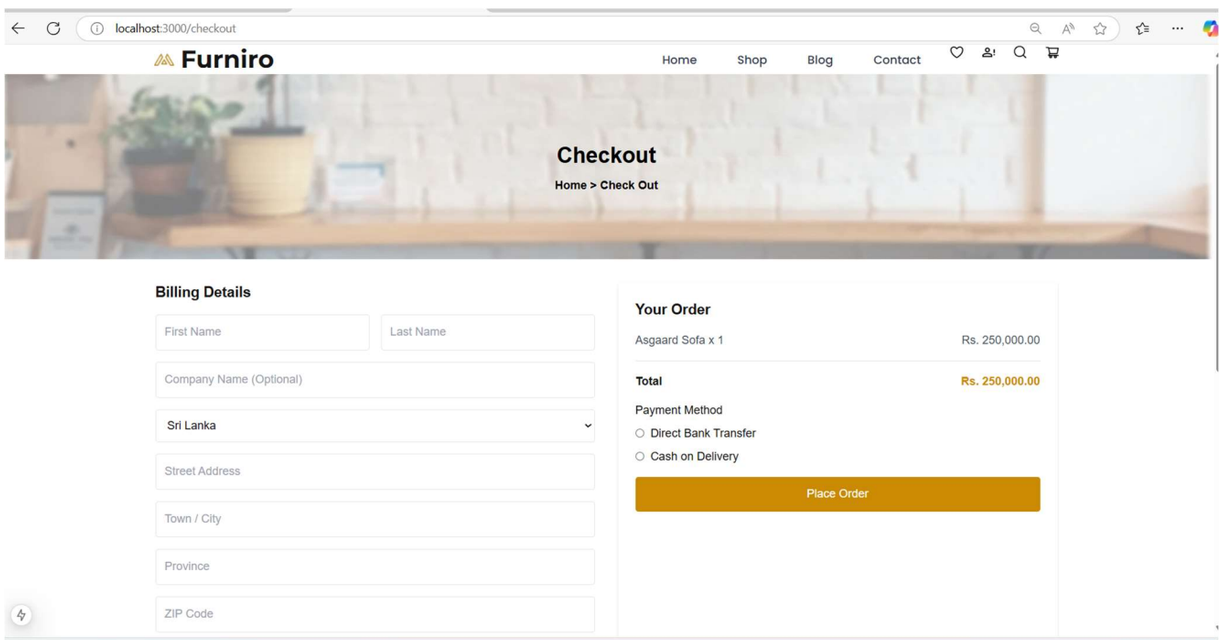



localhost:3000
Share your setup with
#FuniroFurniture
Funiro

# Cart Page data from public folder




**Although this product is from local folder but it’s clickable and button of check out is also working.**

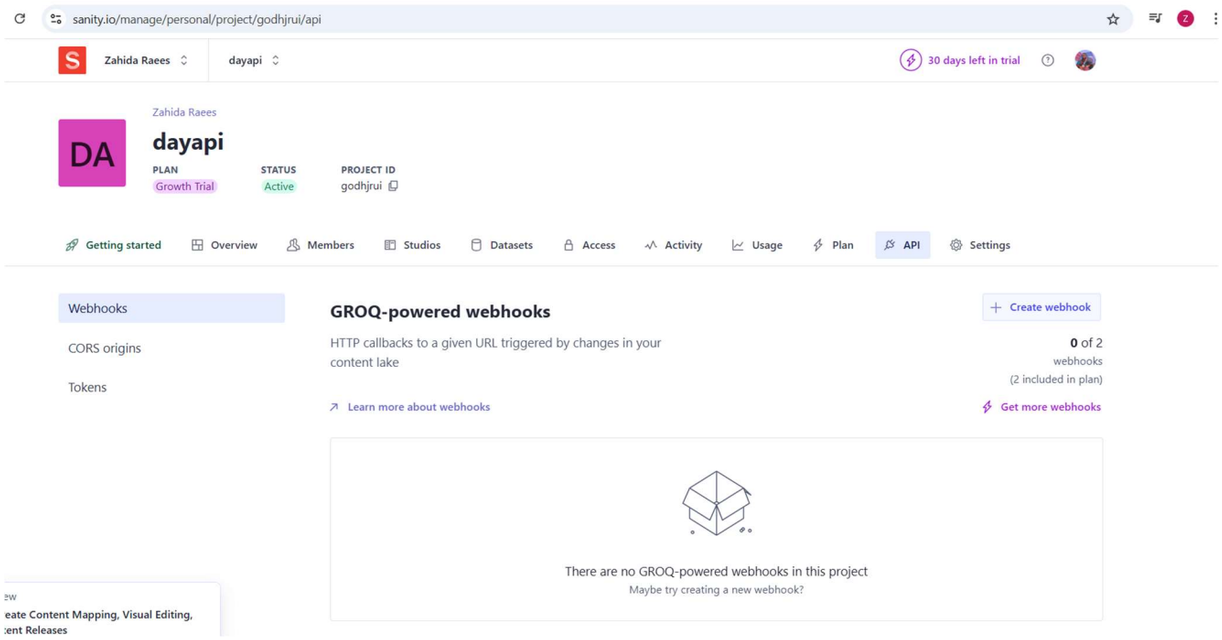


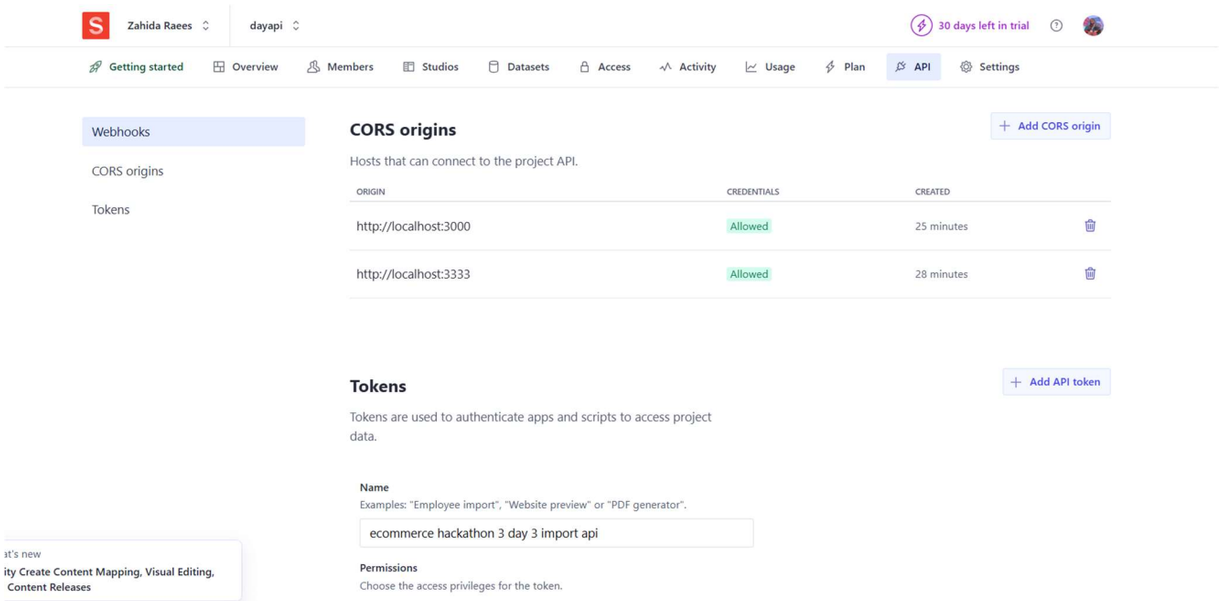
# Day 4 ASSIGNMENT

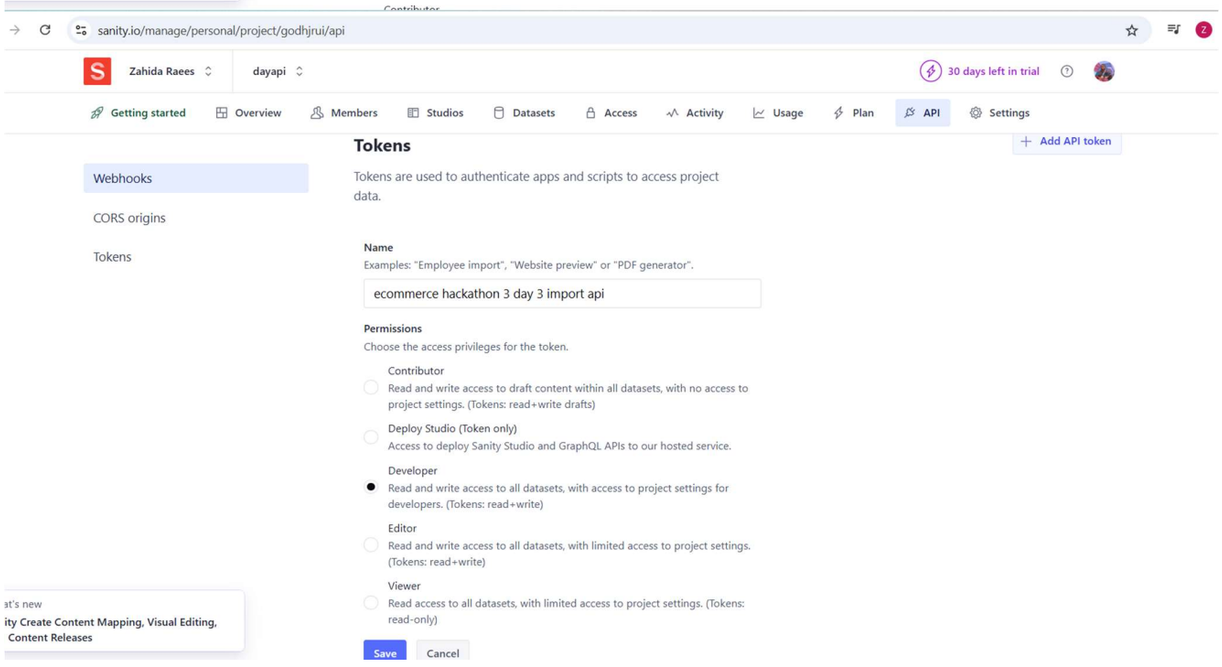
For day 4 I have utilized sanity data which I imported from your mentioned document on day 3

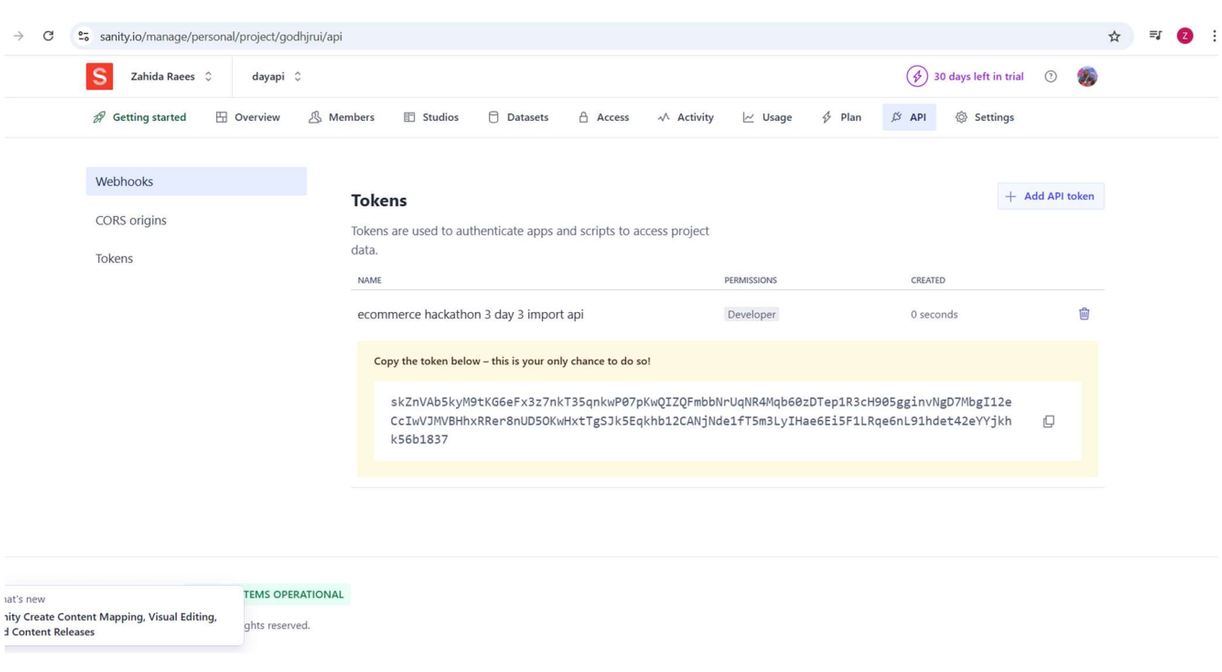
First I have updated **.env.local** file with variables of product id, dataset and then token API from sanity project page. Here is the screen short of Sanity Project:

## Sanity Project

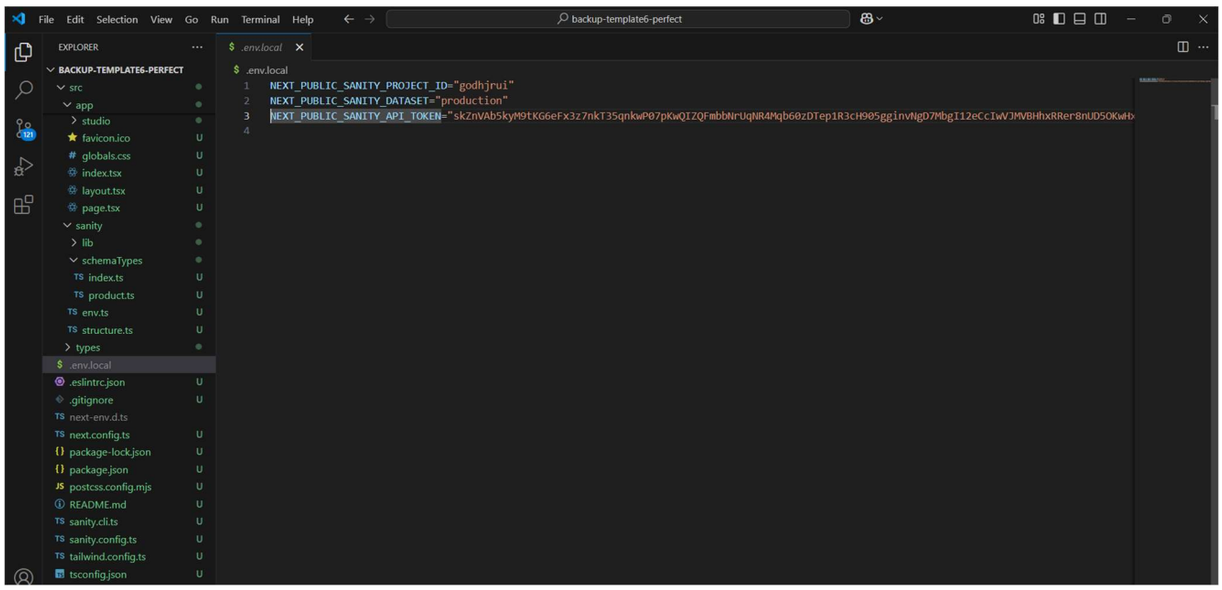



Zahida Raees
dayapi
30 days left in trial
Getting started
Overview
Members
Studios
Datasets
Access
Activity
Usage
Plan
API
Settings
Webhooks
CORS origins
Tokens
CORS origins
Add CORS origin
Hosts that can connect to the project API.
ORIGIN
CREDENTIALS
CREATED
http://localhost:3000
Allowed
25 minutes
http://localhost:3333
Allowed
28 minutes
Tokens
Add API token
Tokens are used to authenticate apps and scripts to access project data.
Name
Examples: "Employee import", "Website preview" or "PDF generator".
ecommerce hackathon 3 day 3 import api
Permissions
Choose the access privileges for the token.


sanity.io/manage/personal/project/godhjrui/api
Zahida Raees
dayapi
30 days left in trial
Getting started
Overview
Members
Studios
Datasets
Access
Activity
Usage
Plan
API
Settings
Webhooks
CORS origins
Tokens
Tokens
Add API token
Tokens are used to authenticate apps and scripts to access project data.
Name
Examples: "Employee import", "Website preview" or "PDF generator".
ecommerce hackathon 3 day 3 import api
Permissions
Choose the access privileges for the token.
Contributor
Read and write access to draft content within all datasets, with no access to project settings. (Tokens: read+write drafts)
Deploy Studio (Token only)
Access to deploy Sanity Studio and GraphQL APIs to our hosted service.
Developer
Read and write access to all datasets, with access to project settings for developers. (Tokens: read+write)
Editor
Read and write access to all datasets, with limited access to project settings. (Tokens: read+write)
Viewer
Read access to all datasets, with limited access to project settings. (Tokens: read-only)
Save
Cancel

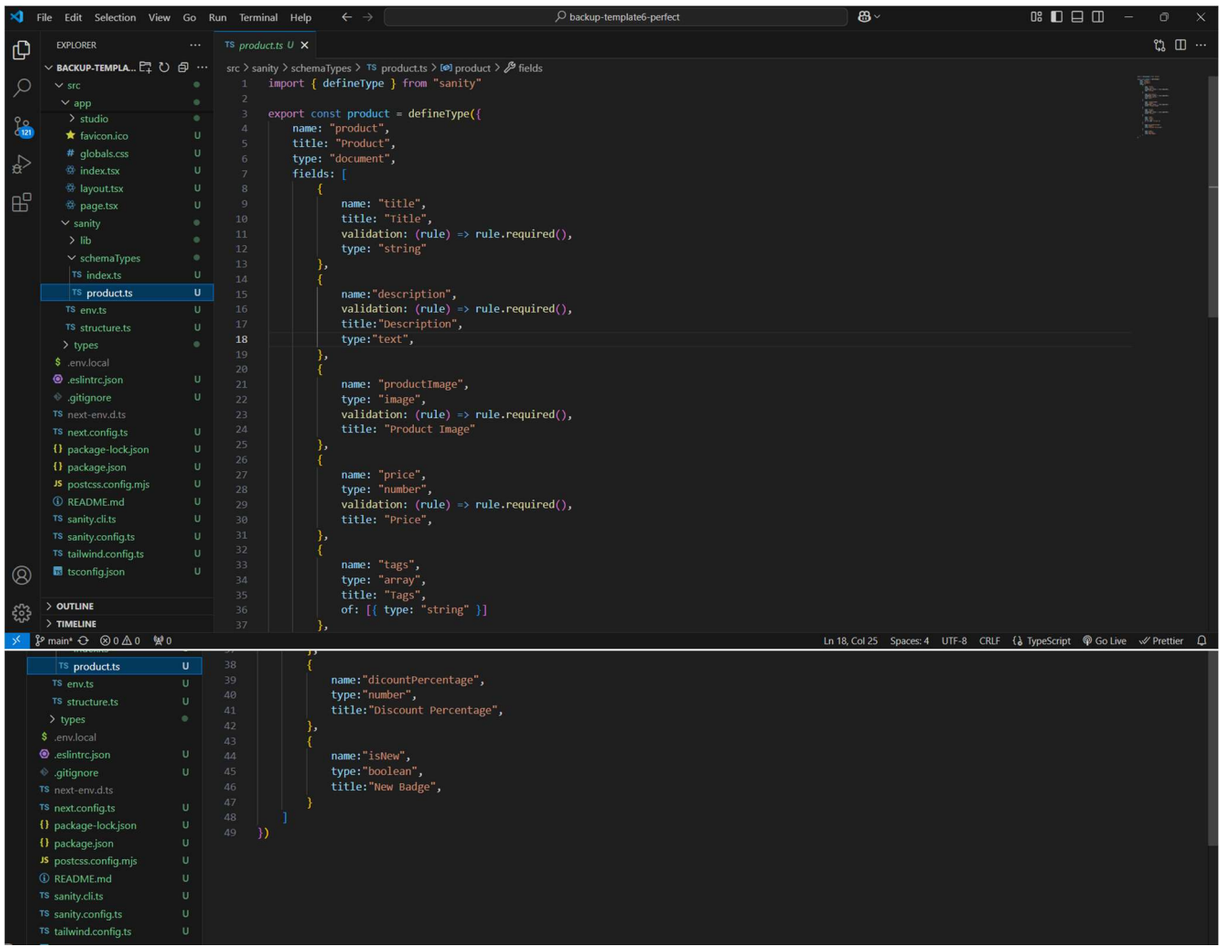

sanity.io/manage/personal/project/godhjrui/api
Zahida Raees
dayapi
30 days left in trial
Getting started
Overview
Members
Studios
Datasets
Access
Activity
Usage
Plan
API
Settings
Webhooks
CORS origins
Tokens
Tokens
Add API token
Tokens are used to authenticate apps and scripts to access project data.
NAME
PERMISSIONS
CREATED
ecommerce hackathon 3 day 3 import api
Developer
0 seconds
Copy the token below – this is your only chance to do so!
skZnVAb5kyM9tKG6eFx3z7nkT35qnkwP07pKwQIZQFmbbNrUqNR4Mqb60zDTep1R3cH905gginvNgD7MbgI12e
CcIwVJMVBHhxRRer8nUD5OKwHxtTgSJk5Eqkhb12CANjNde1fT5m3LyIHae6Ei5F1LRqe6nL91hdet42eYYjkh
k56b1837

# .env.local



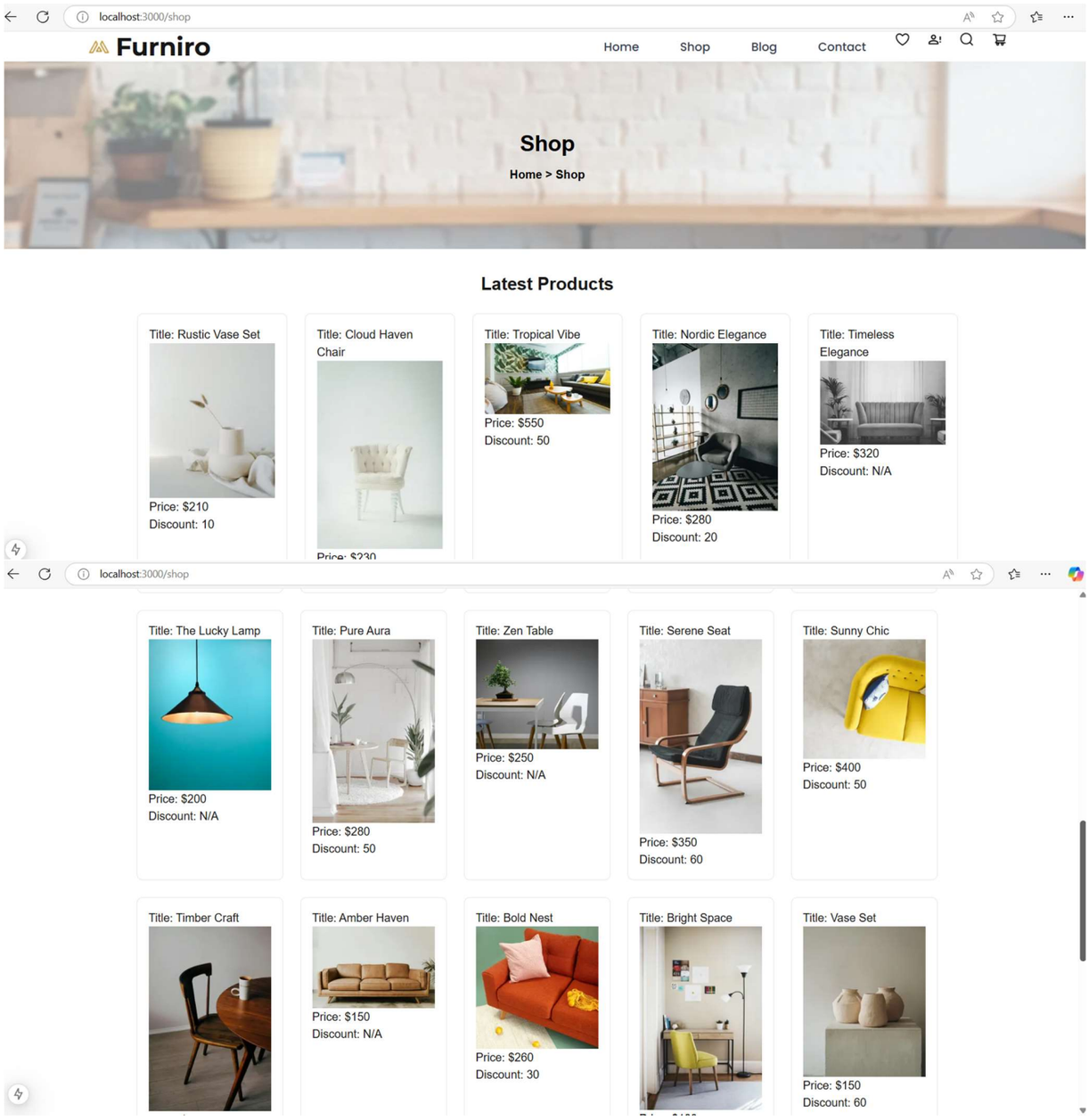

# Product Schema

Here is the screen product schema which I got from the pdf about template 6.



# Product listing

Product listing on shop page with Product title, image, price, discount. Every image is clickable. Here header title is also dynamic means I am using single image for all pages excluding landing page. Using parameters of title and breadcrumb to update titles.



localhost:3000/shop

Title: Timber Craft

Price: $320
Discount: 30

Title: Amber Haven

Price: $150
Discount: N/A

Title: Bold Nest

Price: $260
Discount: 30

Title: Bright Space

Price: $180
Discount: 10

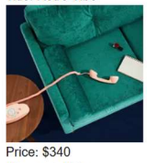
Title: Vase Set

Price: $150
Discount: 60

Title: Wood Chair

Price: $100
Discount: 10

Title: Retro Vibe

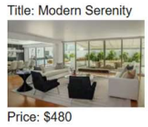


Price: $340
Discount: N/A

Title: Sleek Living

Price: $300
Discount: N/A

Title: Modern Serenity



Price: $480
Discount: N/A

localhost:3000/shop

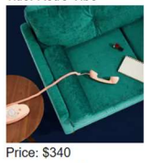
Discount: 30

Discount: 10

Title: Wood Chair

Price: $100
Discount: 10

Title: Retro Vibe

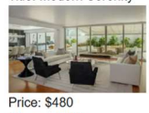


Price: $340
Discount: N/A

Title: Sleek Living

Price: $300
Discount: N/A

Title: Modern Serenity



Price: $480
Discount: N/A



localhost:3000/shop

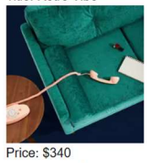
Discount: 30

Discount: 10

Title: Wood Chair

Price: $100
Discount: 10

Title: Retro Vibe

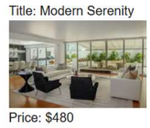


Price: $340
Discount: N/A

Title: Sleek Living

Price: $300
Discount: N/A

Title: Modern Serenity



Price: $480
Discount: N/A

Funiro

400 University Drive Suite 200 Caral Gables
FL 33134 USA

LINKS

Home
Shop
Blog
Contact

Help

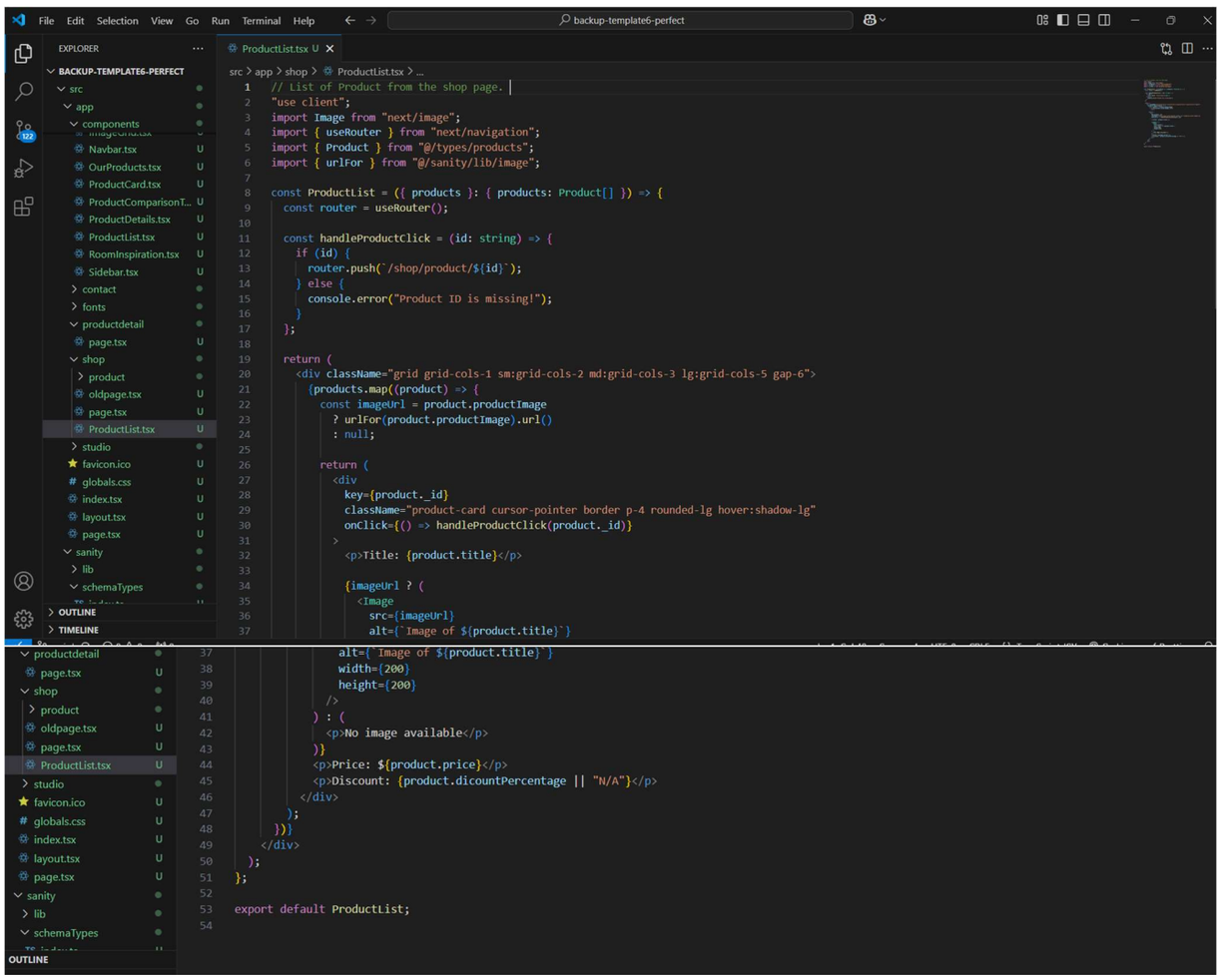
Payment Options
Returns
Privacy Policy

Enter Your Email Address

SUBSCRIBE

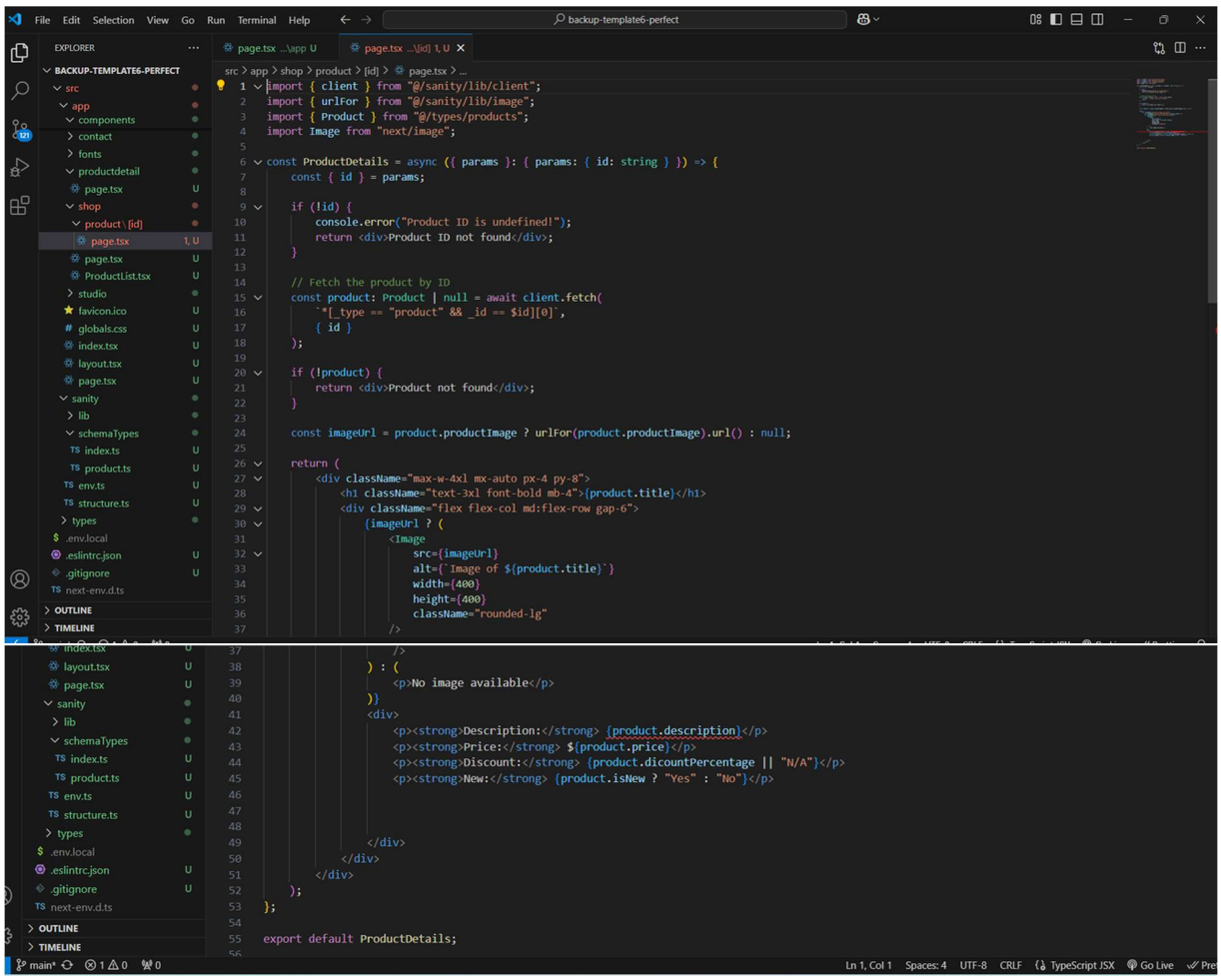
2023 Furino. All rights reserved.

# Product listing code

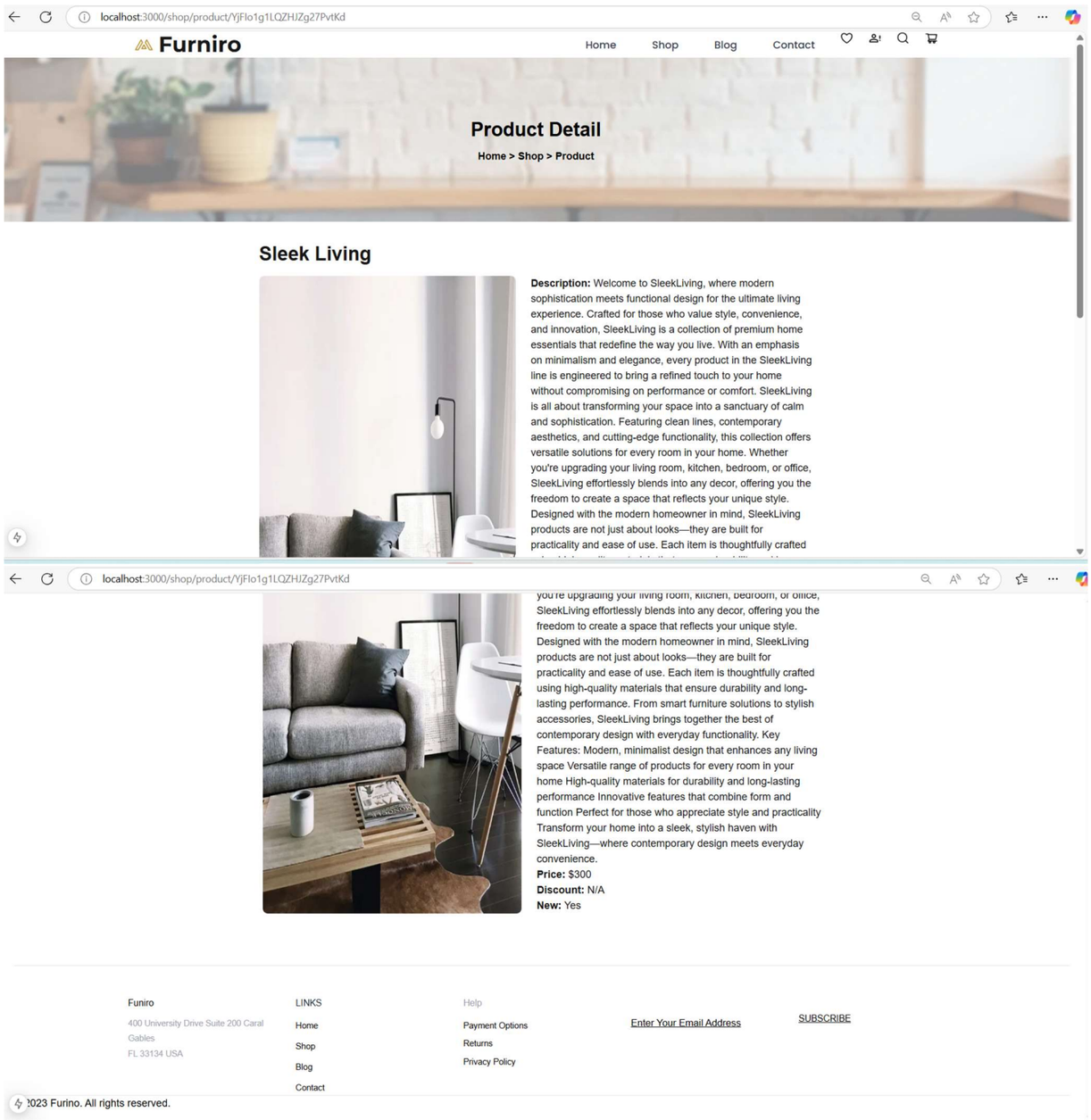


# Single Product Code:

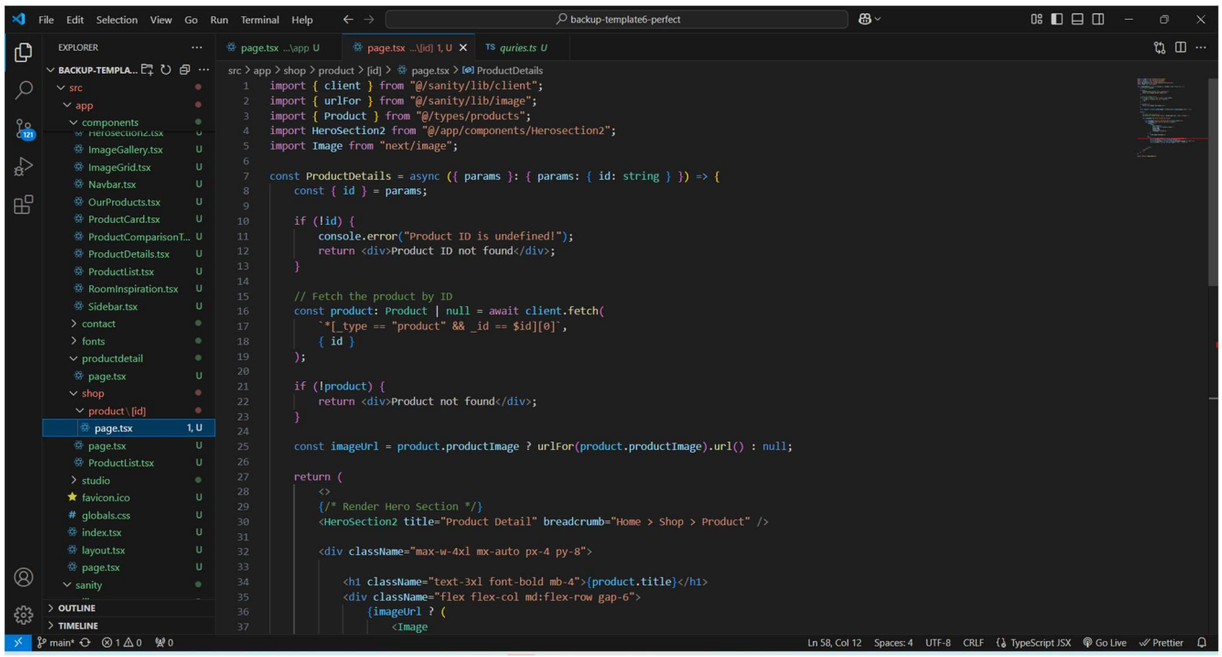
Here I am getting error in rich text field. On display there is no error but in coding I am getting it and still in search to know its reason.



# Single Product Screen Shot

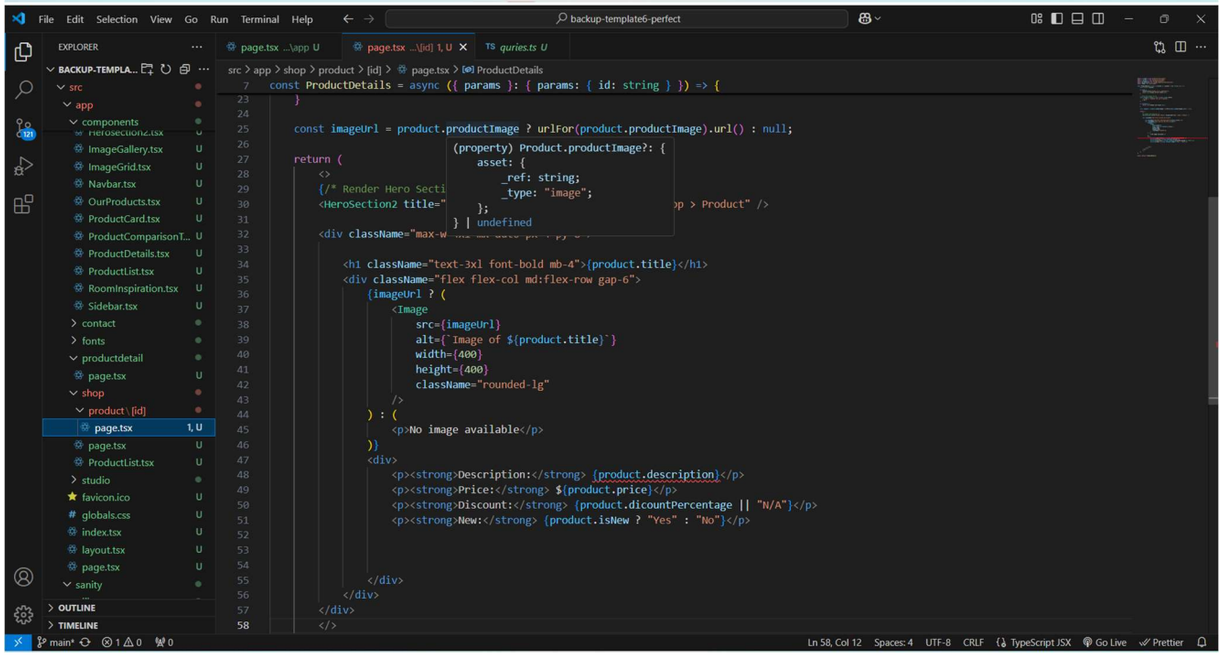



Here is single product code


File Edit Selection View Go Run Terminal Help
backup-template6-perfect
EXPLORER
page.tsx ...\app U
page.tsx ...\[id] 1, U
queries.ts U
src > app > shop > product > [id] > page.tsx > ProductDetails
import { client } from "@/sanity/lib/client";
import { urlFor } from "@/sanity/lib/image";
import { Product } from "@/types/products";
import HeroSection2 from "@/app/components/Herosection2";
import Image from "next/image";
const ProductDetails = async ({ params }: { params: { id: string } }) => {
const { id } = params;
if (!id) {
console.error("Product ID is undefined!");
return Product ID not found ;
}
// Fetch the product by ID
const product: Product | null = await client.fetch(
`*[_type == "product" && _id == $id][0]`,
{ id }
);
if (!product) {
return Product not found ;
}
const imageUrl = product.productImage ? urlFor(product.productImage).url() : null;
return (
<>
{/* Render Hero Section */}
<HeroSection2 title="Product Detail" breadcrumb="Home > Shop > Product" />

<h1 className="text-3xl font-bold mb-4">{product.title}</h1>

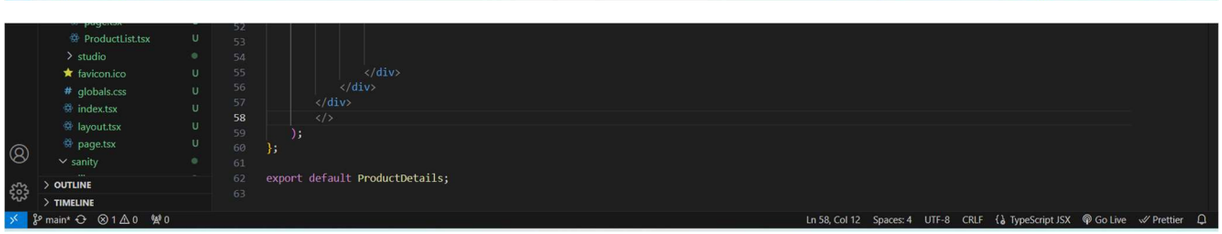
{imageUrl ? (
<Image
main* 1 0 0
Ln 58, Col 12 Spaces: 4 UTF-8 CRLF TypeScript JSX Go Live Prettier


File Edit Selection View Go Run Terminal Help
backup-template6-perfect
EXPLORER
page.tsx ...\app U
page.tsx ...\[id] 1, U
queries.ts U
src > app > shop > product > [id] > page.tsx > ProductDetails
const ProductDetails = async ({ params }: { params: { id: string } }) => {
}
const imageUrl = product.productImage ? urlFor(product.productImage).url() : null;
(property) Product.productImage?: {
asset: {
_ref: string;
_type: "image";
};
} | undefined
return (
<>
{/* Render Hero Secti
<HeroSection2 title="
op > Product" />
<div className="max-w
<h1 className="text-3xl font-bold mb-4">{product.title}</h1>

{imageUrl ? (
<Image
src={imageUrl}
alt={`Image of ${product.title}`}
width={400}
height={400}
className="rounded-lg"
/>
) : (
No image available
)}

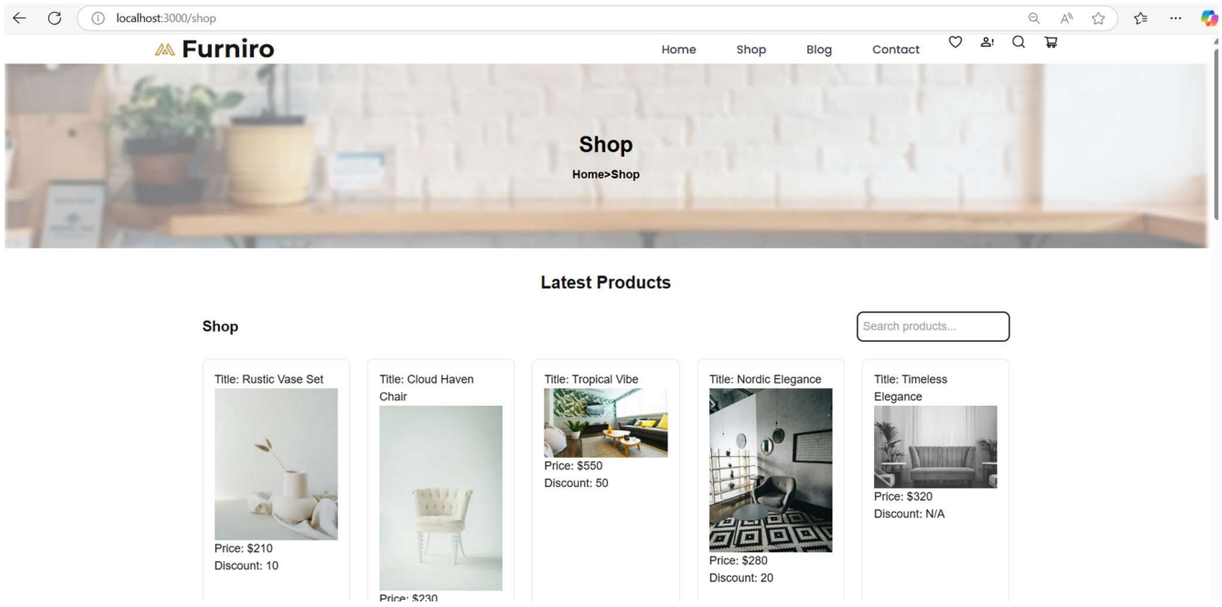
Description: {product.description}
Price: ${product.price}
Discount: {product.dicountPercentage || "N/A"}
New: {product.isNew ? "Yes" : "No"}

</>
main* 1 0 0
Ln 58, Col 12 Spaces: 4 UTF-8 CRLF TypeScript JSX Go Live Prettier

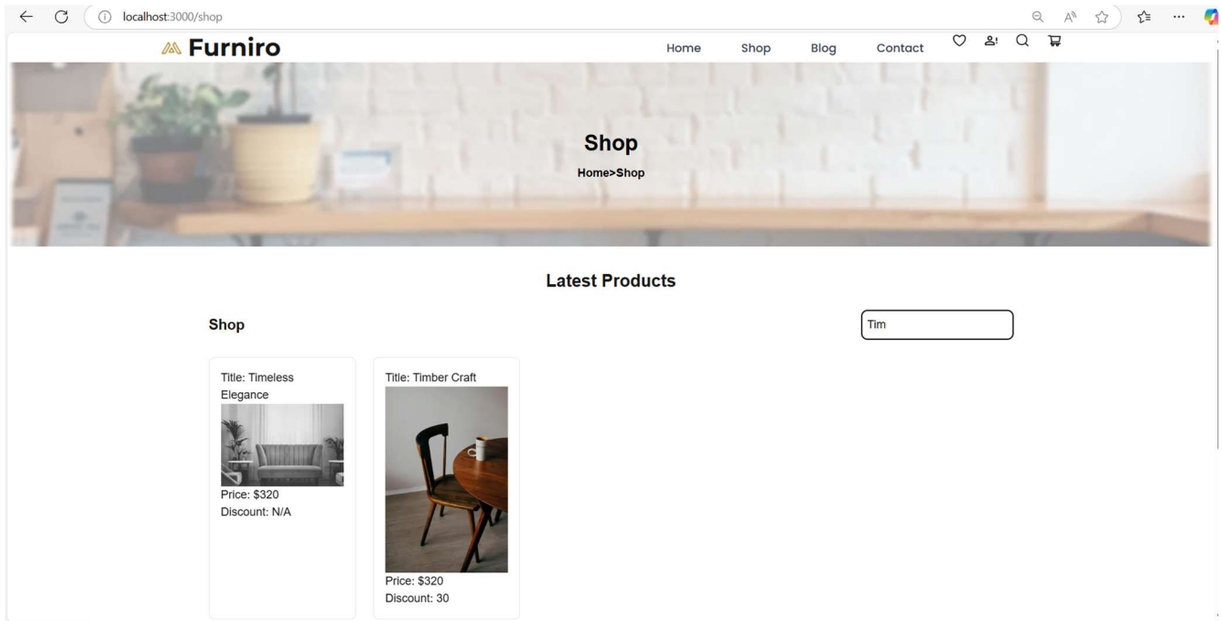

ProductList.tsx
studio
favicon.ico
globals.css
index.tsx
layout.tsx
page.tsx
sanity
OUTLINE
TIMELINE

</>
);
};
export default ProductDetails;
main* 1 0 0
Ln 58, Col 12 Spaces: 4 UTF-8 CRLF TypeScript JSX Go Live Prettier

# Now added Search Button of Product within Product List

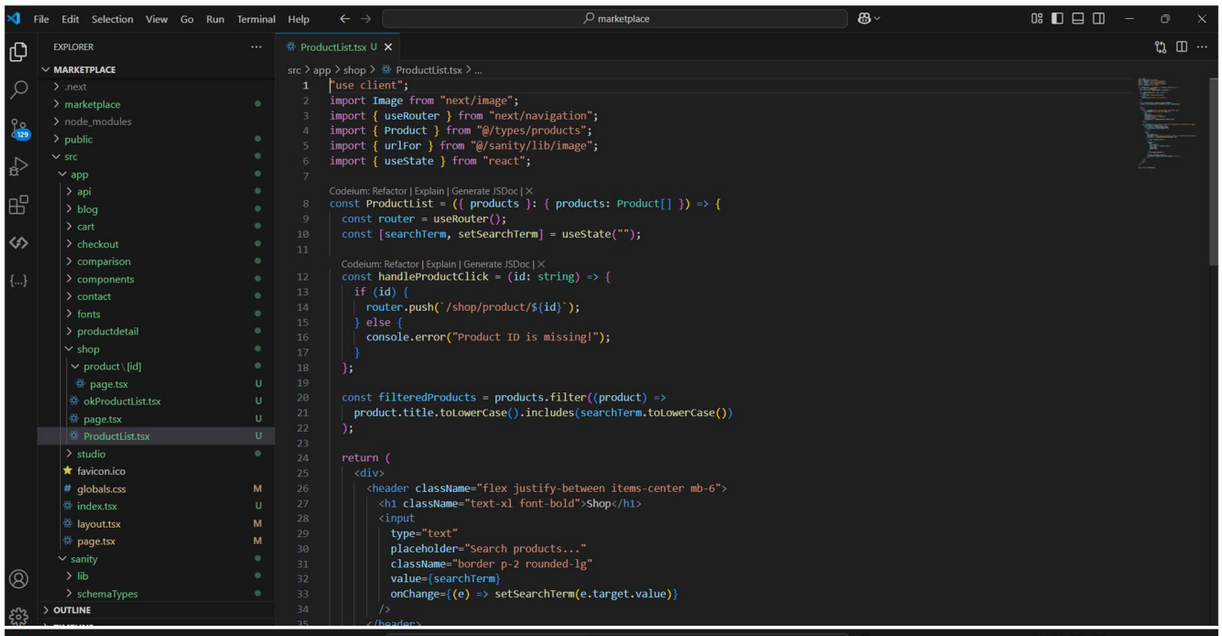


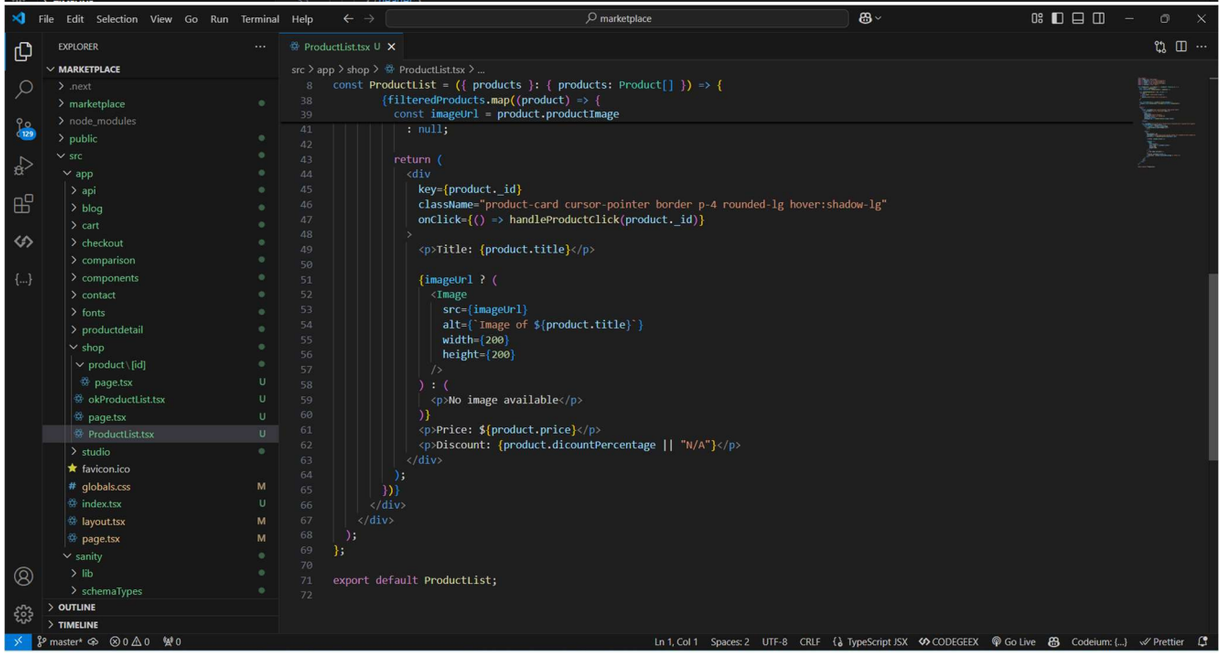

# Search Specific Product within Product List



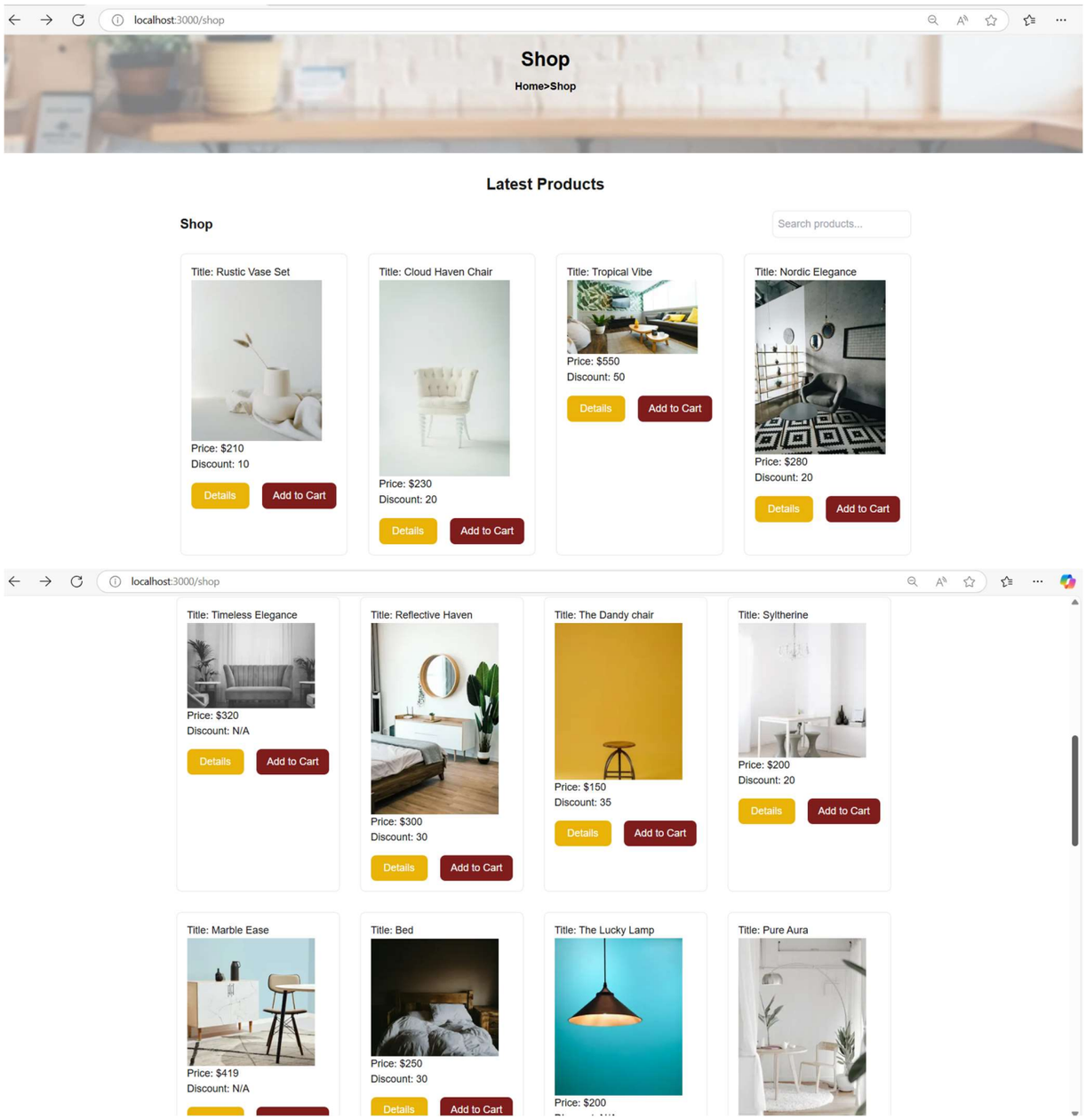
# Search Product within Product List Code

Here you will observe two files of ProductList.tsx which is because I always make copy of correct file to experiment new page or content and after successful completion of testing I remove previous file and update new one.

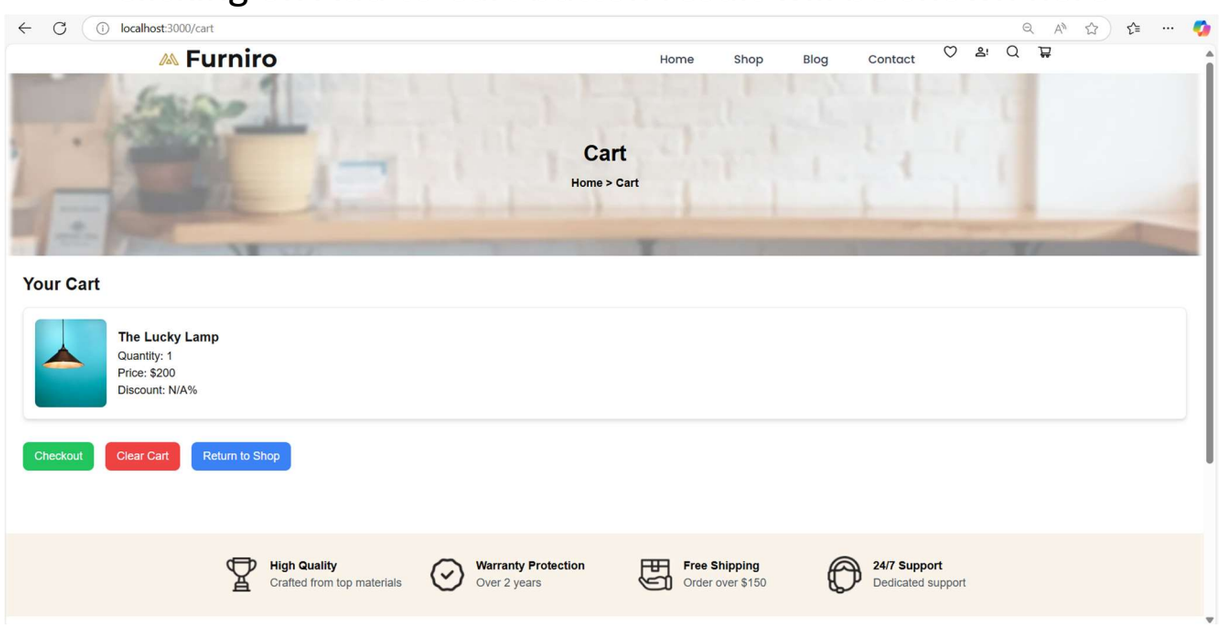


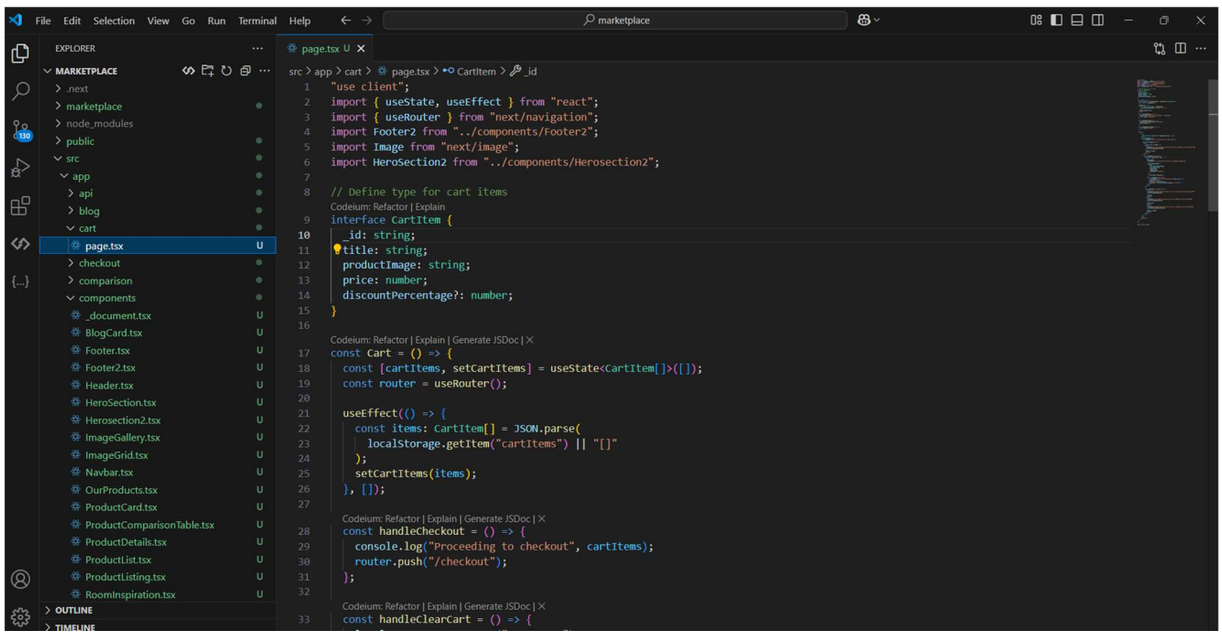
Now Added, Add to Cart and Product Detail Buttons under each product of Product List

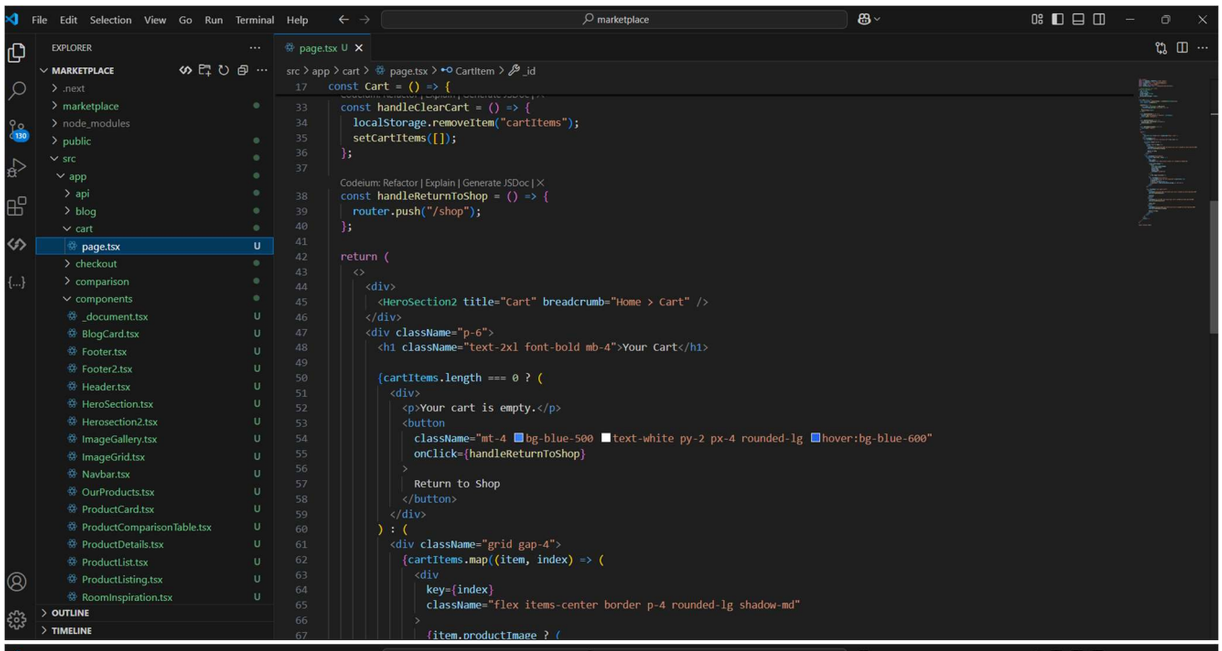



Clicking on Add to Cart button result will be shown here



# Here is the code of /cart/page.tsx




File Edit Selection View Go Run Terminal Help
marketplace
EXPLORER
MARKETPLACE
page.tsx U
src > app > cart > page.tsx > CartItem > _id
const Cart = () => {
const handleClearCart = () => {
localStorage.removeItem("cartItems");
setCartItems([]);
};
Codeium: Refactor | Explain | Generate JSDoc |
const handleReturnToShop = () => {
router.push("/shop");
};
return (
<>

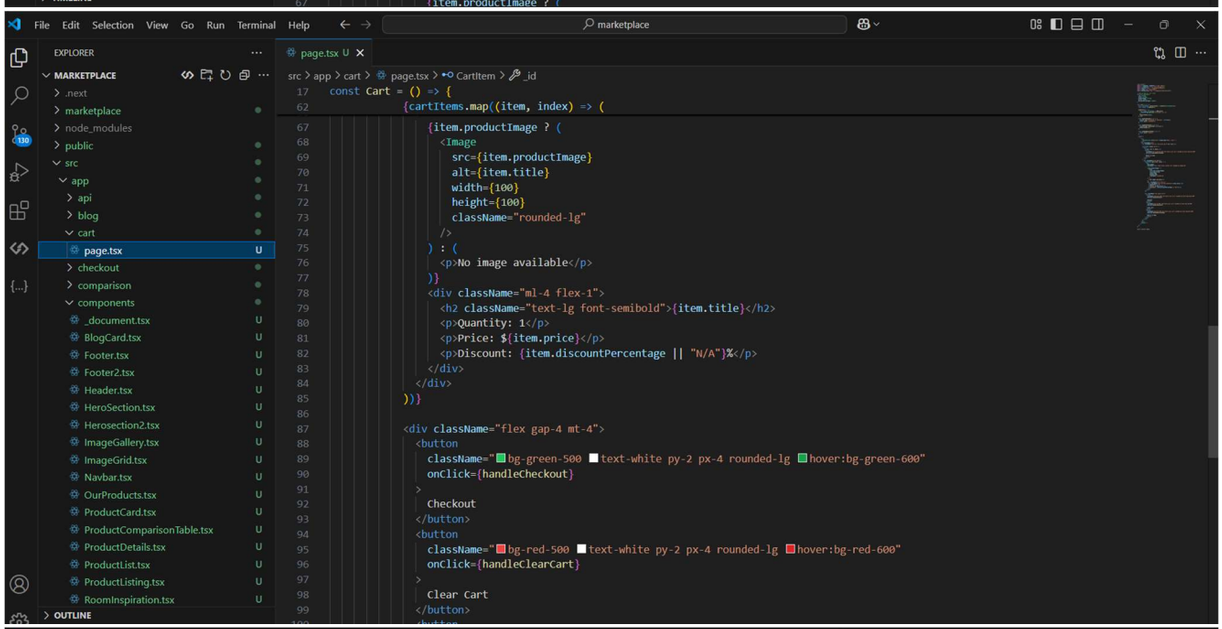
<HeroSection2 title="Cart" breadcrumb="Home > Cart" />

<h1 className="text-2xl font-bold mb-4">Your Cart</h1>
{cartItems.length === 0 ? (

Your cart is empty.
<button
className="mt-4 bg-blue-500 text-white py-2 px-4 rounded-lg hover:bg-blue-600"
onClick={handleReturnToShop}
>
Return to Shop
</button>

) : (

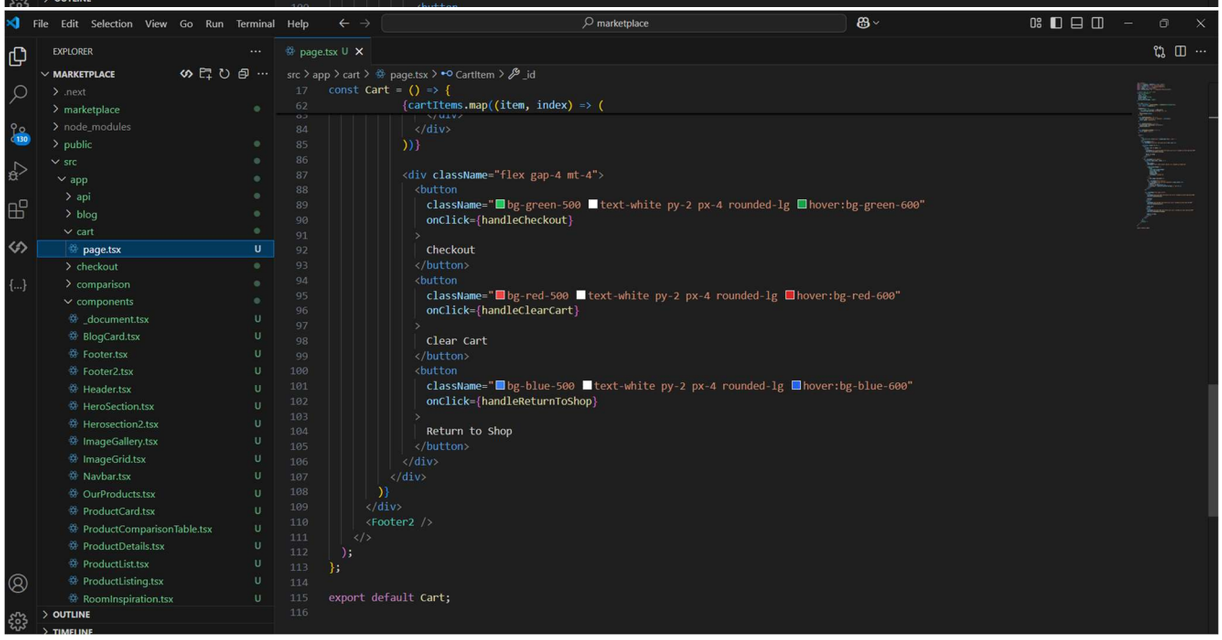
{cartItems.map((item, index) => (
<div
key={index}
className="flex items-center border p-4 rounded-lg shadow-md"
>
{item.productImage ? (
.next
marketplace
node_modules
public
src
app
api
blog
cart
page.tsx U
checkout
comparison
components
_document.tsx U
BlogCard.tsx U
Footer.tsx U
Footer2.tsx U
Header.tsx U
HeroSection.tsx U
Herosection2.tsx U
ImageGallery.tsx U
ImageGrid.tsx U
Navbar.tsx U
OurProducts.tsx U
ProductCard.tsx U
ProductComparisonTable.tsx U
ProductDetails.tsx U
ProductList.tsx U
ProductListing.tsx U
RoomInspiration.tsx U
OUTLINE
TIMELINE


File Edit Selection View Go Run Terminal Help
marketplace
EXPLORER
MARKETPLACE
page.tsx U
src > app > cart > page.tsx > CartItem > _id
const Cart = () => {
{cartItems.map((item, index) => (
{item.productImage ? (
<Image
src={item.productImage}
alt={item.title}
width={100}
height={100}
className="rounded-lg"
/>
) : (
No image available
)}

<h2 className="text-lg font-semibold">{item.title}</h2>
Quantity: 1
Price: ${item.price}
Discount: {item.discountPercentage || "N/A"}%

))}

<button
className="bg-green-500 text-white py-2 px-4 rounded-lg hover:bg-green-600"
onClick={handleCheckout}
>
Checkout
</button>
<button
className="bg-red-500 text-white py-2 px-4 rounded-lg hover:bg-red-600"
onClick={handleClearCart}
>
Clear Cart
</button>
.next
marketplace
node_modules
public
src
app
api
blog
cart
page.tsx U
checkout
comparison
components
_document.tsx U
BlogCard.tsx U
Footer.tsx U
Footer2.tsx U
Header.tsx U
HeroSection.tsx U
Herosection2.tsx U
ImageGallery.tsx U
ImageGrid.tsx U
Navbar.tsx U
OurProducts.tsx U
ProductCard.tsx U
ProductComparisonTable.tsx U
ProductDetails.tsx U
ProductList.tsx U
ProductListing.tsx U
RoomInspiration.tsx U
OUTLINE


File Edit Selection View Go Run Terminal Help
marketplace
EXPLORER
MARKETPLACE
page.tsx U
src > app > cart > page.tsx > CartItem > _id
const Cart = () => {
{cartItems.map((item, index) => (

))}

<button
className="bg-green-500 text-white py-2 px-4 rounded-lg hover:bg-green-600"
onClick={handleCheckout}
>
Checkout
</button>
<button
className="bg-red-500 text-white py-2 px-4 rounded-lg hover:bg-red-600"
onClick={handleClearCart}
>
Clear Cart
</button>
<button
className="bg-blue-500 text-white py-2 px-4 rounded-lg hover:bg-blue-600"
onClick={handleReturnToShop}
>
Return to Shop
</button>

)}

<Footer2 />
</>
);
};
export default Cart;
.next
marketplace
node_modules
public
src
app
api
blog
cart
page.tsx U
checkout
comparison
components
_document.tsx U
BlogCard.tsx U
Footer.tsx U
Footer2.tsx U
Header.tsx U
HeroSection.tsx U
Herosection2.tsx U
ImageGallery.tsx U
ImageGrid.tsx U
Navbar.tsx U
OurProducts.tsx U
ProductCard.tsx U
ProductComparisonTable.tsx U
ProductDetails.tsx U
ProductList.tsx U
ProductListing.tsx U
RoomInspiration.tsx U
OUTLINE
TIMELINE