

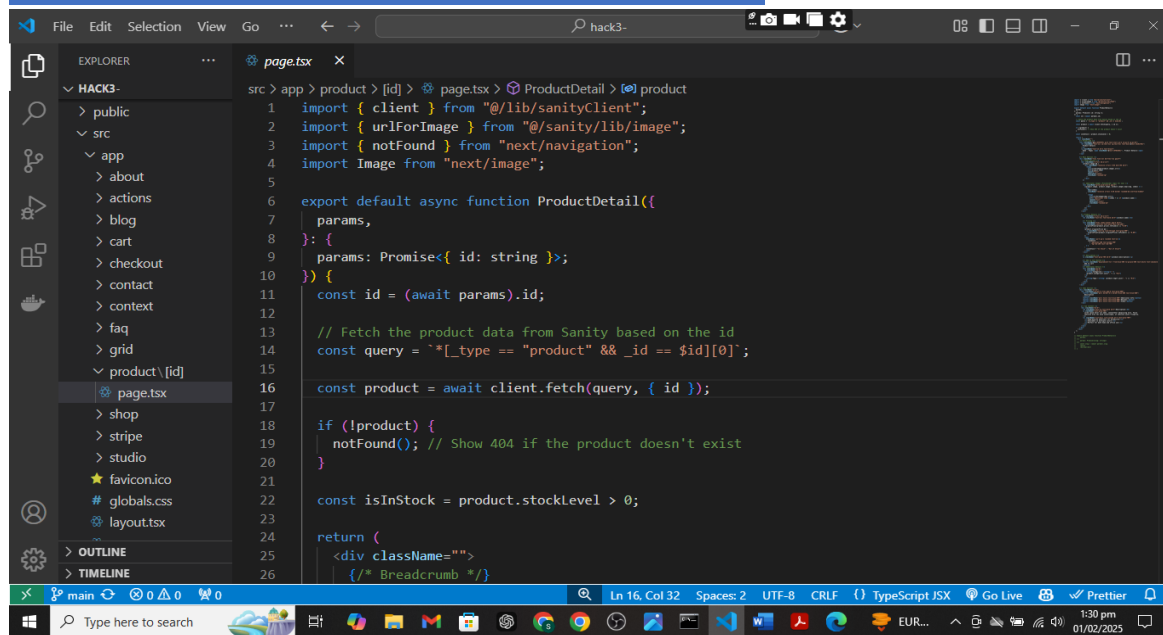
Day4 – Dynamic Fronted Components – [General E-commerce]

1. Functional Deliverables:

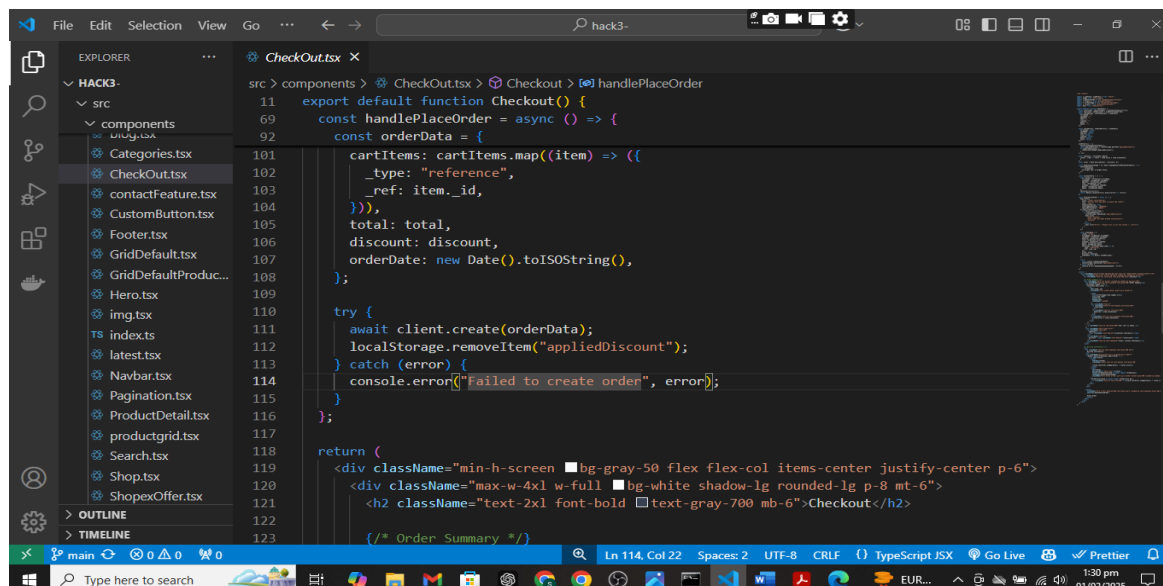
Screenshots of:

.The Product listing page with dynamic data-

Products Dynamic Page Code



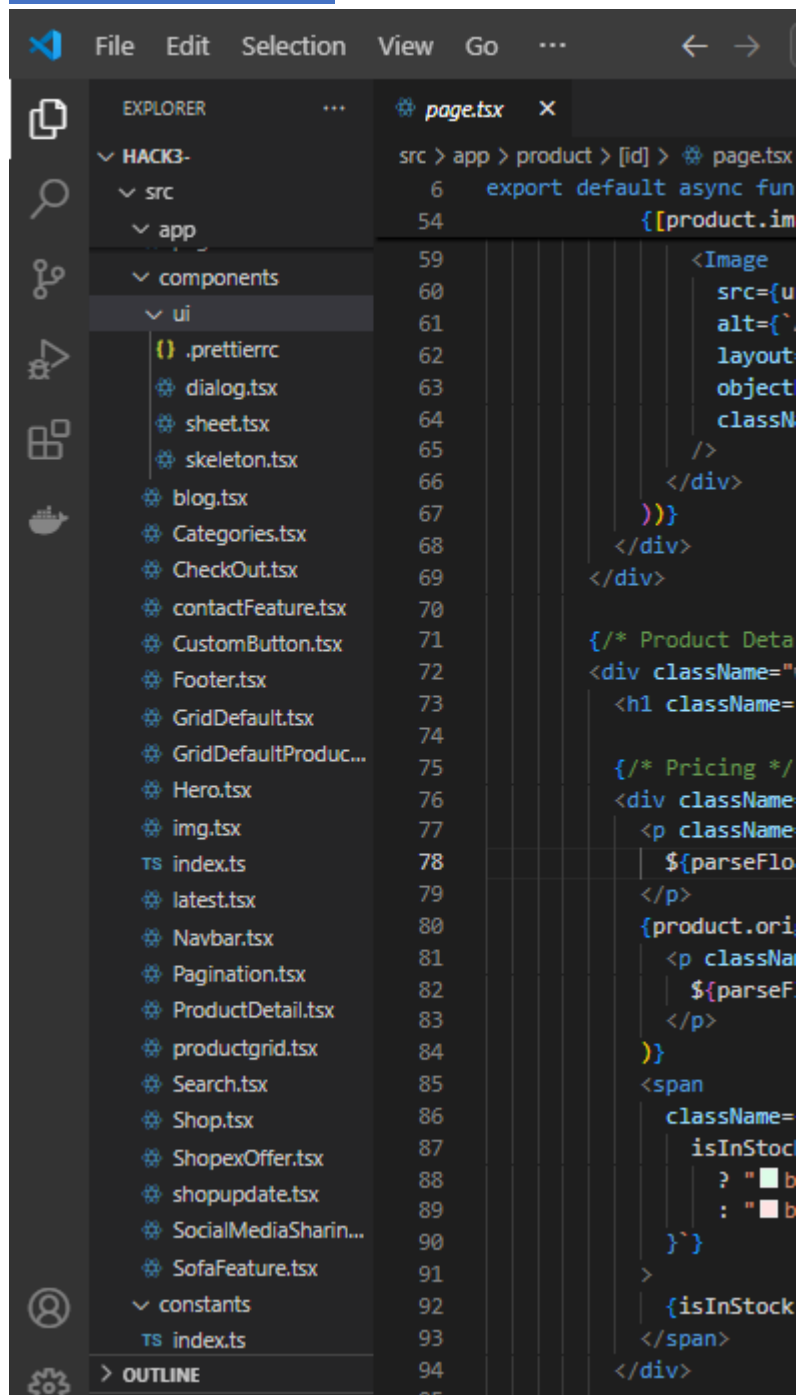
```
src > app > product > [id] > page.tsx > ProductDetail > product
1 import { client } from "@lib/sanityClient";
2 import { urlForImage } from "@sanity/lib/image";
3 import { notFound } from "next/navigation";
4 import Image from "next/image";
5
6 export default async function ProductDetail({
7   params,
8 }: {
9   params: Promise<{ id: string }>;
10 }) {
11   const id = (await params).id;
12
13   // Fetch the product data from Sanity based on the id
14   const query = `*[_type == "product" && _id == $id][0]`;
15
16   const product = await client.fetch(query, { id });
17
18   if (!product) {
19     notFound(); // Show 404 if the product doesn't exist
20   }
21
22   const isInStock = product.stockLevel > 0;
23
24   return (
25     <div className="">
26       /* Breadcrumb */
```



```
src > components > Checkout.tsx > Checkout > handlePlaceOrder
11 export default function Checkout() {
12   const handlePlaceOrder = async () => {
13     const orderData = {
14       cartItems: cartItems.map((item) => ({
15         _type: "reference",
16         _ref: item._id,
17       })),
18       total: total,
19       discount: discount,
20       orderDate: new Date().toISOString(),
21     };
22
23     try {
24       await client.create(orderData);
25       localStorage.removeItem("appliedDiscount");
26     } catch (error) {
27       console.error("Failed to create order", error);
28     }
29   };
30
31   return (
32     <div className="min-h-screen bg-gray-50 flex flex-col items-center justify-center p-6">
33       <div className="max-w-4xl w-full bg-white shadow-lg rounded-lg p-8 mt-6">
34         <h2 className="text-2xl font-bold text-gray-700 mb-6">Checkout</h2>
35
36       </div>
37     </div>
38   );
39 }
```

Day4 – Dynamic Fronted Components – [General E-commerce]

Components



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays a file tree for a project named 'HACK3-'. The tree structure is as follows:

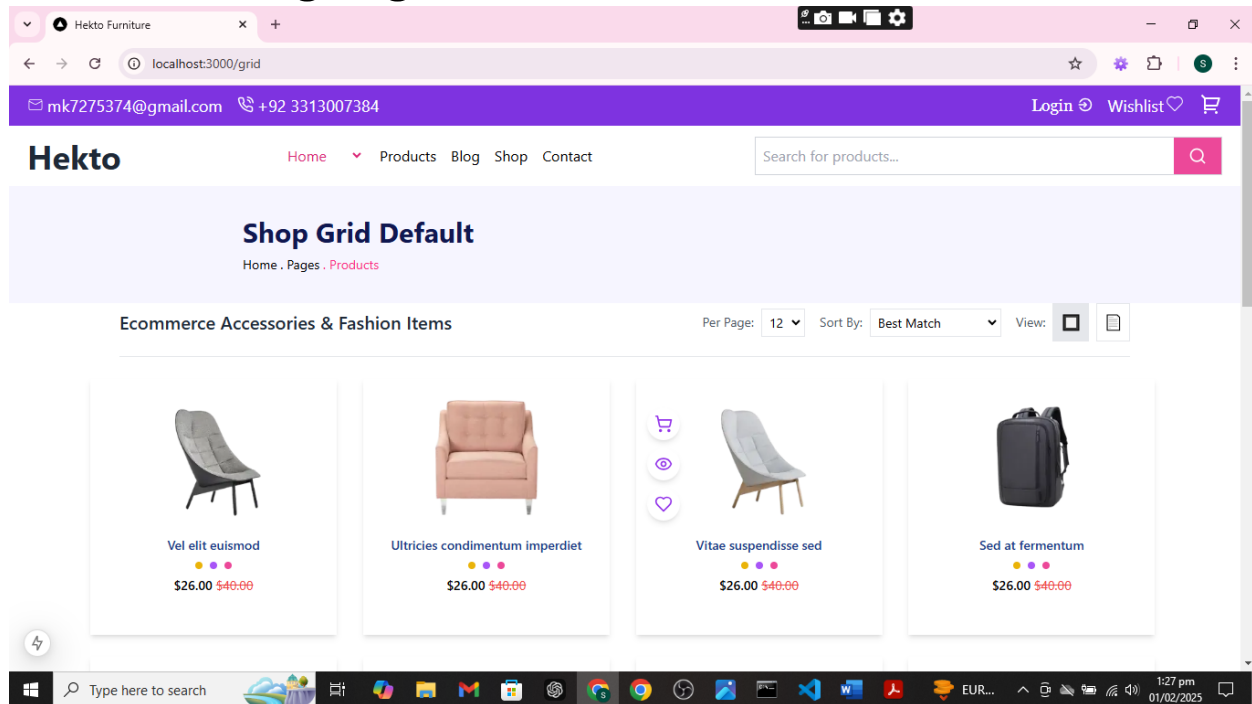
- src
 - app
 - components
 - ui (selected)

Below the 'ui' folder, a list of files is shown, including .prettierrc, dialog.tsx, sheet.tsx, skeleton.tsx, blog.tsx, Categories.tsx, CheckOut.tsx, contactFeature.tsx, CustomButton.tsx, Footer.tsx, GridDefault.tsx, GridDefaultProduc..., Hero.tsx, img.tsx, index.ts, latest.tsx, Navbar.tsx, Pagination.tsx, ProductDetail.tsx, productgrid.tsx, Search.tsx, Shop.tsx, ShopexOffer.tsx, shopupdate.tsx, SocialMediaSharin..., SofaFeature.tsx, constants, and index.ts.

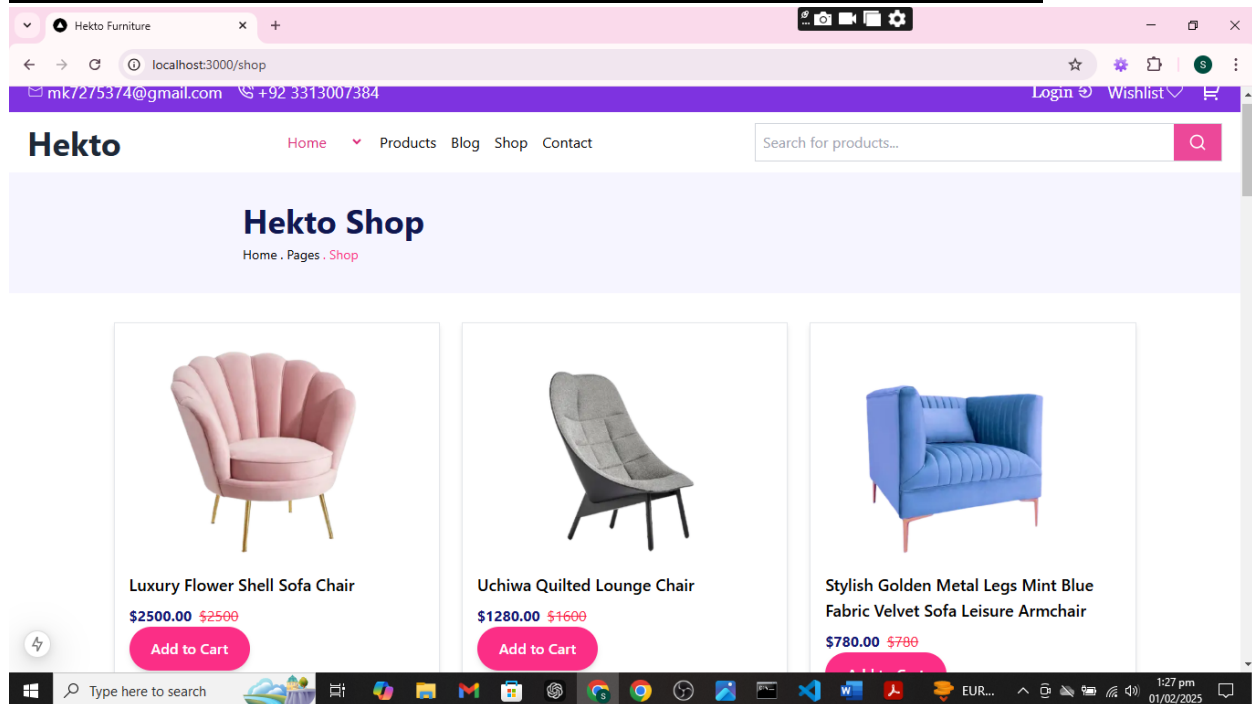
The main editor area on the right shows the content of 'page.tsx'. The code is a TypeScript file that exports a default asynchronous function. It includes JSX elements for rendering a product detail page, such as an image, a heading, and a pricing section. The code uses props like 'product' and 'isInStock' and includes conditional rendering logic for the 'isInStock' status.

```
src > app > product > [id] > page.tsx
6   export default async fun
54   {[product.im
59
60   <Image
61     src={u
62     alt={`
63     layout
64     object
65     classN
66   </div>
67   )}}
68   </div>
69   </div>
70
71   /* Product Deta
72   <div className="
73     <h1 className=
74
75   /* Pricing */
76   <div className
77     <p className
78       ${parseFlo
79   </p>
80   {product.ori
81     <p classNa
82       ${parseF
83     </p>
84   )}}
85   <span
86     className=
87     isInStoc
88     ? "■ b
89     : "■ b
90   }`}}
91   >
92   {isInStock
93   </span>
94   </div>
95
```

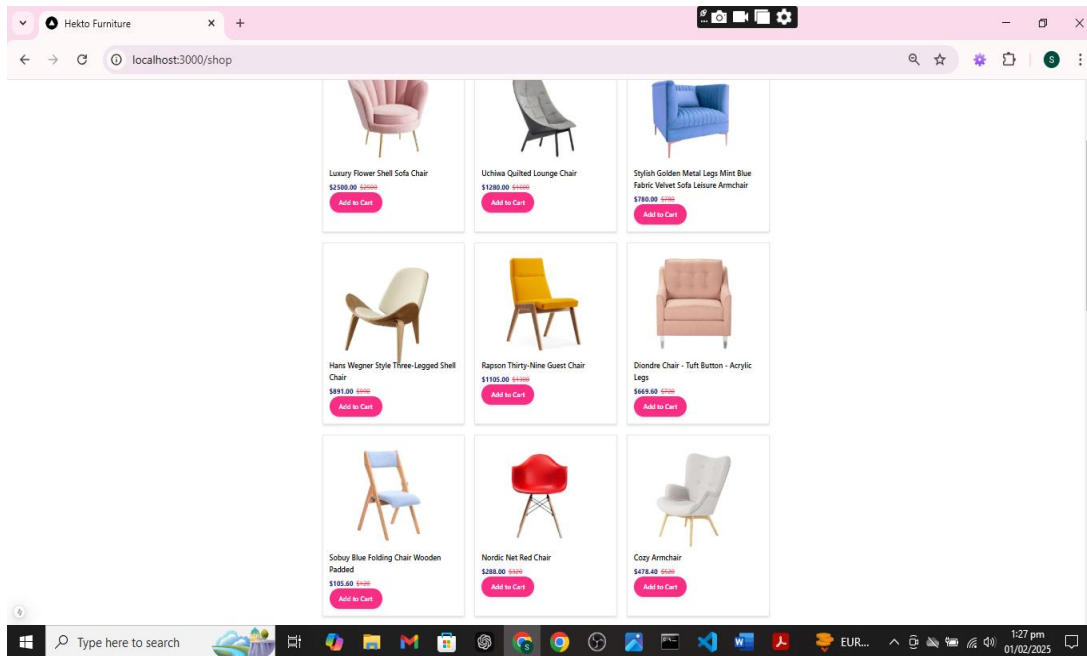
Day4 – Dynamic Fronted Components – [General E-commerce] Product listing Pages



Hekto Shop with dymnamic data fetched from sanity_
Users can view detail info by clicking on product

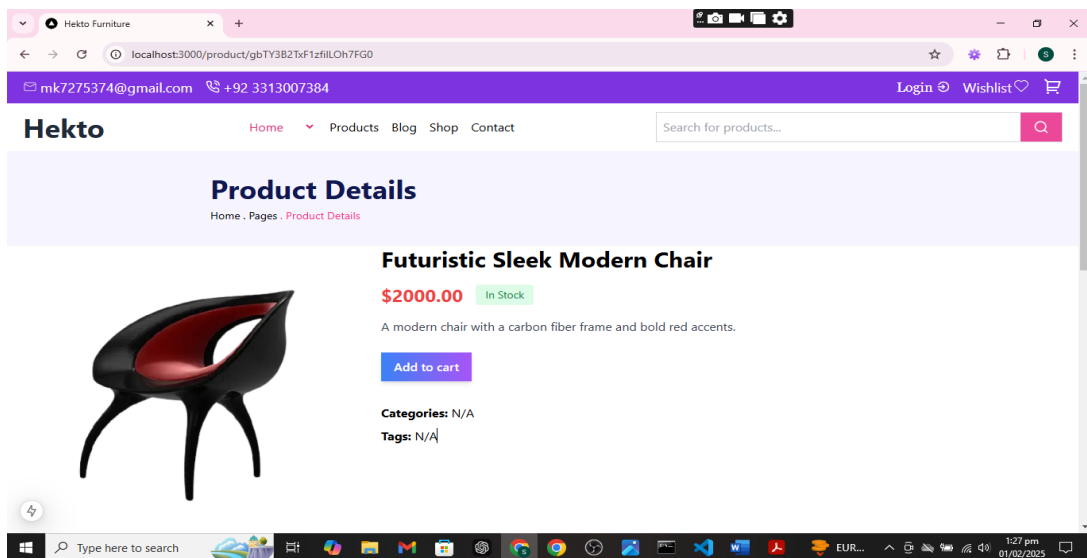


Day4 – Dynamic Fronted Components – [General E-commerce]



Product Detail Page

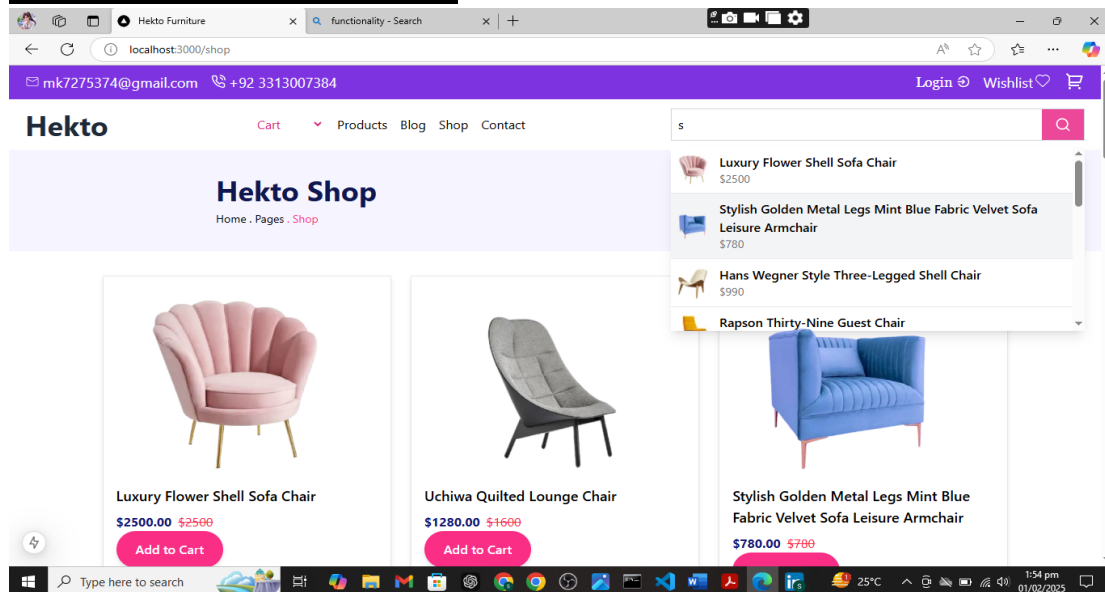
Product Detail Page dynamically loads specific product info such as name, price, description and image.



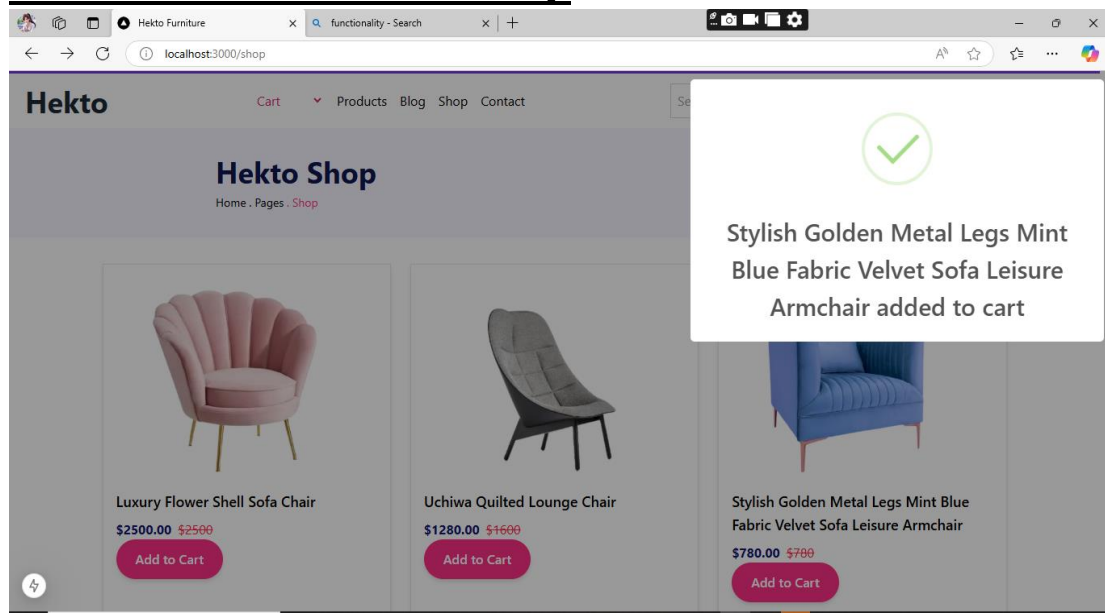
Day4 – Dynamic Fronted Components – [General E-commerce]

Implement Search and Add to cart functionality

Search Functionality



Add To Cart Functionality

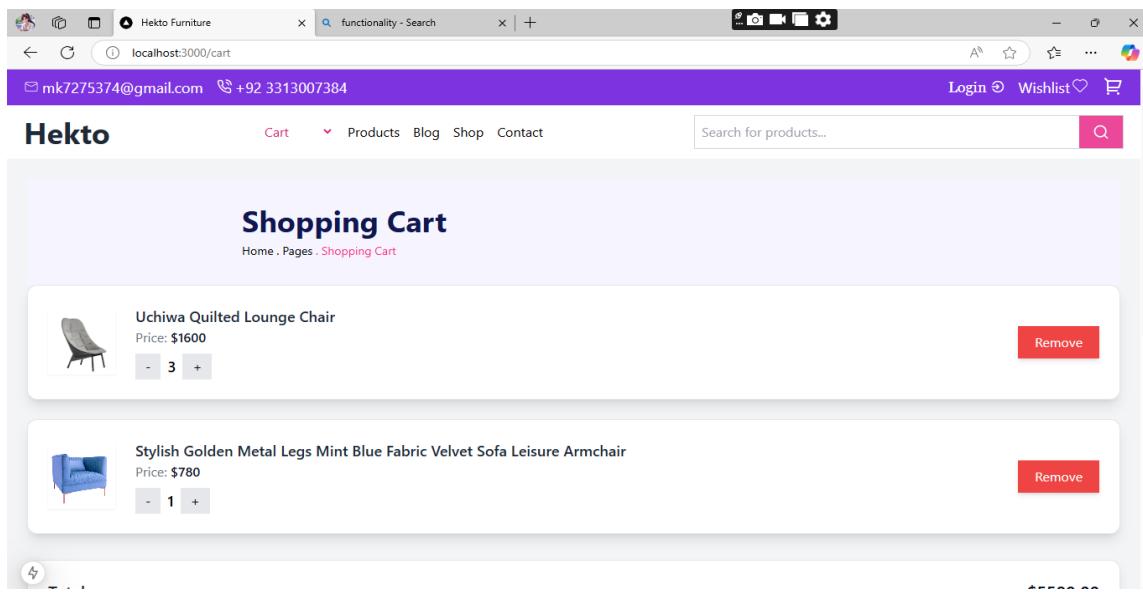


Day4 – Dynamic Fronted Components – [General E-commerce]

After clicking any product add to cart button product will added to cart then we can proceed it to checkout-

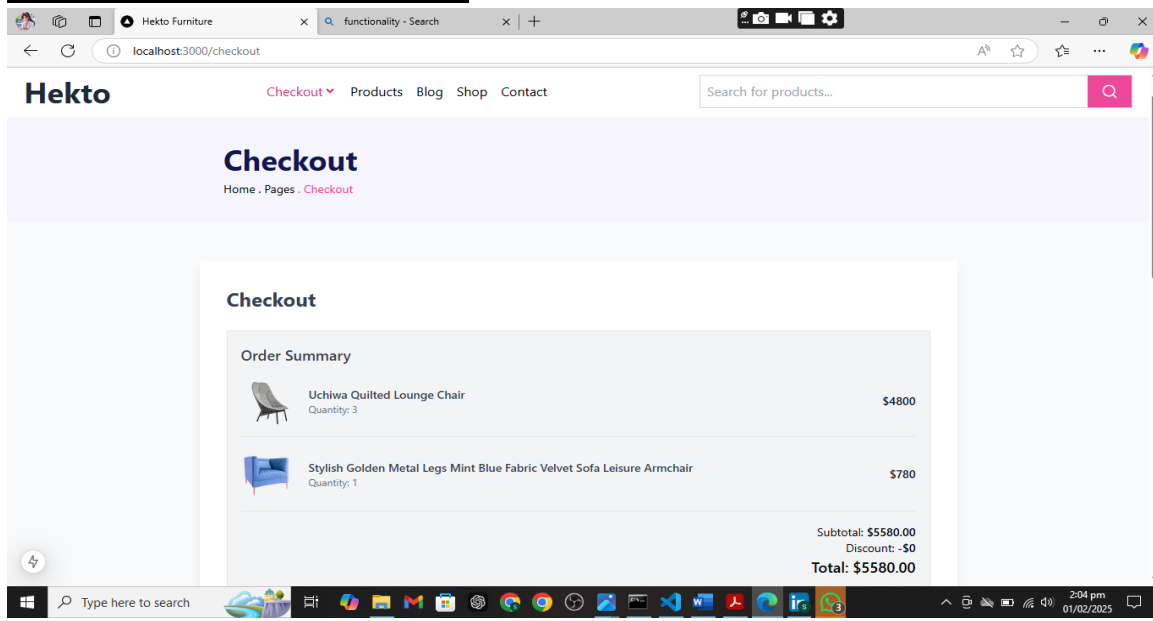
User can see total price of all items in their cart and also remove any item he wants...

Users can proceed to checkout page after reviewing the cart details.....



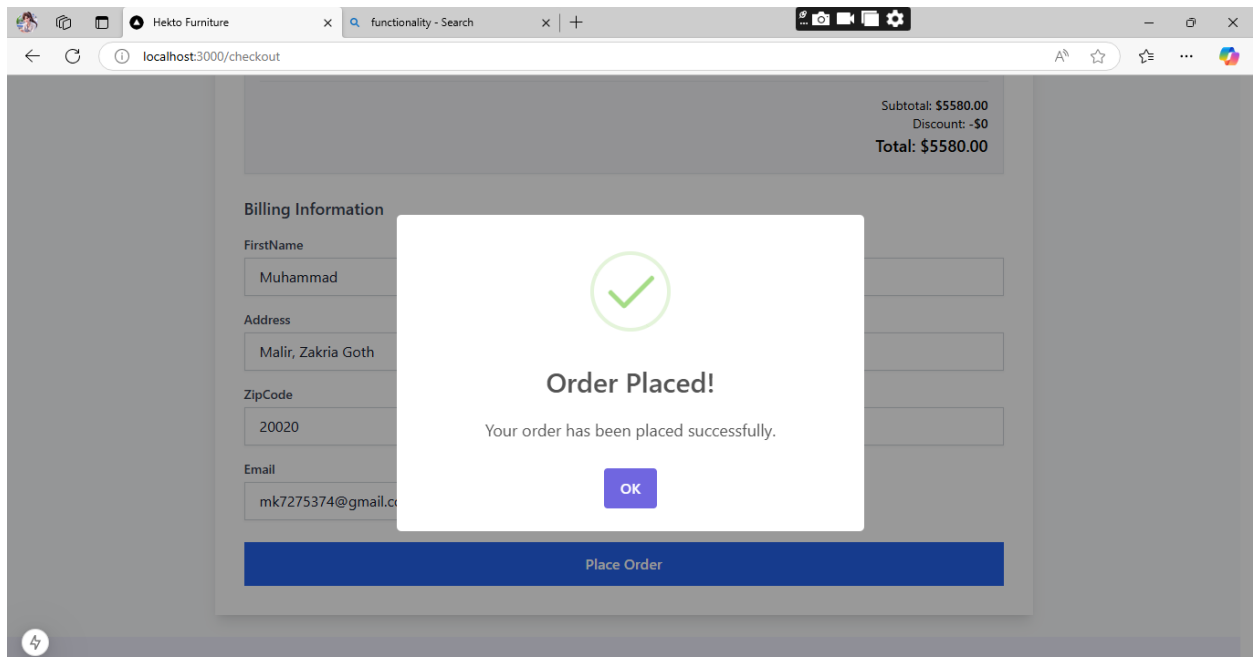
Day4 – Dynamic Fronted Components – [General E-commerce]

Proceed to checkout.



**Order Placed !! and also stored information of user in
sanity studio**

Day4 – Dynamic Fronted Components – [General E-commerce]



Self Validation Check list:

Frontend Component Development:



Documentation and Submission



Styling and Responsiveness



Code Quality



Final Review

