

# **Day-04 (Dynamic Frontend Components - Bandage)**

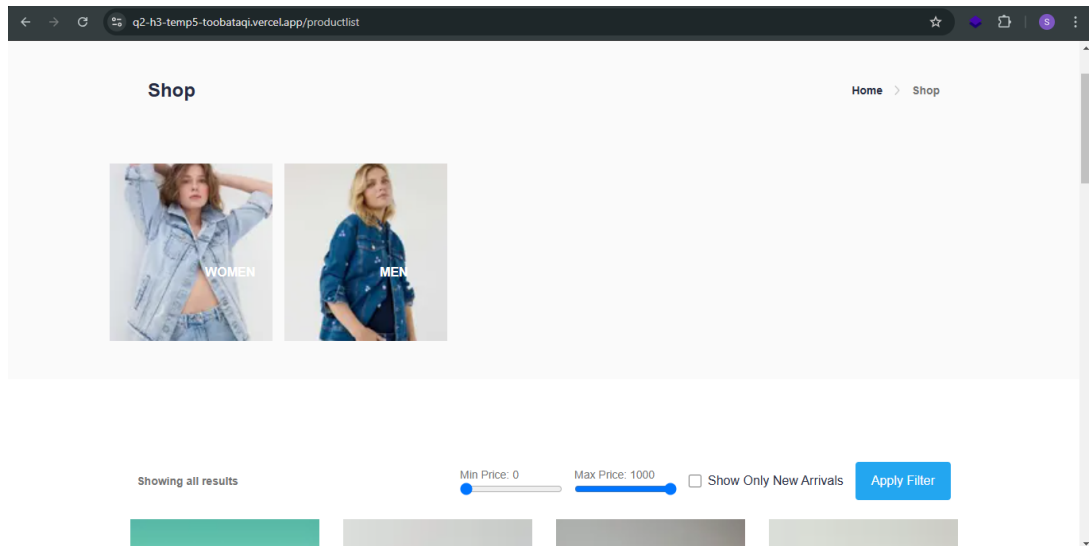
**Name: Tooba Taqi**

**Seat no. : 59935**

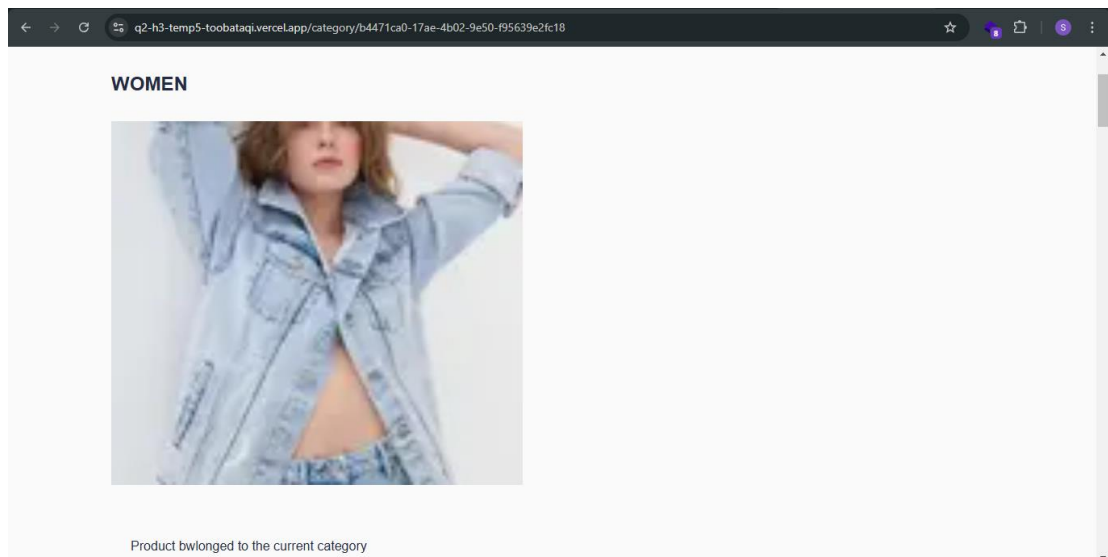
**Class Timing: Saturday Morning (9am – 12pm)**

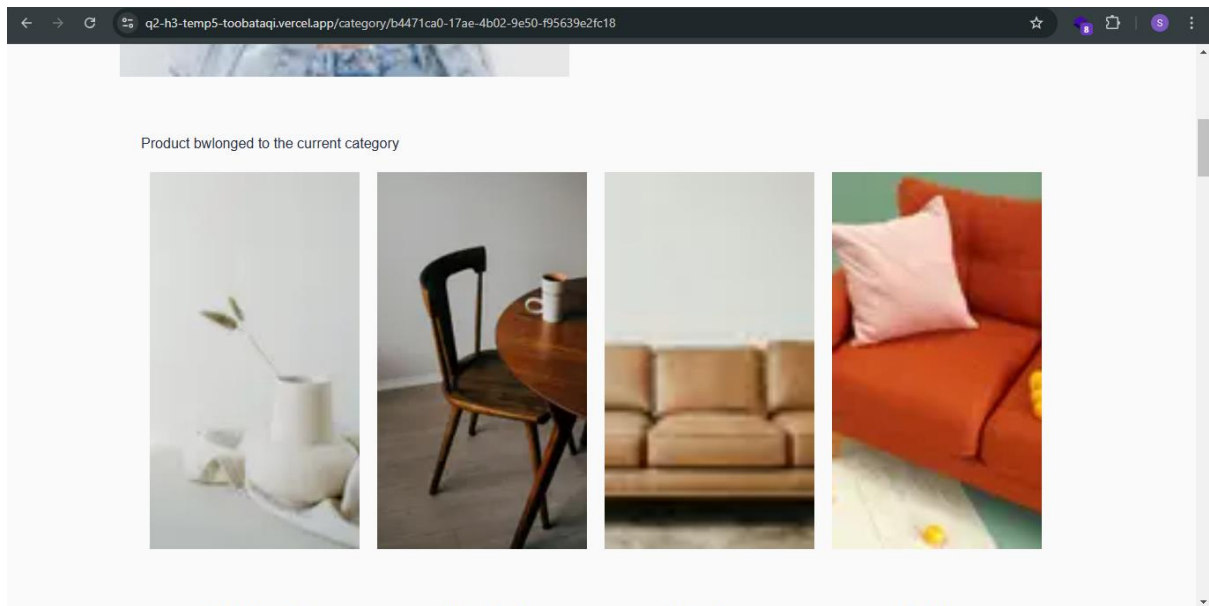
# Functional Deliverables:

- /productlist page serves as the primary page of the marketplace website:
  - First section of this page has categories data (dynamically). On clicking on the category card, it'll lead us to the dynamic page of that specific category “/category/[id]/page.tsx” and show the products related to that category.



- The dynamic page of the selected category:

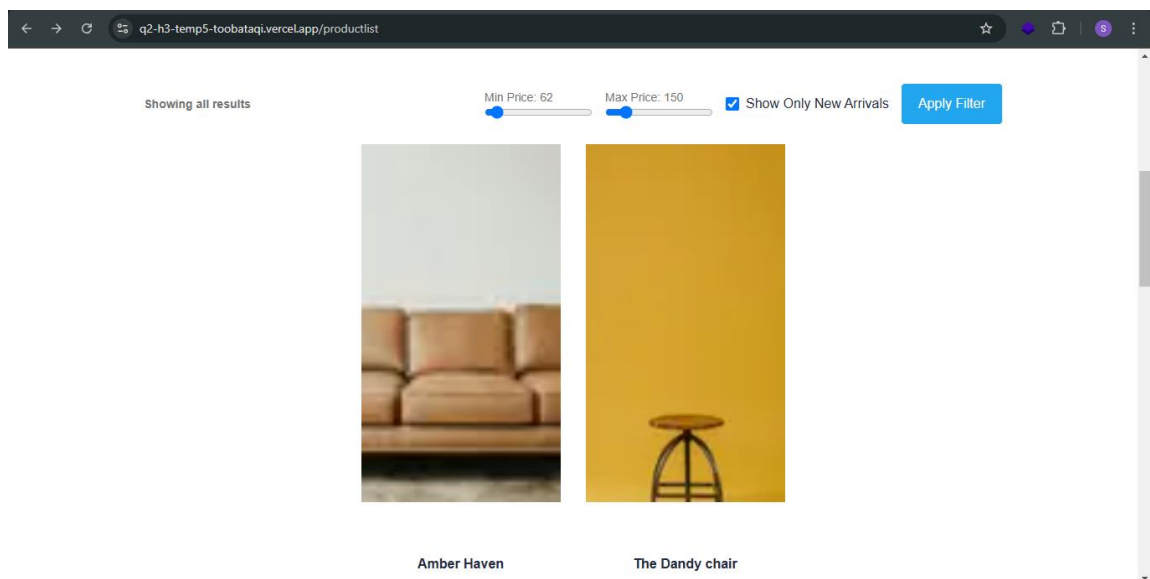




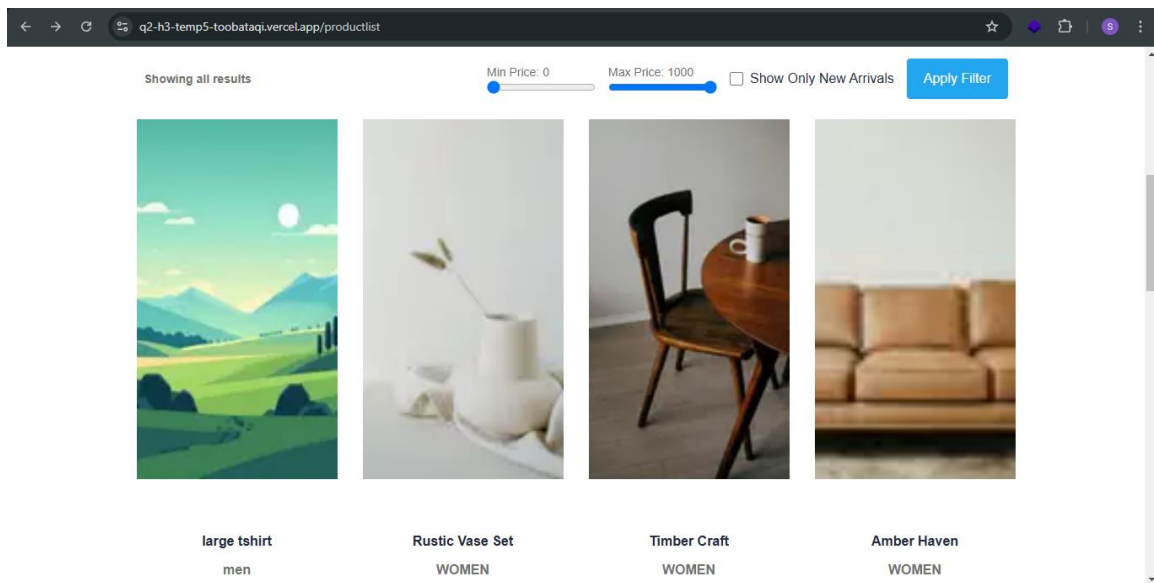
- The /productlist page have dynamically fetched products and also have a filter. The Filter includes price range slider and new arrival checkbox.

**Proof that the filter is working:**

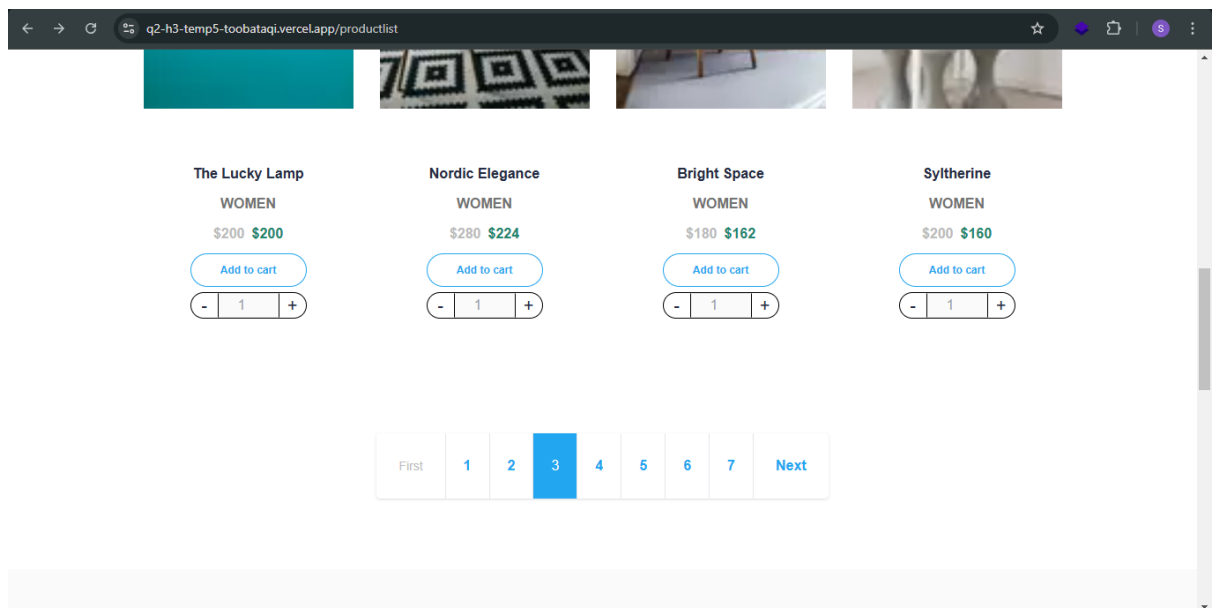
- **Products wit filter applied:**



- **Products with no filter applied:**



- Pagination functionality on `/productlist` page will show only 4 products at a time, and will divide all the products (either they are filtered or not) into multiple chunks, and then it'll make the required numbers of buttons automatically:



- [illegible]

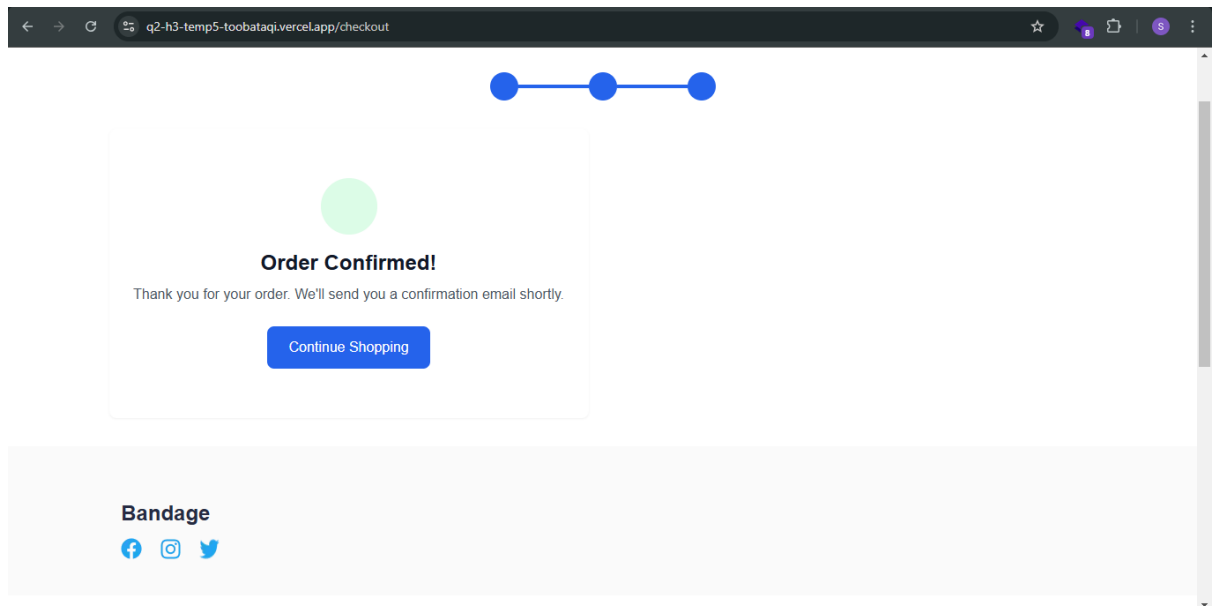
- A screenshot of a web browser displaying a shopping cart. The browser's address bar shows the URL 'q2-h3-temp5-toobataqi.vercelApp/cart'. The page title is 'Your Cart has (1) items'. The cart contains one item: 'large tshirt' with a quantity of 1, priced at 13/-, for a total of 24.04. A 'Delete' link is next to the item. A blue 'Continue Shopping' button is located below the cart. An 'Order Summary' box at the bottom right shows the following details: Subtotal: \$13.00, Shipping: \$10.00, Tax: \$1.04, and Total: \$24.04. A blue 'Proceed to Checkout' button is at the bottom of the summary box.

- “Proceed to Checkout” will lead the user to the /checkout page and after filling the given form, it’ll take us to the Shipment details component while still showing our order details:

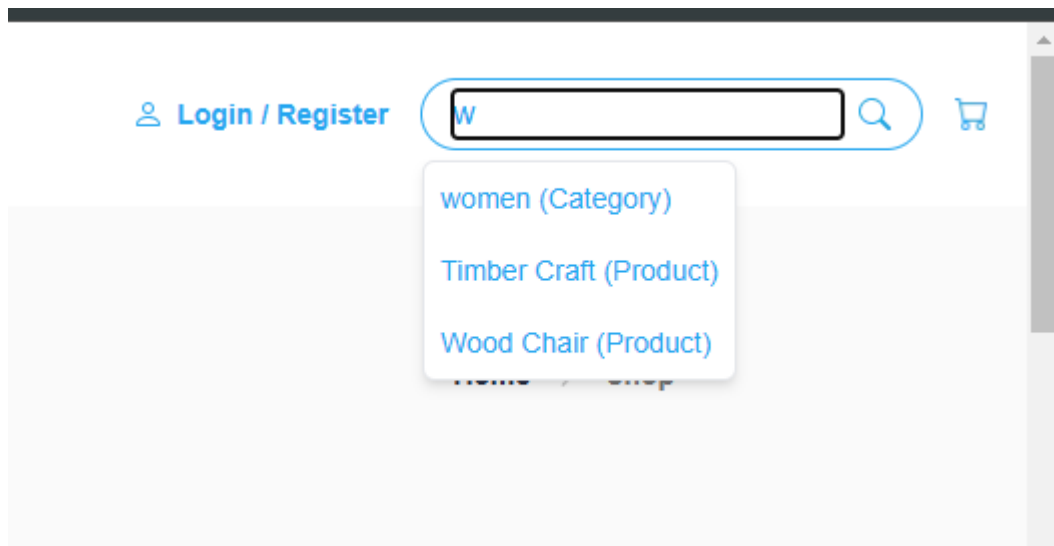
The screenshot shows a web browser at the URL `q2-h3-temp5-toobataqi.vercel.app/checkout`. At the top, there is a progress bar with three circles; the first circle is blue, indicating the current step. The main content area is divided into two columns. The left column, titled "Shipping Information", contains several form fields: "First Name" (value: "as"), "Last Name" (value: "sdsd"), "Email" (value: "abc@gmail.com"), "Address" (value: "asas"), "City" (value: "dsdsd"), "State" (value: "dfdfdf"), "ZIP Code" (value: "23233"), and "Country" (value: "Pak"). A blue button labeled "Continue to Payment" is at the bottom of this column. The right column, titled "Order Summary", displays a table of items and their prices.

Order Summary	
large tshirt x 1	\$13.00
Subtotal	\$13.00
Shipping	\$10.00
Tax	\$1.04
<b>Total</b>	<b>\$24.04</b>

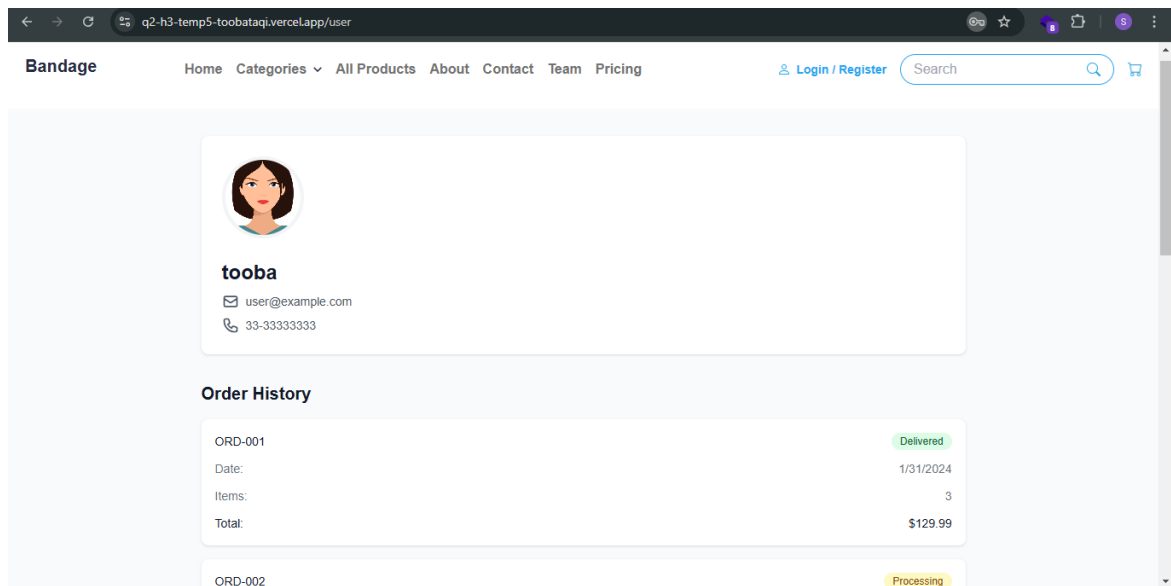
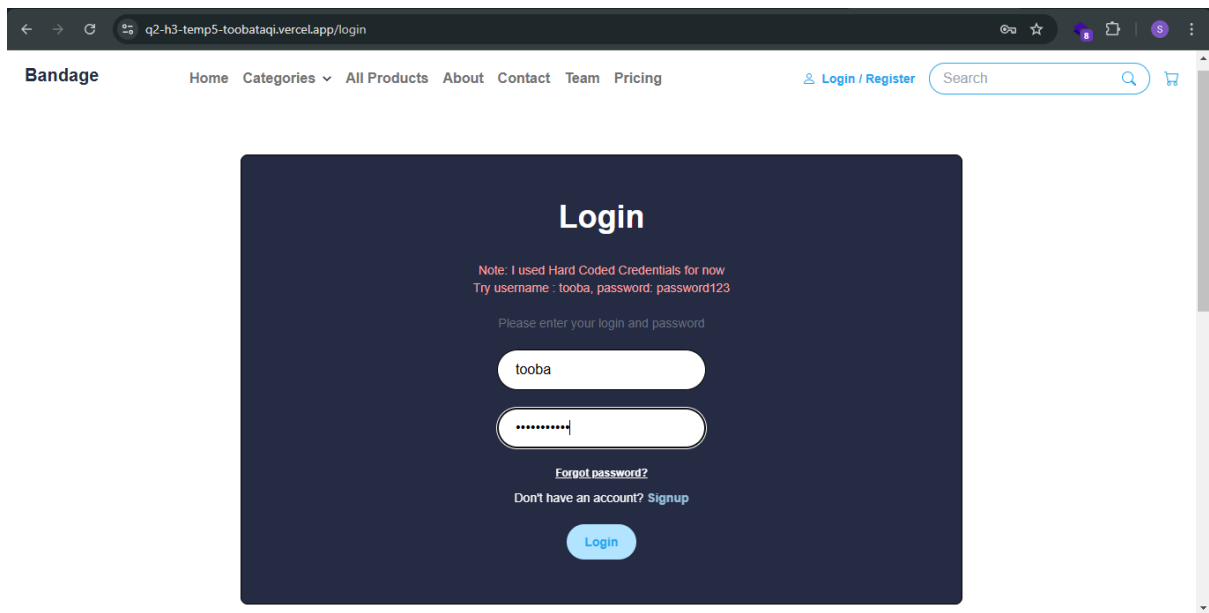
This screenshot shows the same browser window, but the progress bar now has the second circle highlighted in blue. The "Shipping Information" column is no longer visible. The "Payment Information" column is active, containing form fields for "Card Number" (value: "424242424242424"), "Expiry Date" (value: "234"), and "CVV" (value: "123"). Below these fields are two buttons: a blue "Place Order" button and a white "Back to Shipping" button. At the bottom of the page, there is a footer section with the text "Bandage" and three social media icons for Facebook, Instagram, and Twitter.



- Search Functionality searches the products with product tags, category name, and product name and dynamically take us to the product/category page:

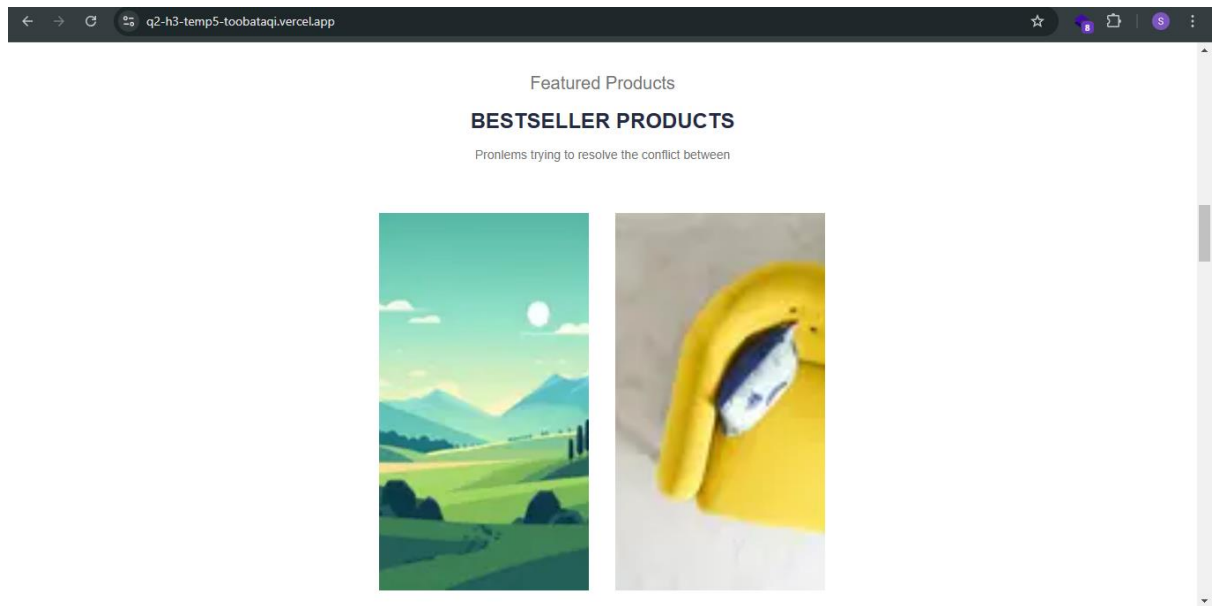


- /user page will navigate to the /login if user not logged in:



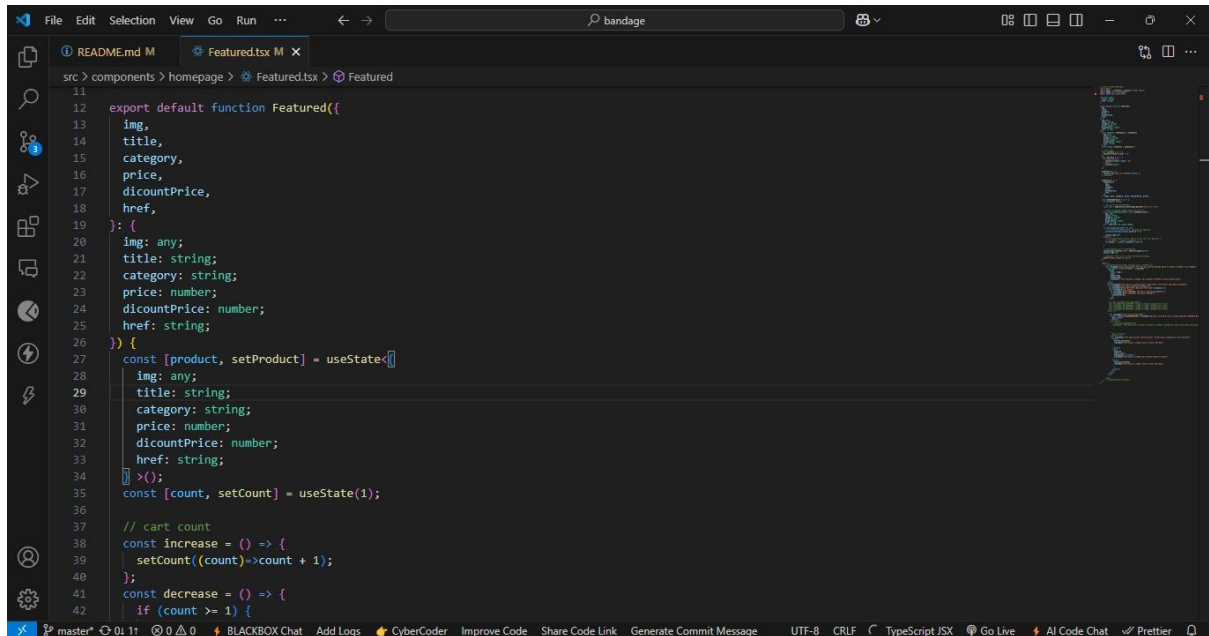


- Best Seller products shows the only products which have Best Seller Boolean Value = true in Sanity.

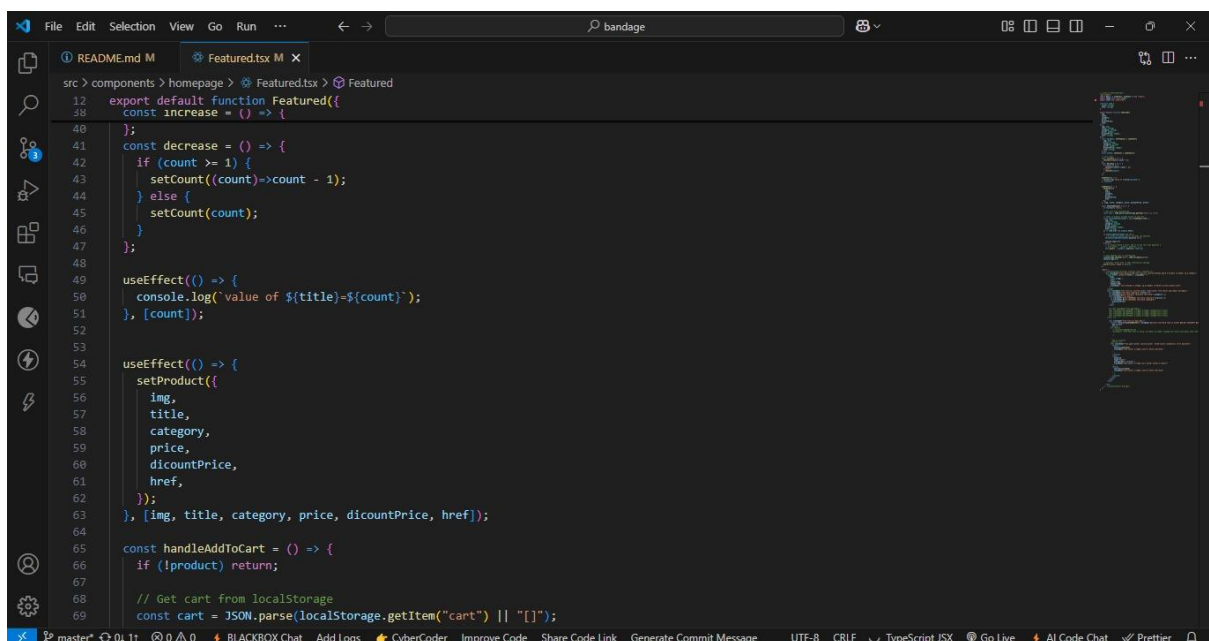


# Code Deliverables:

- The Product Card Component “Featured.tsx” i.e taking product data as **props**. This is the most important component across my entire webstore:



```
11 export default function Featured({
12   img,
13   title,
14   category,
15   price,
16   discountPrice,
17   href,
18 }): {
19   img: any;
20   title: string;
21   category: string;
22   price: number;
23   discountPrice: number;
24   href: string;
25 } {
26   const [product, setProduct] = useState<{
27     img: any;
28     title: string;
29     category: string;
30     price: number;
31     discountPrice: number;
32     href: string;
33   }>({});
34   const [count, setCount] = useState(1);
35
36   // cart count
37   const increase = () => {
38     setCount((count) => count + 1);
39   };
40   const decrease = () => {
41     if (count >= 1) {
42
```



```
43     setCount((count) => count - 1);
44   } else {
45     setCount(count);
46   }
47 };
48
49 useEffect(() => {
50   console.log('value of ${title}-${count}');
51 }, [count]);
52
53
54 useEffect(() => {
55   setProduct({
56     img,
57     title,
58     category,
59     price,
60     discountPrice,
61     href,
62   });
63 }, [img, title, category, price, discountPrice, href]);
64
65 const handleAddToCart = () => {
66   if (!product) return;
67
68   // Get cart from localStorage
69   const cart = JSON.parse(localStorage.getItem("cart") || "[]");
```

```
File Edit Selection View Go Run ... bandage
src > components > homepage > Featured.tsx > Featured > handleAddToCart
12 export default function Featured({
65 const handleAddToCart = () => {
67 //
68 // Get cart from localStorage
69 const cart = JSON.parse(localStorage.getItem("cart") || "[]");
70
71 // Check if product already exists in the cart
72 const existingProductIndex = cart.findIndex((item: {
73   img: any;
74   title: string;
75   category: string;
76   price: number;
77   discountPrice: number;
78   href: string;
79 }) => item.href === product.href);
80
81 if (existingProductIndex !== -1) {
82   // If product already exists, increase the quantity
83   cart[existingProductIndex].quantity += 1;
84
85   console.log(cart);
86 } else {
87   cart.push({ ...product, quantity: count });
88
89   // Save updated cart to localStorage
90   localStorage.setItem("cart", JSON.stringify(cart));
91   console.log(cart);
92
93   // Optional: Alert user on show confirmation message
94   alert("Product added to cart");
95 }
96
```

```
File Edit Selection View Go Run ... bandage
src > components > homepage > Featured.tsx > Featured
12 export default function Featured({
98
99 return [
100   <div className="flex flex-col justify-between gap-8 h-[715px] w-[328px] lg:w-[238px]">
101     <Link href={`/product/${href}`} className="">
102       <Image
103         src={`/${img}`}
104         alt=""
105         width={100}
106         height={100}
107         className="min-h-[427px] w-[328px] lg:w-[238px] h-[427px] mx-auto object-cover"
108       />
109     </Link>
110     <div className="flex flex-col justify-center items-center text-center gap-[10px] mb-[50px]">
111       <h3 className="font-bold leading-6">{title}</h3>
112       <p className="text-[#737373] leading-6 font-bold">{category}</p>
113       <div className="flex gap-2">
114         <p className="text-[#BDBDBD] font-bold leading-6">{price}</p>
115         <p className="text-[#23856D] font-bold leading-6">{discountPrice}</p>
116       </div>
117       <div className="flex flex-col gap-[6px]">
118         <button
119           onClick={handleAddToCart}
120           className="bg-white font-bold text-xs border border-[#23A6F0] text-[#23A6F0] py-[10px] px-[20px] rounded-full hover:text-whi"
121         >
122           Add to cart
123         </button>
124       </div>
125       <div className="flex gap-0 border justify-center border-black rounded-2xl w-fit bg-white">
126         <button
127           onClick={decrease}
128
```

- **Logic to Integrate API:**

- Migrate the data of given api into sanity by using the given code snippet
- Then installed the sanity in my project and fetched in an api folder to optimise loading of data. Can check the api data I migrated into Sanity: “/api/product” and “/api/category”

```
1 // src/app/api/product/route.ts
2 import { NextResponse } from 'next/server';
3 import { client } from '@sanity/lib/client';
4
5 export async function GET() {
6   const query = `*[ _type == "product"
7     { _id, description, isNew, tags,
8       title,
9       _updatedAt,
10      productImage {
11        asset -> {
12          _id,
13          url
14        }
15      }
16    },
17    price,
18    discountPercentage,
19    category -> { title, _id,
20      bestseller }
21  `;
22
23   const products = await client.fetch(query);
24
25   return NextResponse.json(products);
26 }
```

- **Dynamic Routing:**

- Access the selected product through its id and used its id as a dynamic path because id is a unique key each product must have. And then access the id inside the component as url through params. Following line of code show it all:

```
const params = useParams();
const productId = params.id;
```

## Conclusion:

The dynamic frontend implementation was successfully implemented at day 4 with Functional components which are listed below:

- All Products page or Product list page “/productlist”
- All Categories page: “/allcategories”
- Product page “/product/:id”
- Cart functionality with local storage
- Checkout functionality