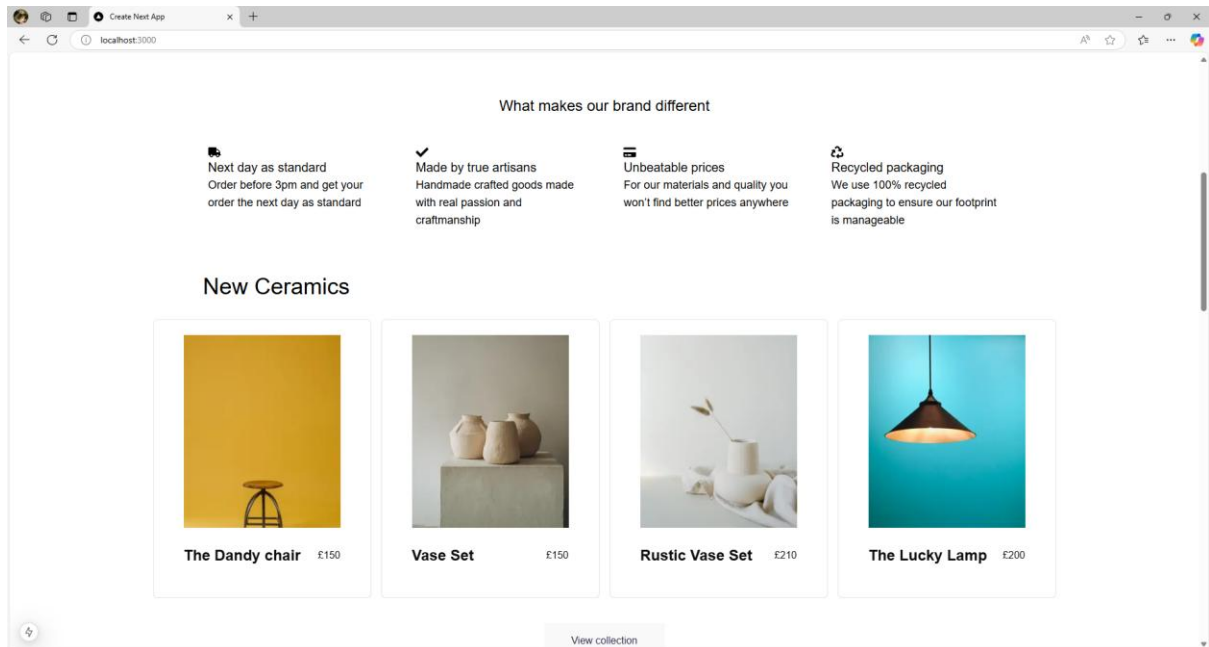


"Day 4 - Dynamic Frontend Components - [Muniba e-shop]"

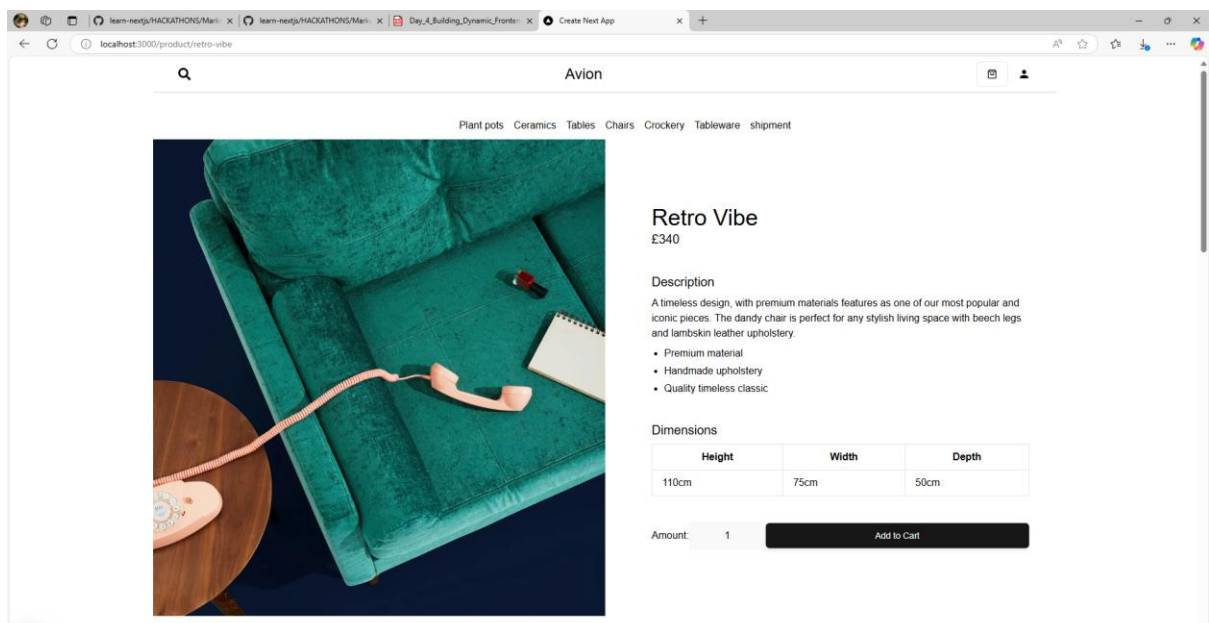
Product Listing Component:

Product List with filter of New ceramic tag.



Product Detail Page:

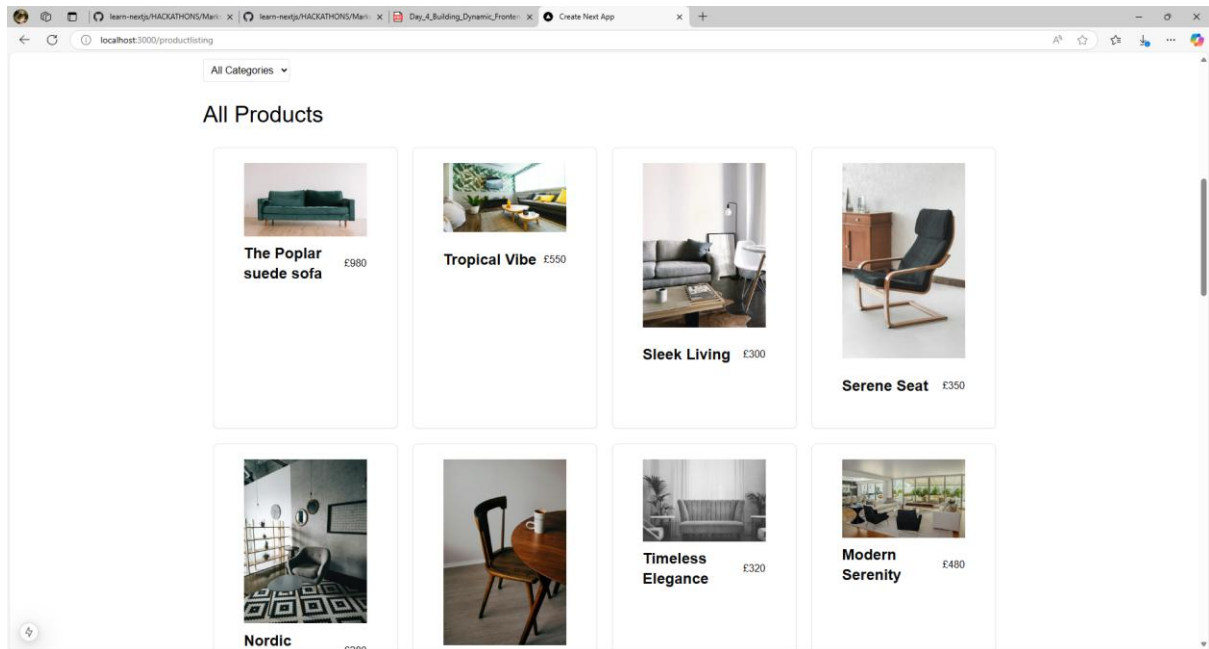
A Dynamic product detail page that is fetching data from sanity on the base of slug field of products.



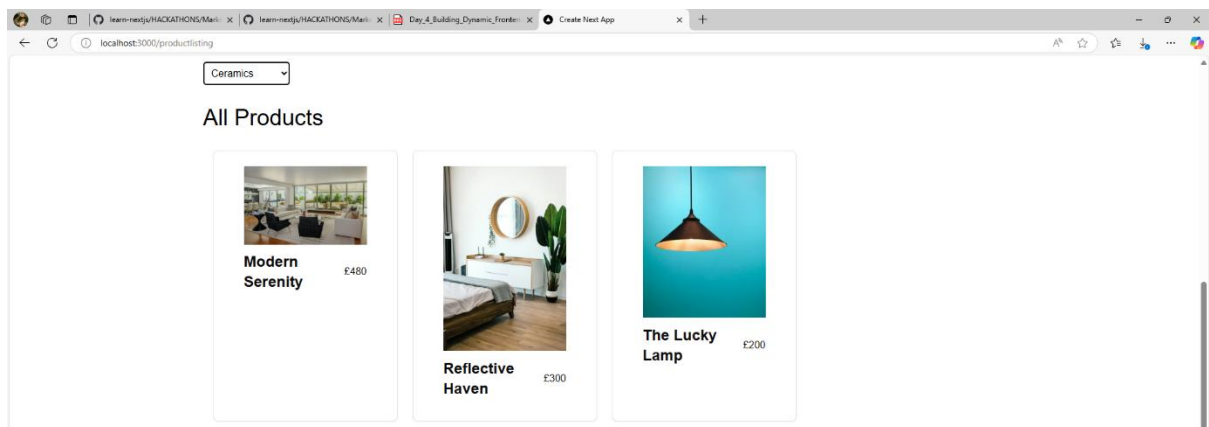
Category Component:

Category component that has a dropdown list to filter categories coming from sanity.

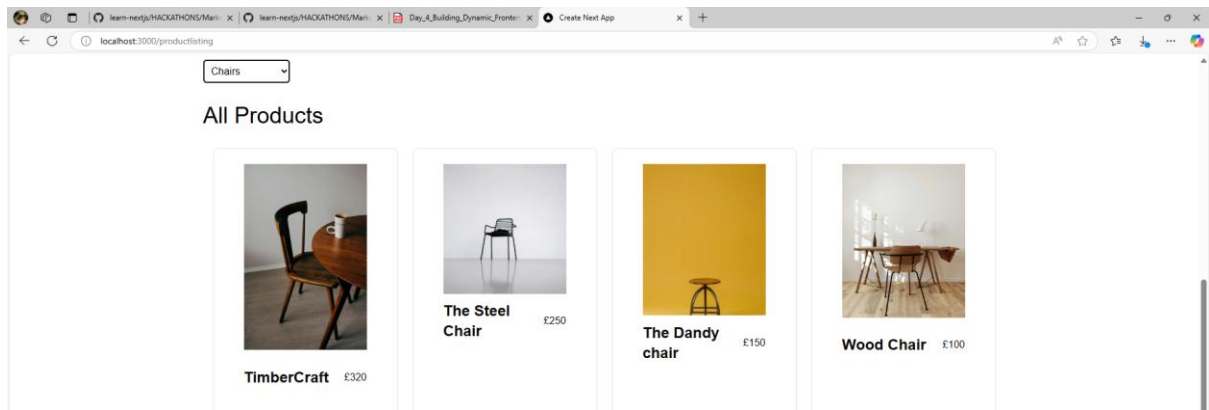
1- All products:



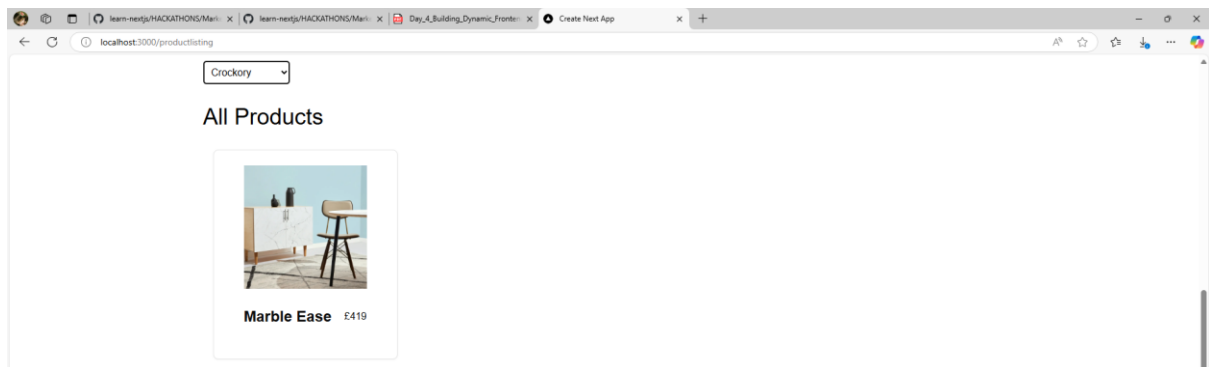
2- Ceramics



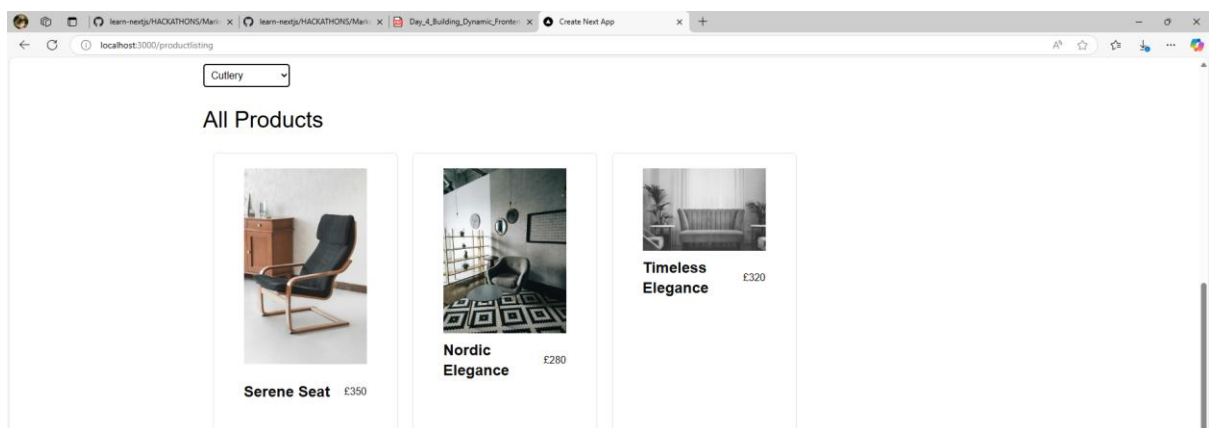
3- Chairs



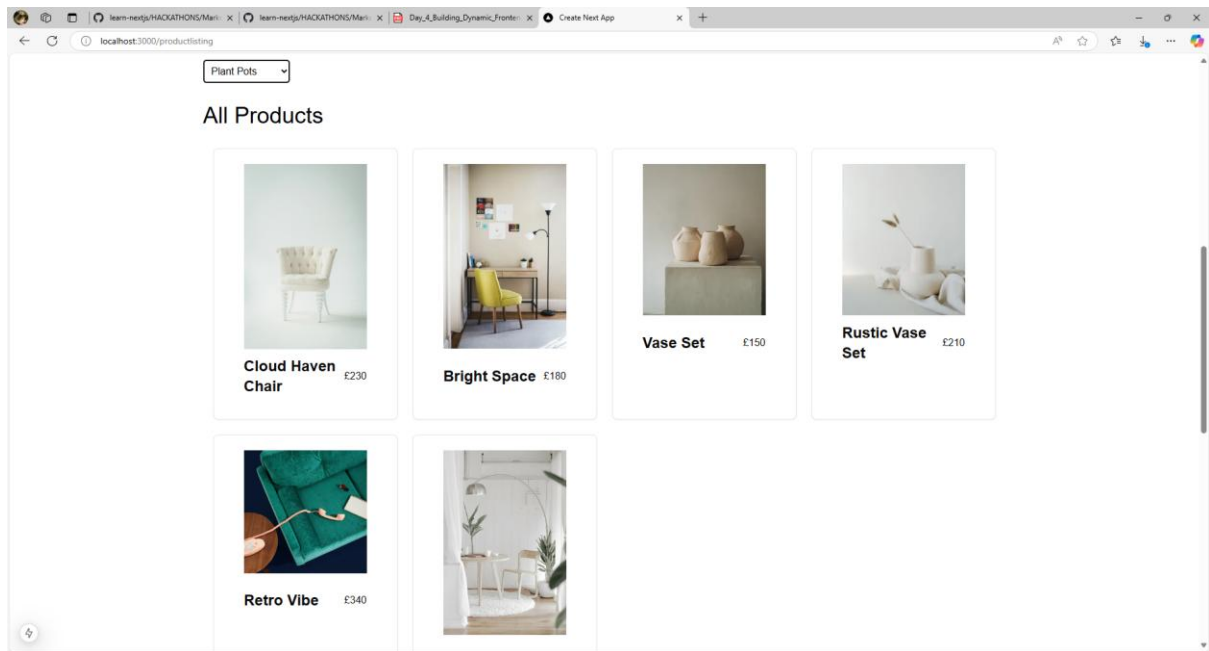
4- Crockery



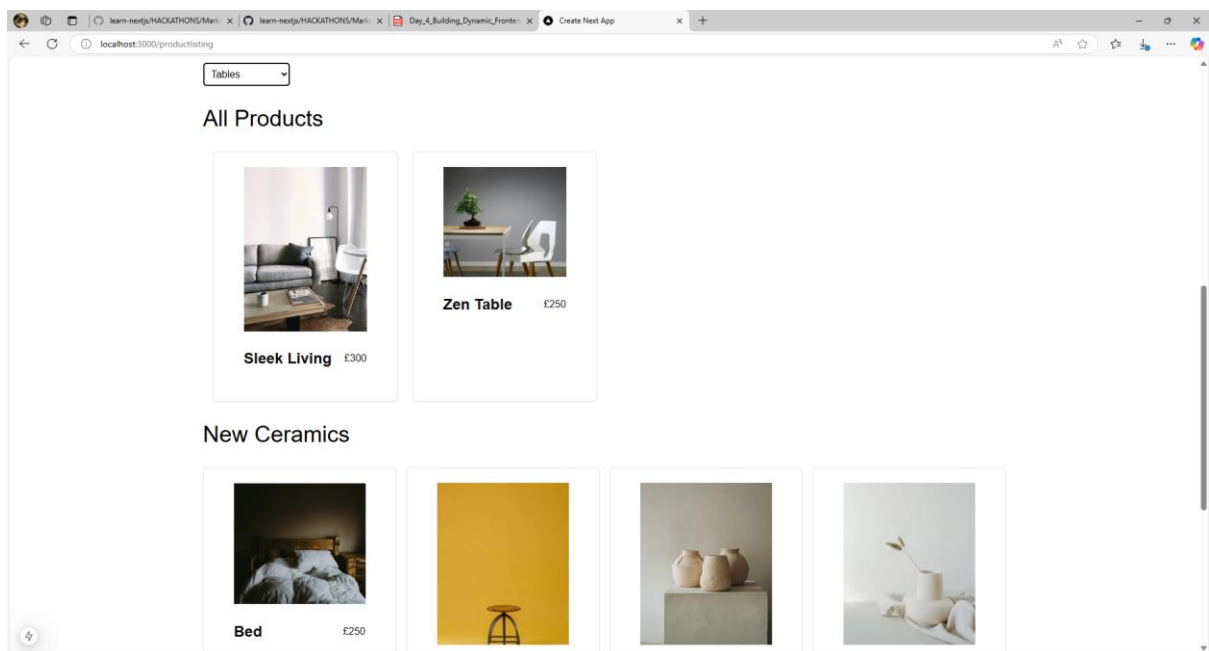
5- Cutlery



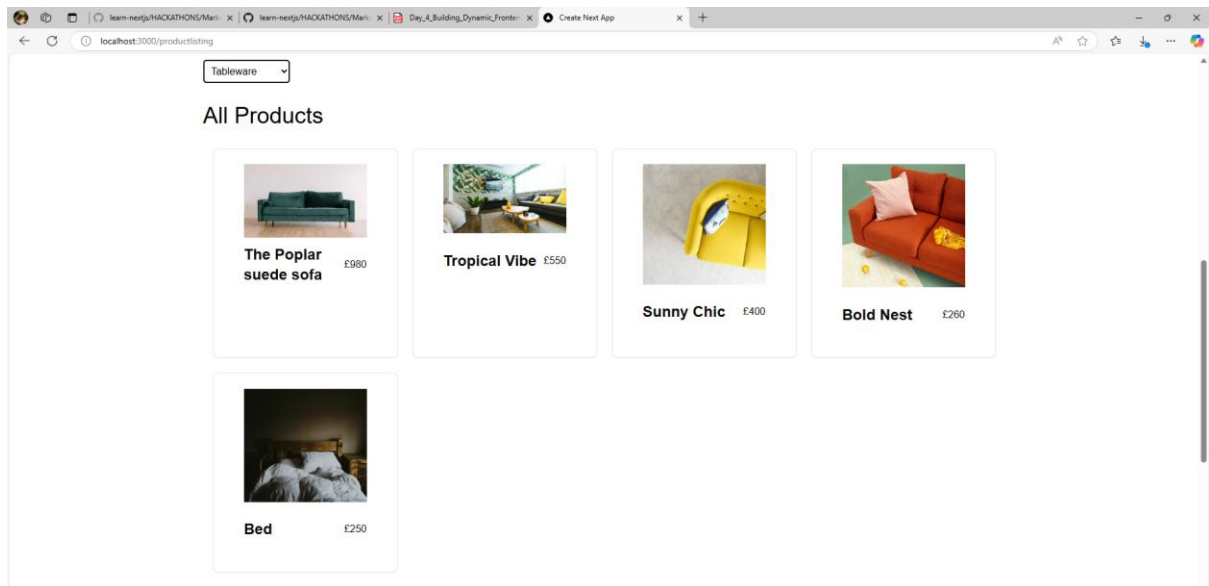
6- Plant pots



7- Tables

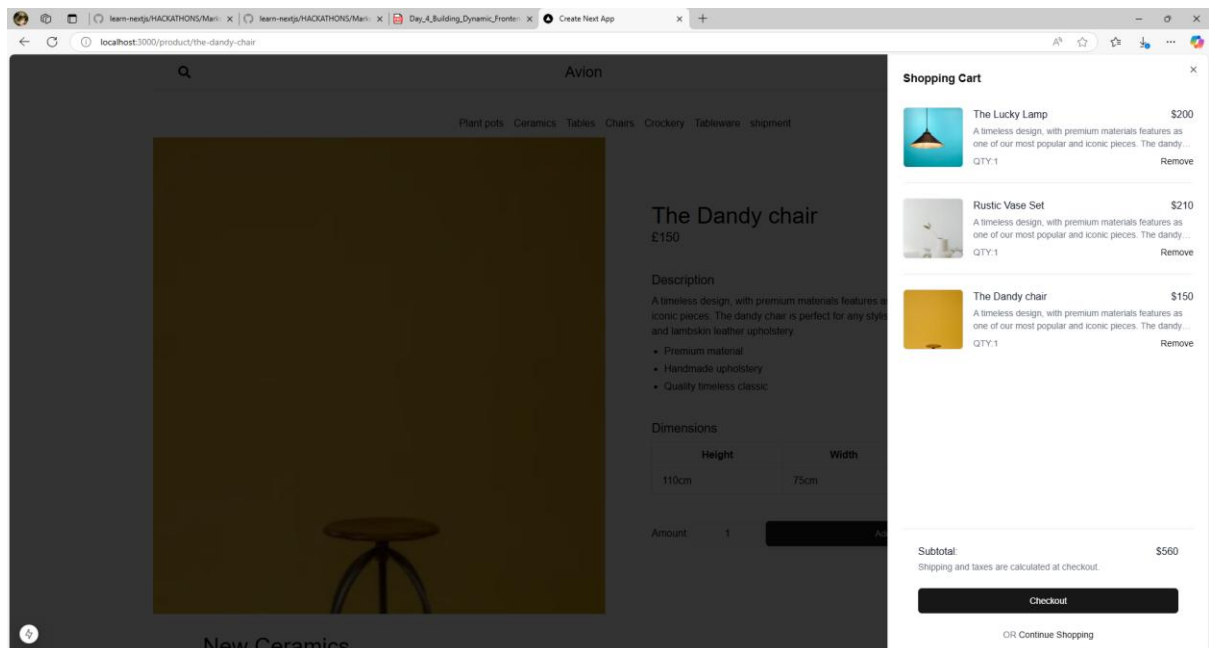


8- Tableware



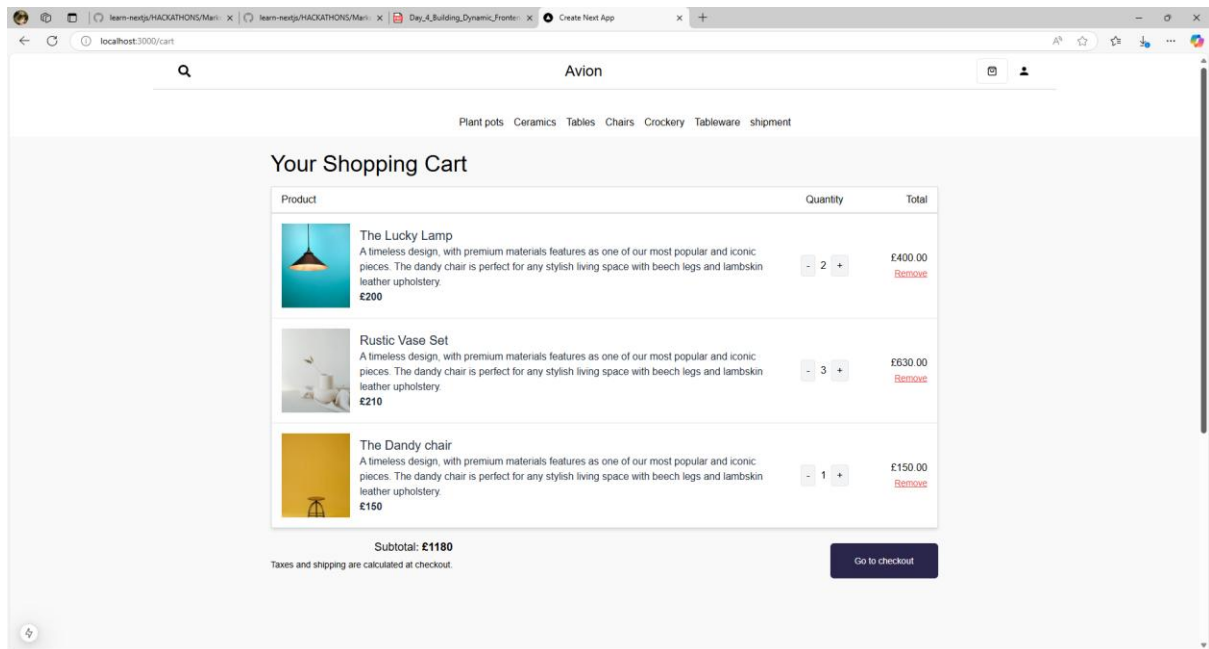
Add to Cart Component:

ShoppingCartModal Component display on right side when click add to cart or cart button on navbar design using shadcn ui sheet and shopping cart hook.



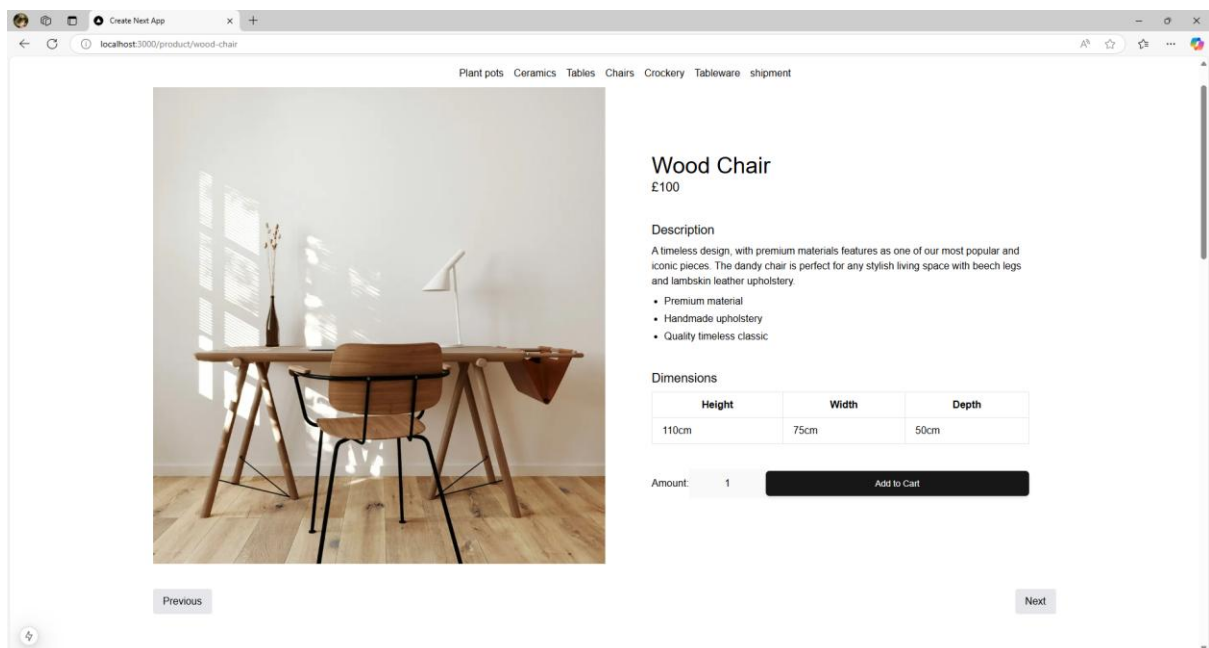
Cart Page:-

For add or subtract quantities of products that add in cart and show price of total products then subtotal of all products.

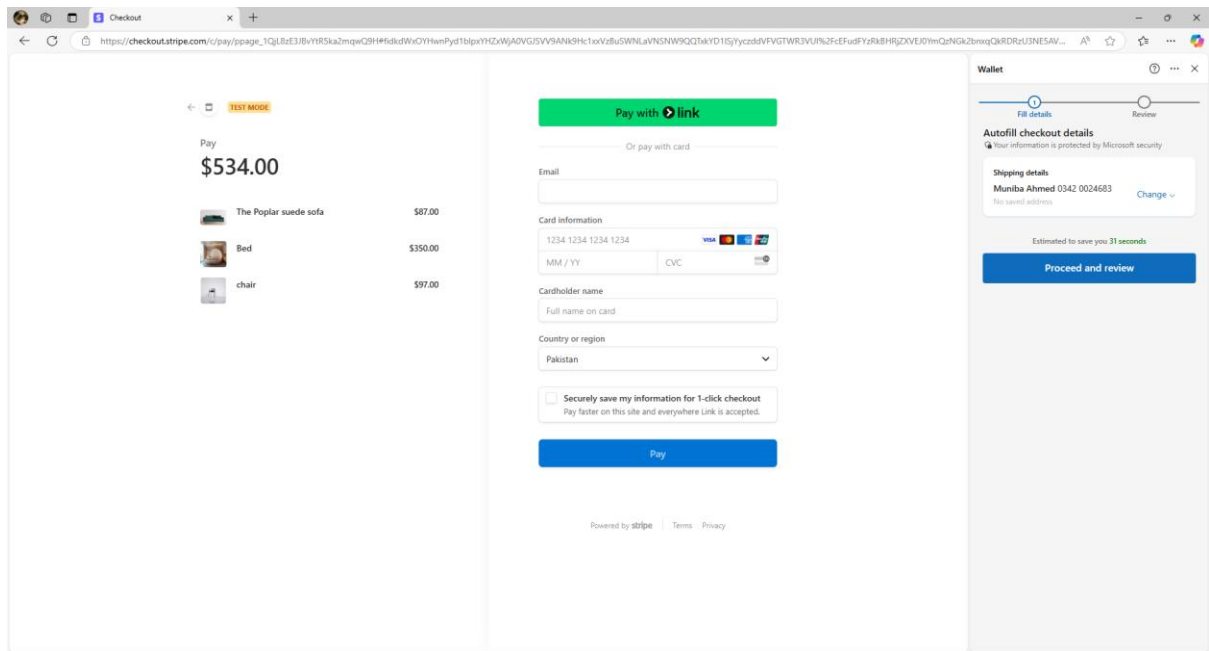


Pagination:

A pagination from shadcn ui added to my product detail dynamic route so we can check the detail of previous and next product through pagination:



Checkout component with stripe:



Code Snippet for important components and pages:

1- Product listing with new ceramic tag:

```

1 // Code hereimport client from "@sanity/lib/client";
2 import Image from "next/image";
3 import { Carousel, CarouselContent, CarouselItem } from "./ui/carousel";
4 import { Card, CardContent, CardFooter, CardHeader } from "./ui/card";
5 import Link from "next/link";
6
7 interface Product {
8   image: string;
9   name: string;
10  price: number;
11  description: string;
12  slug: { current: string }; // Correct slug structure from Sanity
13  price_id: string;
14  tags: string[];
15 }
16
17 const Ceramics = async () => {
18   // Sanity query to fetch products with the 'new ceramics' tag
19   const query = `*[ _type == 'product' && 'new ceramics' in tags ]{
20     "image": image.asset->url,
21     name,
22     price,
23     description,
24     slug, // Fetch slug object
25     price_id,
26     tags
27   }`;
28
29   const res: Product[] = await client.fetch(query);
30
31   return (
32     <div className="md:max-w-[1440px] w-full h-auto mx-auto">
33       { /* New Ceramics Section */ }
34       <div>
35         <div className="w-[1280px] mx-auto">
36           <h1 className="text-4xl my-8">New Ceramics</h1>
37         </div>
38
39         { /* Carousel Container */ }
40         <Carousel className="w-full max-w-full">
41           <CarouselContent>
42             { /* Map through the products */ }
43             {res.map((product: Product, index: number) => (
44               <CarouselItem
45                 key={index}
46                 className="md:basis-1/2 lg:basis-1/4"
47               >
48                 <Card className="w-full">
49                   <CardContent className="flex flex-col items-center justify-center">
50                     <CardHeader>
51                       <Link href={` /product/${product.slug.current} `}>
52                         <Image
53                           src={product.image || "/product1.png"}
54                           alt={product.name}
55                           height={1000}
56                           width={1000}
57                           className="h-full w-full object-cover cursor-pointer"
58                         />
59                       </Link>
60                     </CardHeader>
61                     <CardFooter className="flex items-center justify-between w-full h-[63px]">
62                       <p className="font-bold text-2xl">{product.name}</p>
63                       <p>£{product.price}</p>
64                     </CardFooter>
65                   </CardContent>
66                 </Card>
67               </CarouselItem>
68             )))
69           </CarouselContent>
70         </Carousel>
71       </div>
72     </div>
73   );
74 };
75
76 export default Ceramics;
77

```


2- Product detail dynamic route with pagination:

3- Category component that filters category:

```

1 //use client;
2
3 import { useEffect, useState } from "react";
4 export client from "@/sanity/lib/client";
5 import Image from "next/image";
6 import { Card, CardContent, CardFooter, CardHeader } from "../ui/card";
7 import Link from "next/link";
8
9 interface Product {
10   _id: string;
11   slug: string;
12   name: string;
13   price: number;
14   description: string;
15   slug?: current: string; // Correct slug structure from Sanity
16   price_id: string;
17   tags: string;
18 };
19
20 interface Category {
21   _id: string;
22   name: string;
23 };
24
25 slug {
26   current: string;
27 };
28
29 const Ceramics = () => {
30   const [products, setProducts] = useState<Product[]>([]);
31   const [categories, setCategories] = useState<Category[]>([]);
32   const [selectedCategory, setSelectedCategory] = useState<string>("");
33
34   // fetch all categories as component mount
35   useEffect(() => {
36     const fetchCategories = async () => {
37       try {
38         const categoriesQuery = `*[_type == "category"]{_id, name, "slug": slug.current}`;
39         const categoriesData: Category[] = await client.fetch(categoriesQuery);
40       } catch (error) {}
41       console.error("Error fetching categories:", error);
42     }
43   });
44
45   fetchCategories();
46
47 }, []);
48
49 const handleCategoryChange = (e: React.ChangeEvent<HTMLSelectElement>) => {
50   setSelectedCategory(e.target.value);
51 };
52
53 // Fetch products based on selected category
54 useEffect(() => {
55   const fetchProducts = async () => {
56     try {
57       const categoryFilter = selectedCategory
58         ? `*_refereces[?[_type == "category" && name == "${selectedCategory}"]._id]"`
59         : "";
60
61       const query = `*[_type == "product" ${categoryFilter}]{
62         _id,
63         "image": image.asset.url,
64         name,
65         price,
66         description,
67         slug,
68         price_id,
69         tags
70       }`;
71
72       const productData: Product[] = await client.fetch(query);
73       setProducts(productData);
74     } catch (error) {}
75     console.error("Error fetching products:", error);
76   };
77
78   fetchProducts();
79 }, [selectedCategory]);
80
81 return (
82   <div className="md:maw w-[1440px] w-full h-auto mx-auto">
83     <p>--current is filter by category --</p>
84     <div className="w-[1280px] mx-auto my-4">
85       <select
86         value={categories.find((category) => category._id === selectedCategory)?.slug.current}
87         onChange={handleCategoryChange}
88         className="border p-2 rounded"
89       >
90         <option value="">All Categories/options</option>
91         {categories.map((category) => {
92           <option key={category._id} value={category.slug.current}>
93             {category.name}
94           </option>
95         })}
96       </select>
97     </div>
98
99     <div>
100       <p>San Ceramics Section 4</p>
101
102       <div className="w-[1280px] mx-auto">
103         <div className="text-4xl my-8">All Products/hits</div>
104
105         <div>
106           <div>Grid Container</div>
107           <div className="grid grid-cols-1 lg:grid-cols-2 lg:grid-cols-4 gap-6 px-4">
108             <div>Map through the products --</div>
109             {products.length > 0 ? (
110               products.map((product) => {
111                 <Card key={product._id} className="w-full">
112                   <CardContent className="flex flex-col items-center justify-center">
113                     <Link href="/product/${product.slug.current}">
114                       <Image
115                         src={product.image || "/product1.png"}
116                         alt={product.name}
117                         height={100}
118                         width={100}
119                         className="h-full w-full object-cover cursor-pointer"
120                       />
121                     </Link>
122                     </CardContent>
123                     <CardFooter className="flex items-center justify-between w-full h-[50px]">
124                       <p>className="font-bold text-2xl">{product.name}</p>
125                       <p>className="text-sm">{product.price}</p>
126                     </CardFooter>
127                   </Card>
128                 )}
129               </div>
130             </div>
131             <div>
132               <p>No products found for this category.</p>
133             </div>
134           </div>
135         </div>
136       </div>
137     </div>
138 );
139 };
```

Conclusion:

- 1- Here is the outcome of front-end ui that is rendered from the headless CMS i.e: sanity. I write groq for fetching products, product details , category and tags in different component and pages.
- 2- Challenges I have faced when filtering category on the base of slug then i filter with name in place of slug.
- 3- Secondly I have a challenge in implementing pagination in dynamic route of product detail.
- 4- At the end the outcome achieve from the 4th day is product listing, product detail dynamic route, category component and add to cart component and page.