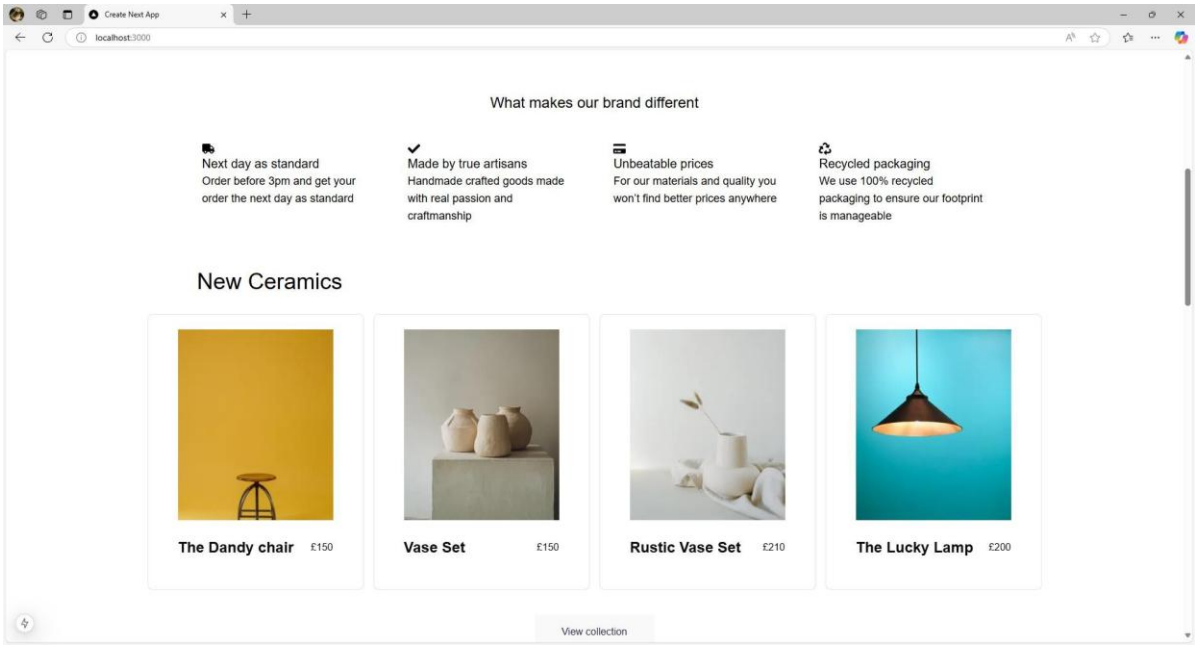
Day 4
Dynamic Frontend Components - [Avion]

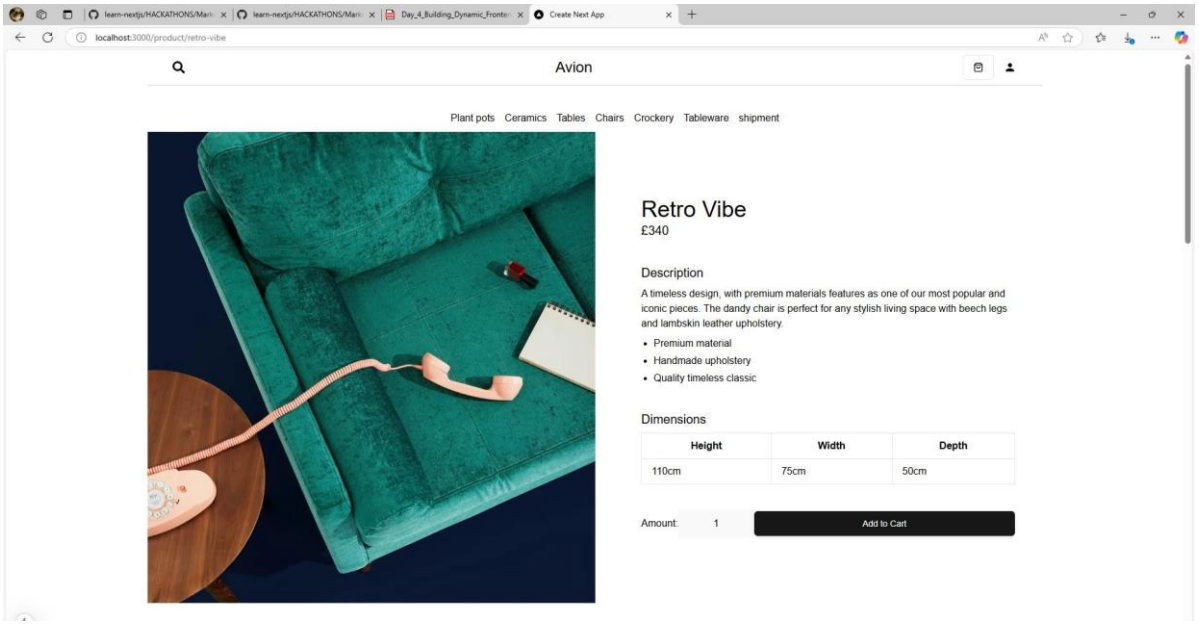
Product Listing Component:

Product List with filter of New ceramic tag.



Product Detail Page:

A dynamic product detail page that uses the items' slug field to retrieve data from Sanity

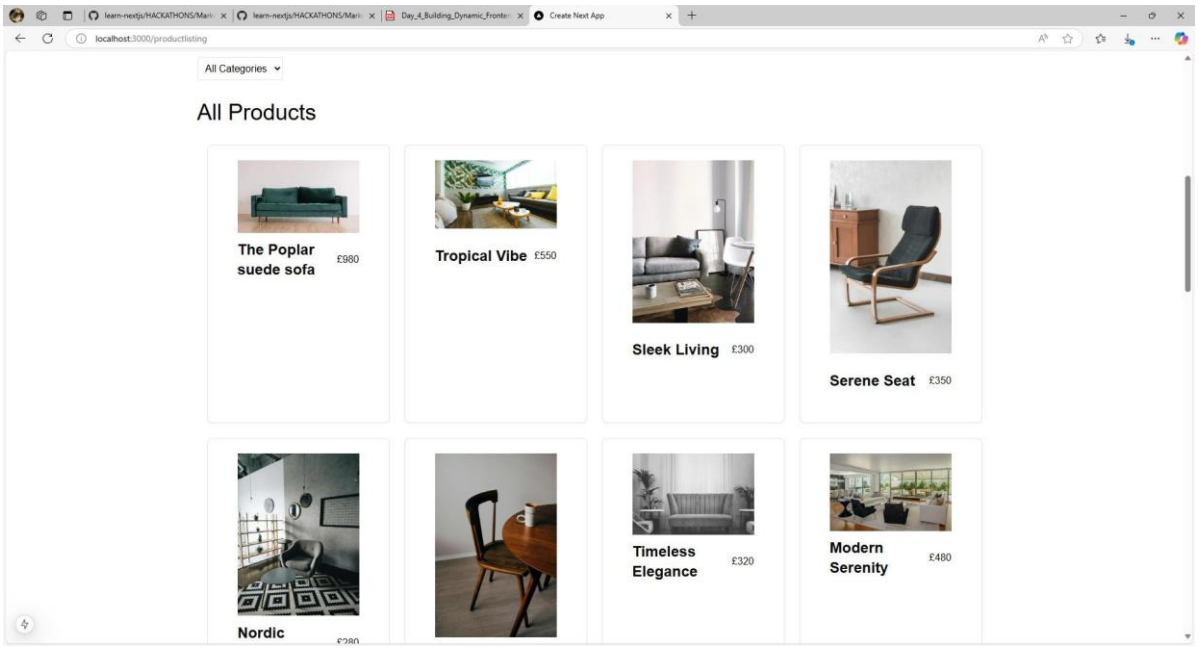


.

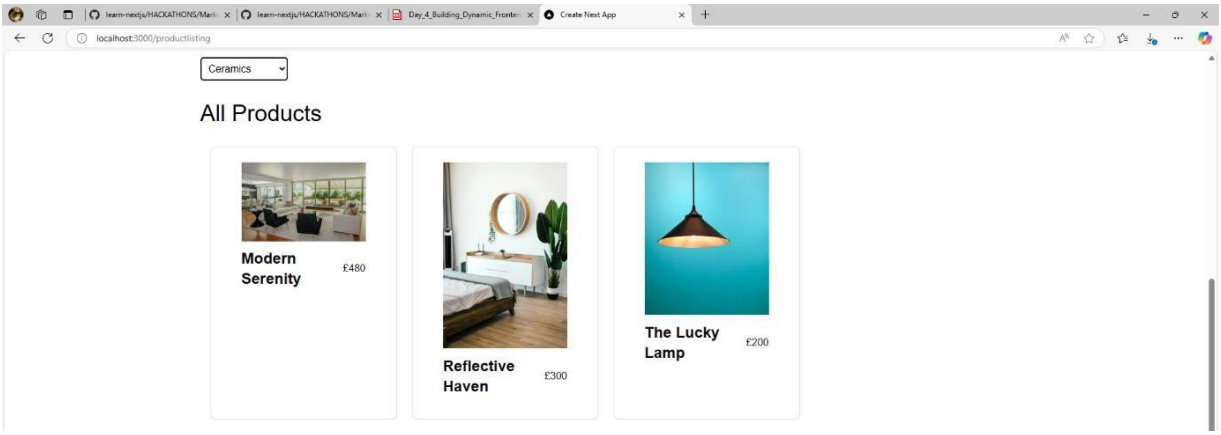
Category Component:

Category component with a drop-down list for sanity-derived category filtering.

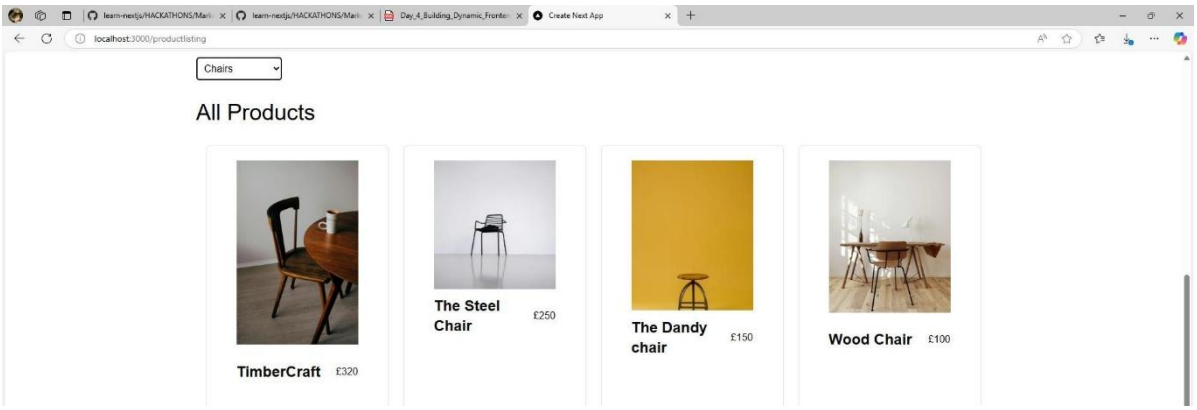
1- Products:



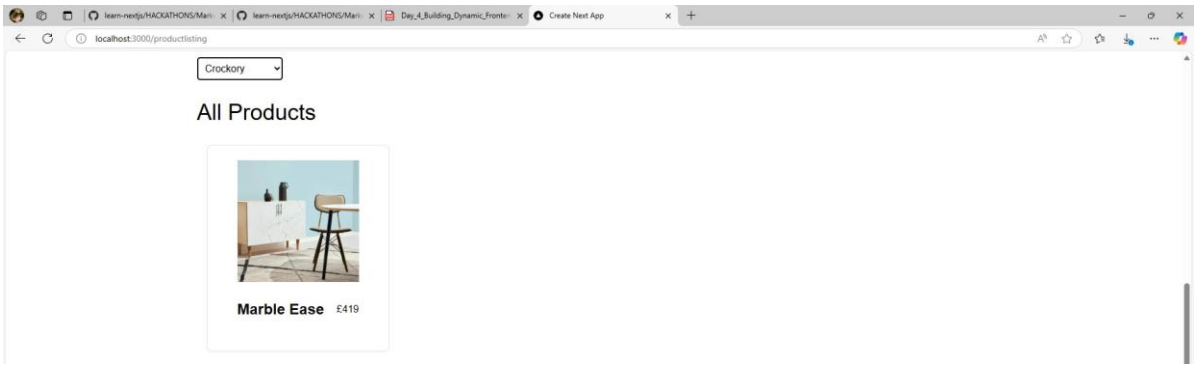
2- Ceramics



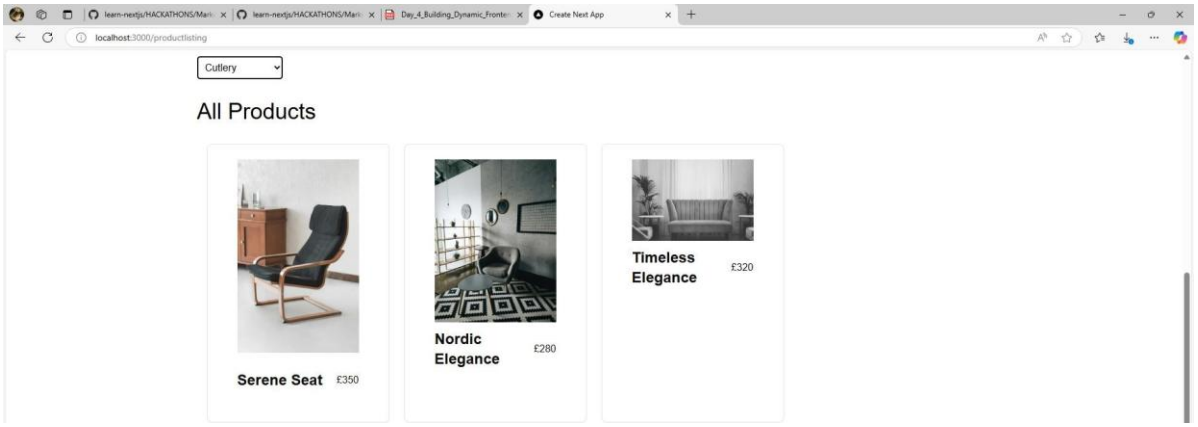
3- Chairs



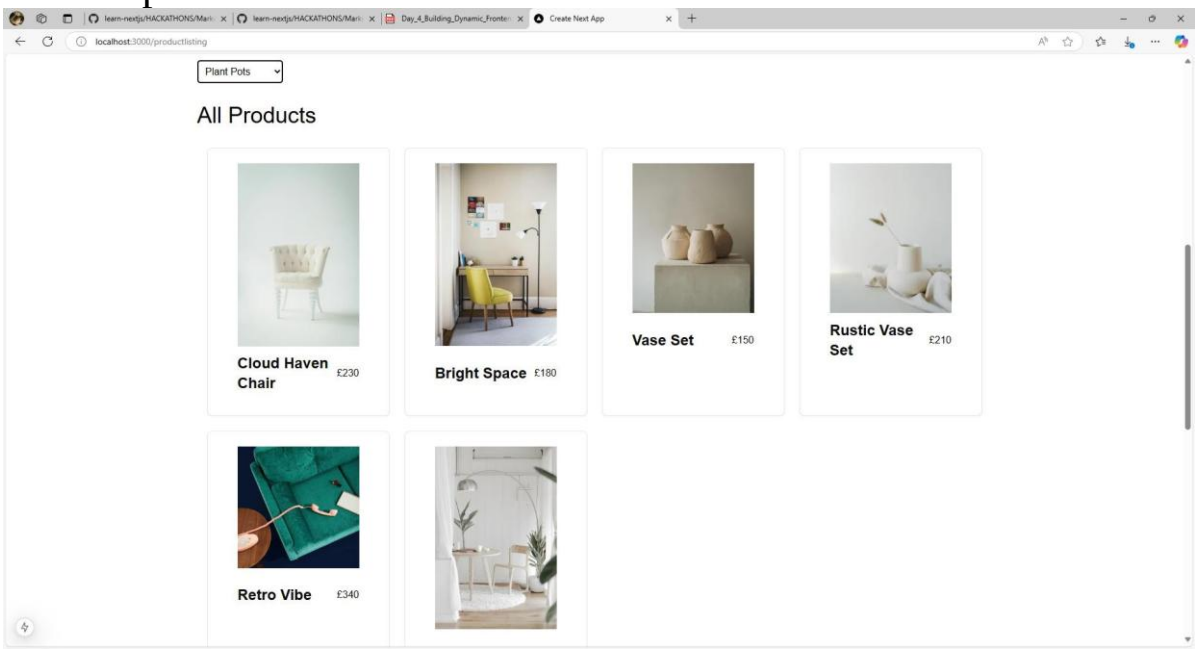
4- Crockery



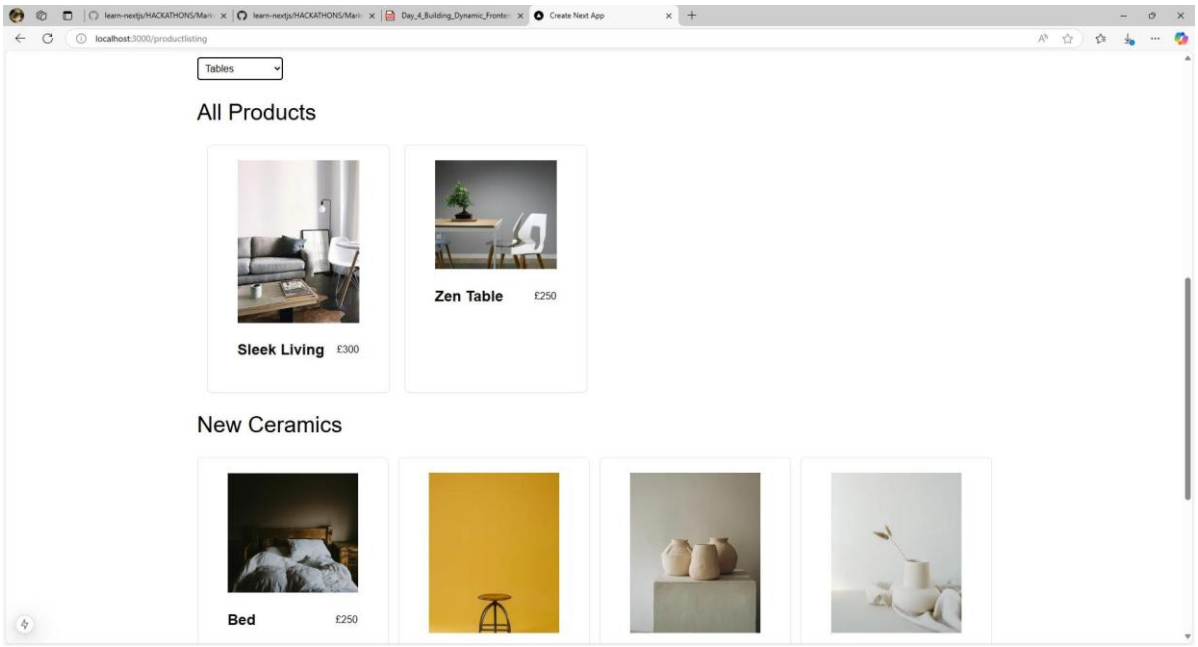
5- Cutlery



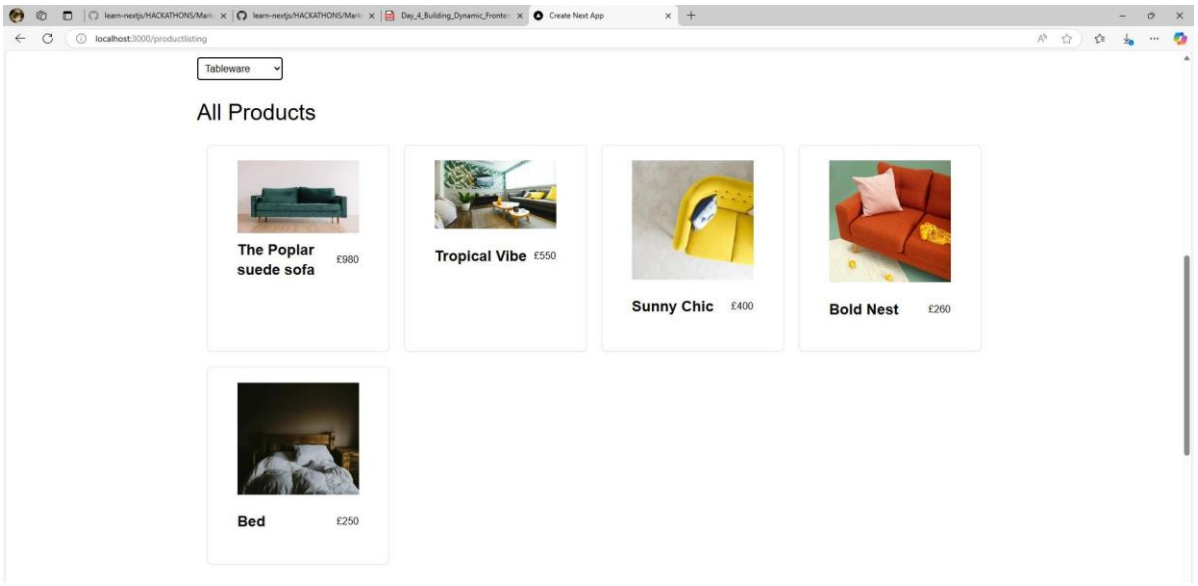
6- Plant pots



7- Tables

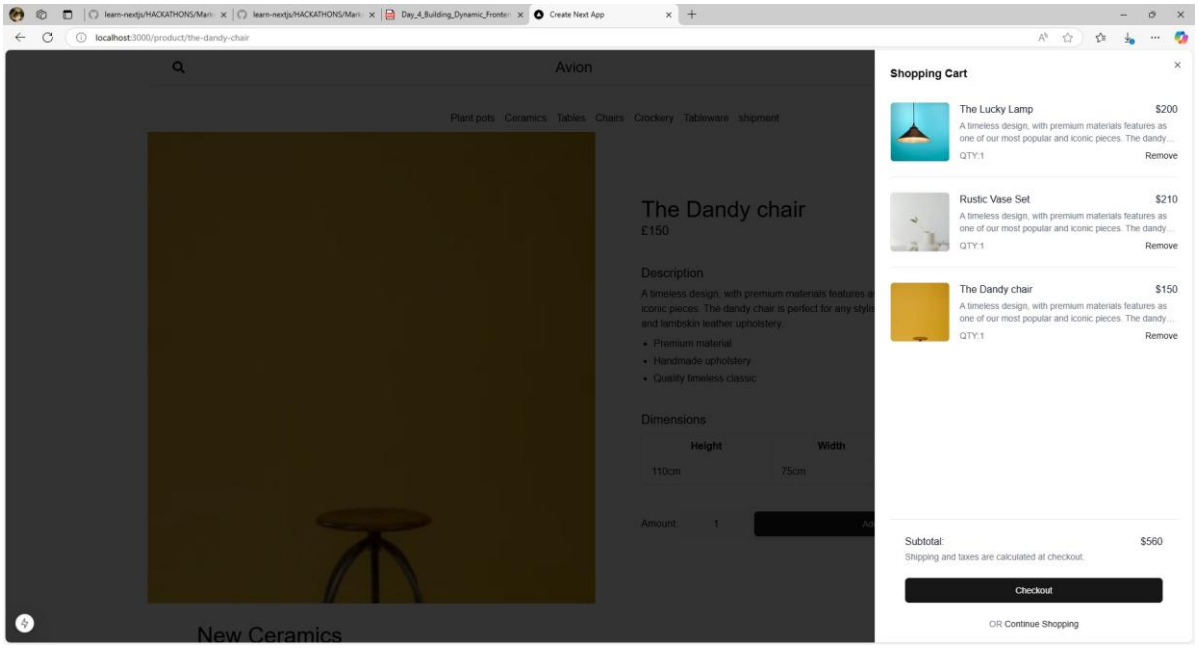


8- Tableware



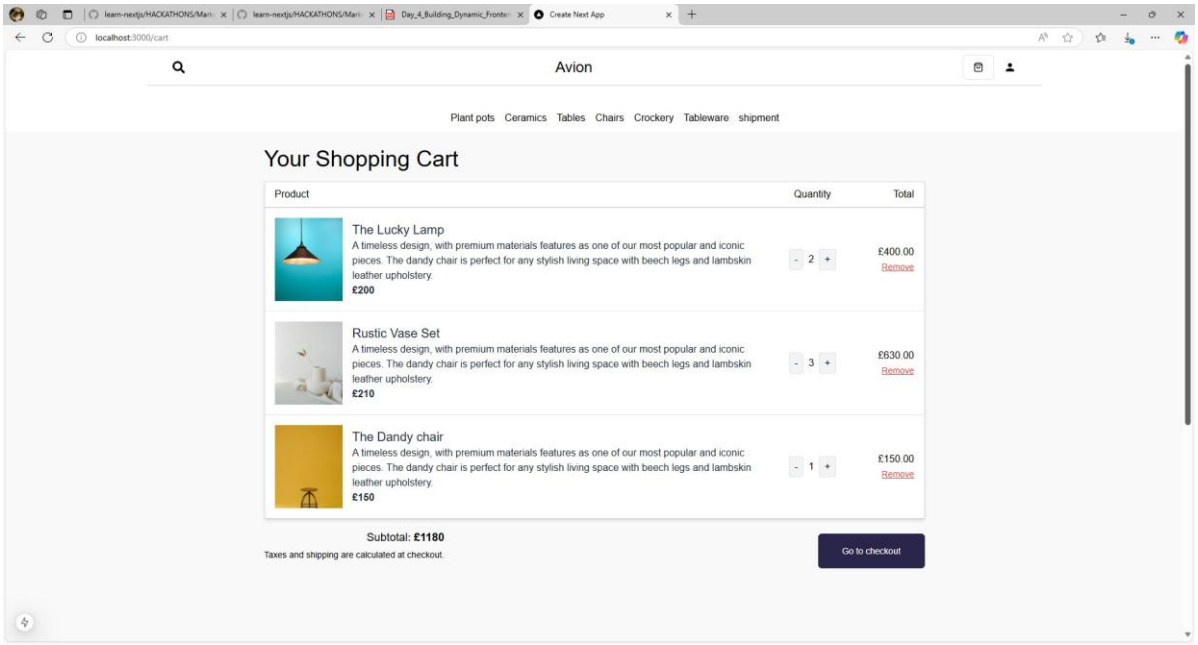
Add to Cart Component:

Using the shadcn ui sheet and shopping cart hook, the ShoppingCartModal Component appears on the right side when you click the add to cart or cart button on the navbar design.



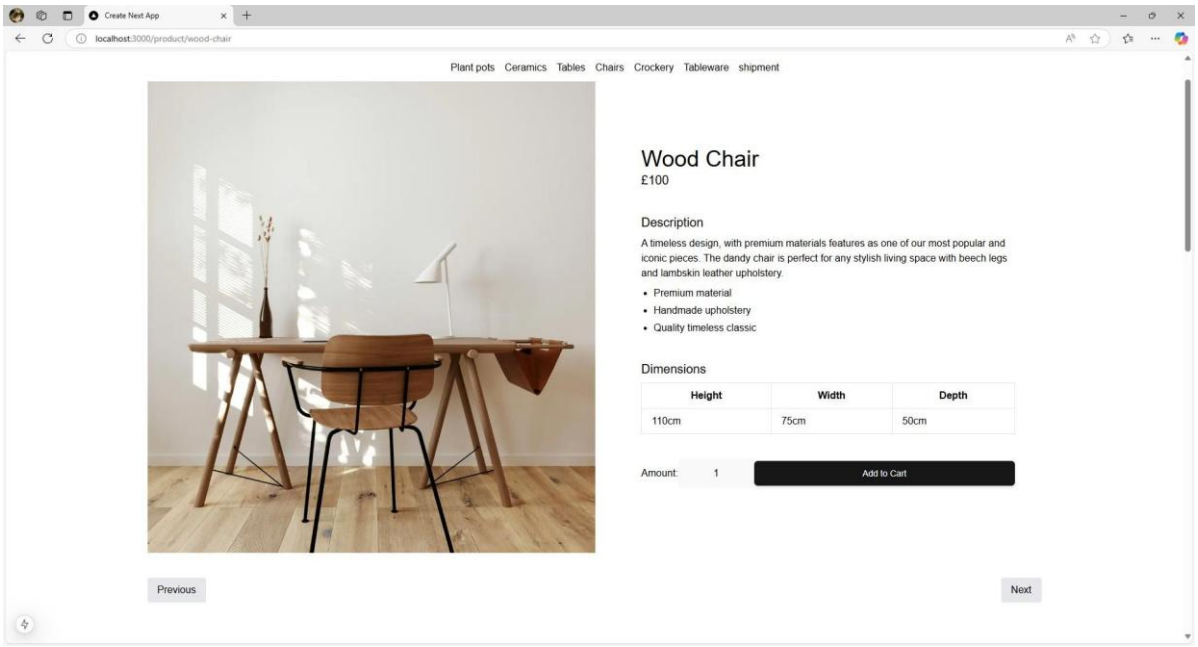
Cart Page: -

When you add or remove things from your cart, the price of all the products is displayed, followed by the subtotal of all the products.



Pagination:

My product detail dynamic route now has a pagination from Shadcn UI added so that we can use it to view the details of the prior and subsequent products:



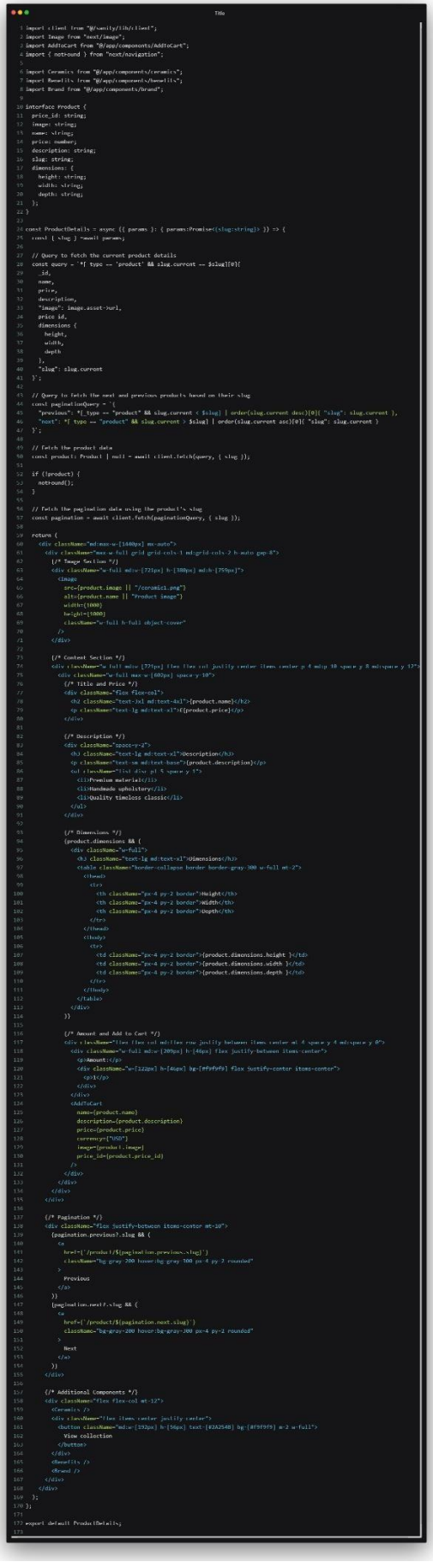
Code Snippet for important components and pages:

1) Product listing with new ceramic tag:

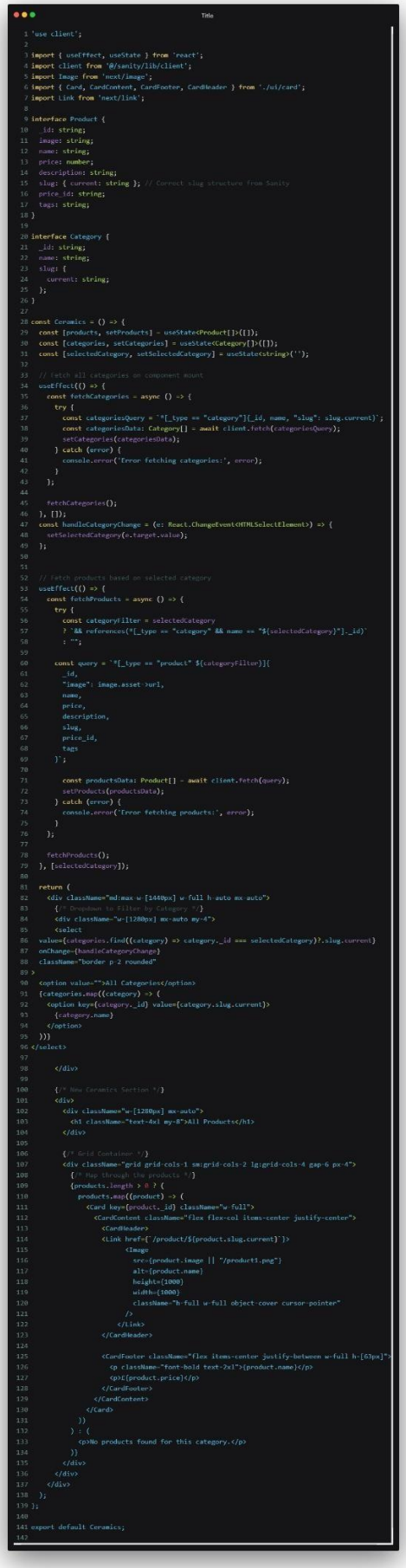
Title

```tsx
// Code hereimport client from "@/sanity/lib/client";
import Image from "next/image";
import { Carousel, CarouselContent, CarouselItem } from "./ui/carousel";
import { Card, CardContent, CardFooter, CardHeader } from "./ui/card";
import Link from "next/link";

interface Product {
  image: string;
  name: string;
  price: number;
  description: string;
  slug: { current: string }; // Correct slug structure from Sanity
  price_id: string;
  tags: string[];
}

const Ceramics = async () => {
  // Sanity query to fetch products with the 'new ceramics' tag
  const query = `*[_type == 'product' && 'new ceramics' in tags]{
    "image": image.asset->url,
    name,
    price,
    description,
    slug, // Fetch slug object
    price_id,
    tags
  }`;

  const res: Product[] = await client.fetch(query);

  return (
    <div className="md:max-w-[1440px] w-full h-auto mx-auto">
      {/* New Ceramics Section */}
      <div>
        <div className="w-[1280px] mx-auto">
          <h1 className="text-4xl my-8">New Ceramics</h1>
        </div>

        {/* Carousel Container */}
        <Carousel className="w-full max-w-full">
          <CarouselContent>
            {/* Map through the products */}
            {res.map((product: Product, index: number) => (
              <CarouselItem
                key={index}
                className="md:basis-1/2 lg:basis-1/4"
              >
                <Card className="w-full">
                  <CardContent className="flex flex-col items-center justify-center">
                    <CardHeader>
                      <Link href={`/product/${product.slug.current}`}>
                        <Image
                          src={product.image || "/product1.png"}
                          alt={product.name}
                          height={1000}
                          width={1000}
                          className="h-full w-full object-cover cursor-pointer"
                        />
                      </Link>
                    </CardHeader>
                    <CardFooter className="flex items-center justify-between w-full h-[63px]">
                      <p className="font-bold text-2xl">{product.name}</p>
                      <p>£{product.price}</p>
                    </CardFooter>
                  </CardContent>
                </Card>
              </CarouselItem>
            ))}
          </CarouselContent>
        </Carousel>
      </div>
    </div>
  );
};

export default Ceramics;

```

2) Product detail dynamic route with pagination:



3) Category component that filters category



Conclusion:

1. This is the result of the headless CMS's front-end user interface, or sanity. To retrieve goods, product data, categories, and tags from various components and sites, I create groq.
2. I encountered difficulties while filtering categories based on slugs and then filtering using names instead of slugs.
3. Secondly, I'm having trouble putting pagination into the dynamic product detail route.

4-The product listing, product detail dynamic route, category component, add to cart component, and page are the final results of the fourth day.