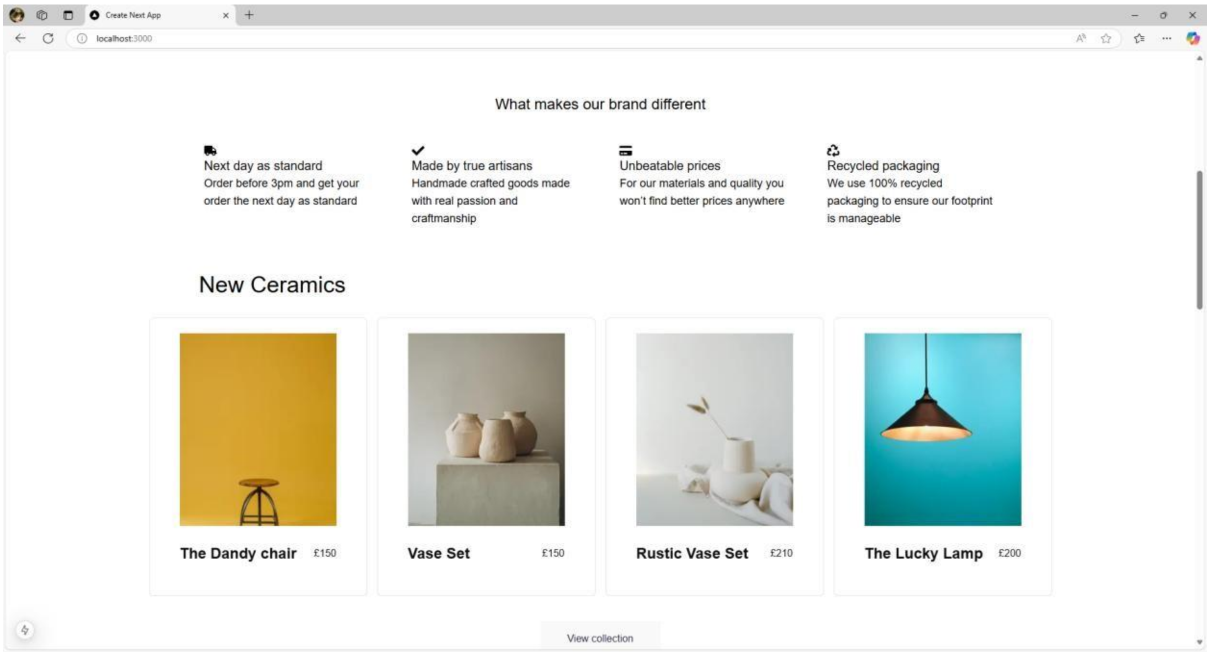
"Day 4 - Dynamic Frontend Components

Product Listing Component:

Product List with filter of New ceramic tag.



What makes our brand different

Next day as standard
Order before 3pm and get your order the next day as standard

Made by true artisans
Handmade crafted goods made with real passion and craftmanship

Unbeatable prices
For our materials and quality you won't find better prices anywhere

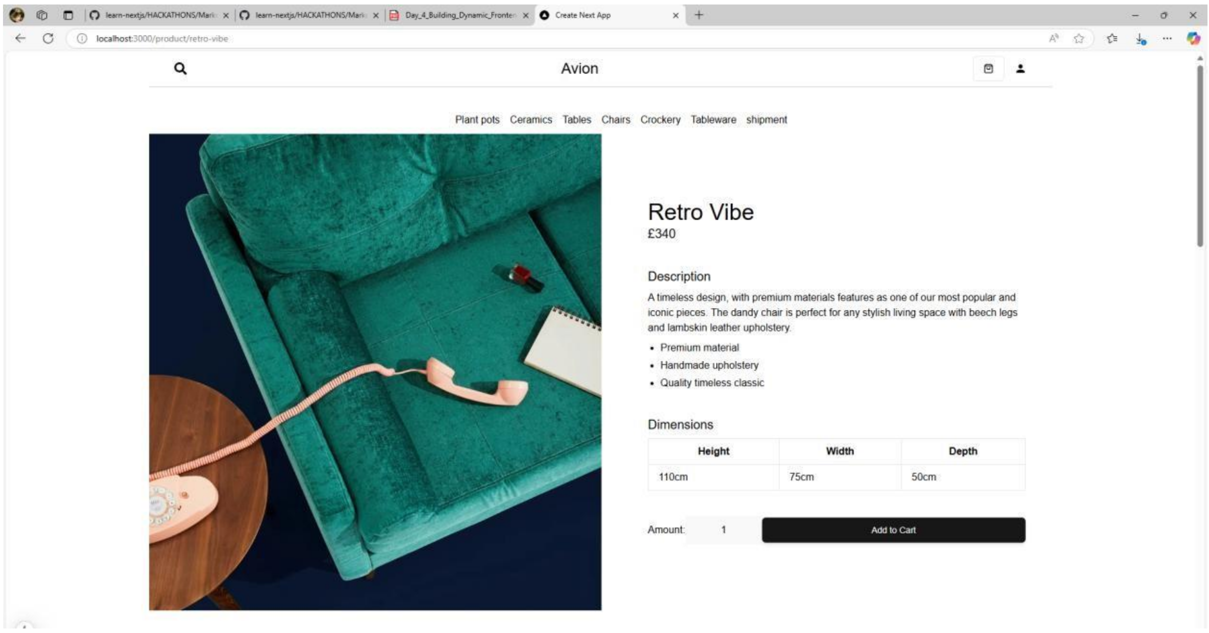
Recycled packaging
We use 100% recycled packaging to ensure our footprint is manageable

New Ceramics

The Dandy chair	£150
Vase Set	£150
Rustic Vase Set	£210
The Lucky Lamp	£200

Product Detail Page:

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


learn-nextjs/HACKATHONS/Mark
learn-nextjs/HACKATHONS/Mark
Day_4_Building_Dynamic_Fronten
Create Next App
localhost:3000/product/retro-vibe
Avion
Plant pots
Ceramics
Tables
Chairs
Crockery
Tableware
shipment
Add to Cart

Retro Vibe
£340

Description

A timeless design, with premium materials features as one of our most popular and iconic pieces. The dandy chair is perfect for any stylish living space with beech legs and lambskin leather upholstery.

- Premium material
- Handmade upholstery
- Quality timeless classic

Dimensions

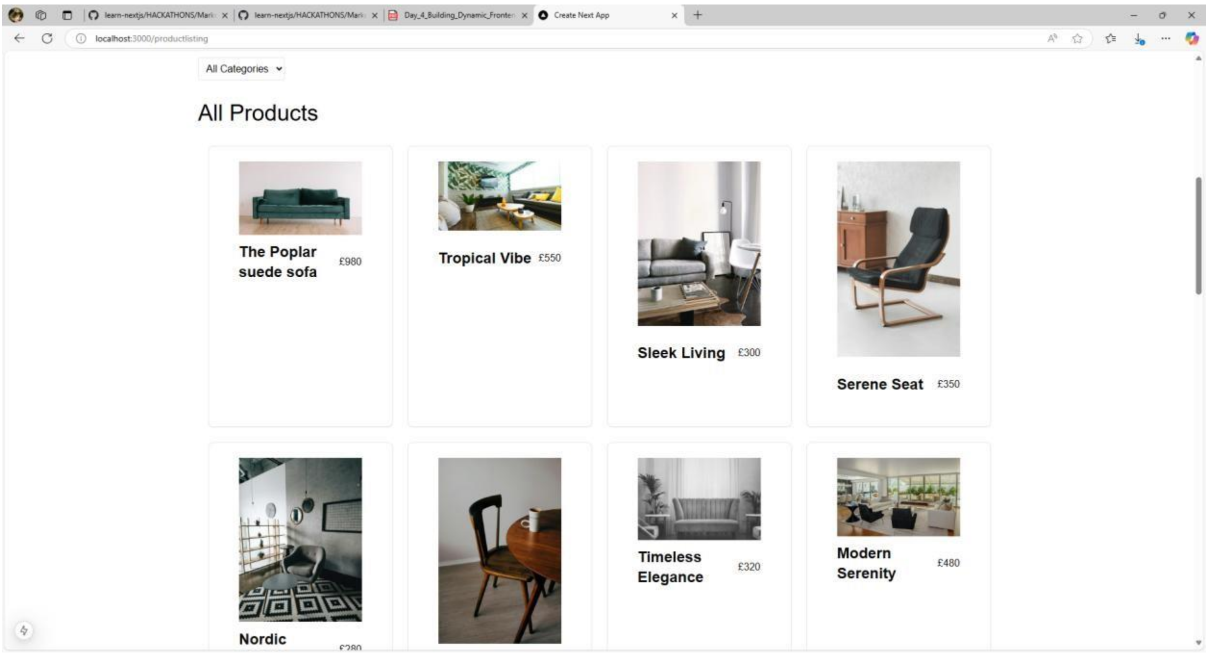
Height	Width	Depth
110cm	75cm	50cm

Amount: 1

Category Component:

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

1- All products:


learn-nextjs/HACKATHONS/Mark
learn-nextjs/HACKATHONS/Mark
Day_4_Building_Dynamic_Fronten
Create Next App
localhost:3000/productlisting
All Categories

All Products

The Poplar suede sofa £980

Tropical Vibe £550

Sleek Living £300

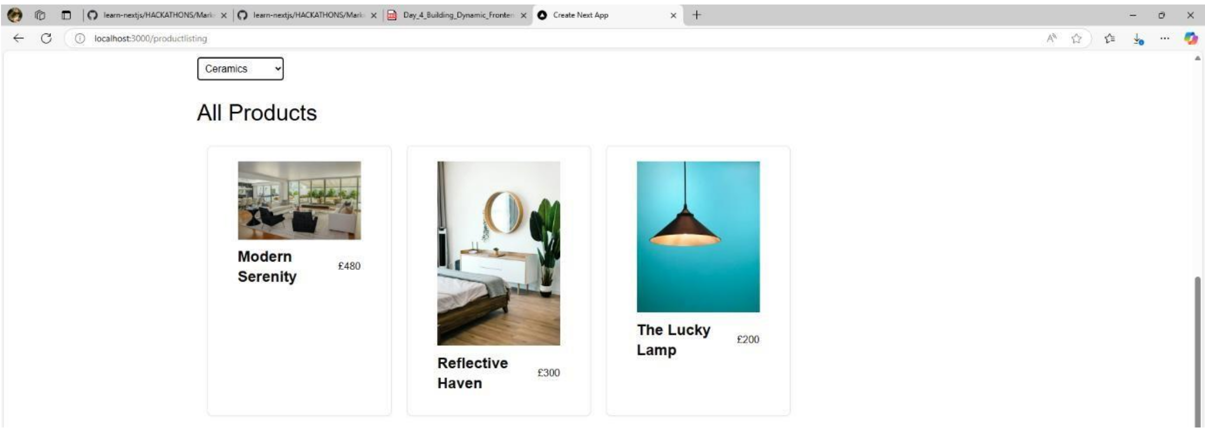
Serene Seat £350

Nordic £280

Timeless Elegance £320

Modern Serenity £480

2- Ceramics


learn-nextjs/HACKATHONS/Mark
learn-nextjs/HACKATHONS/Mark
Day_4_Building_Dynamic_Fronten
Create Next App
localhost:3000/productlisting
Ceramics

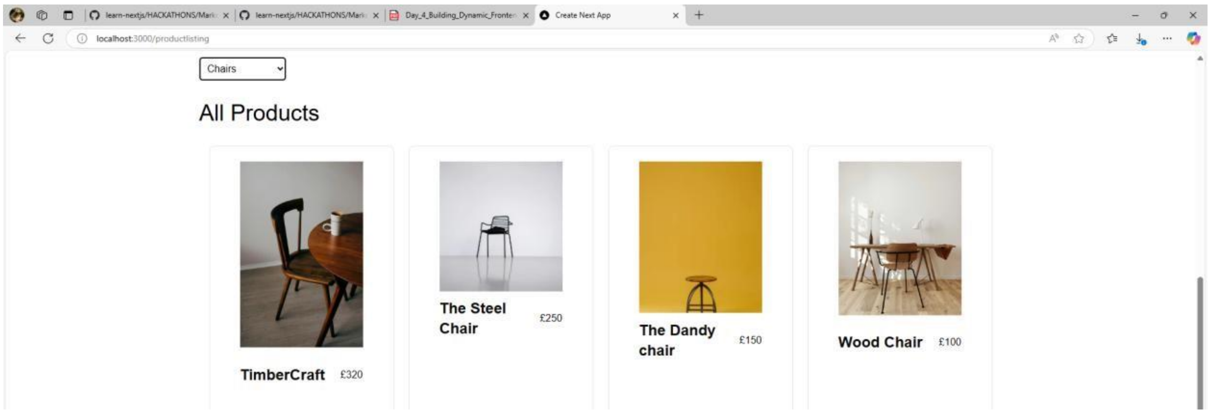
All Products

Modern Serenity £480

Reflective Haven £300

The Lucky Lamp £200

3- Chairs


learn-nextjs/HACKATHONS/Mark
learn-nextjs/HACKATHONS/Mark
Day_4_Building_Dynamic_Fronte
Create Next App
localhost:3000/productlisting

Chairs

All Products

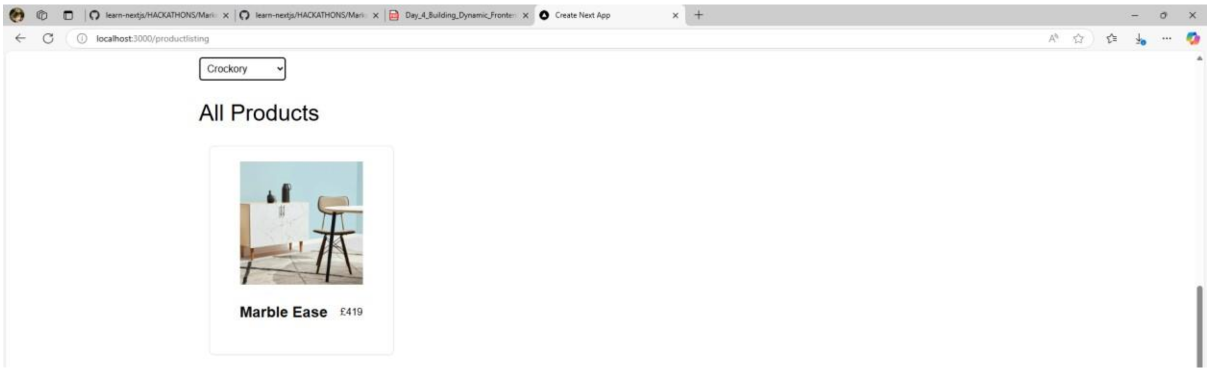
TimberCraft £320

The Steel Chair £250

The Dandy chair £150

Wood Chair £100

4- Crockery

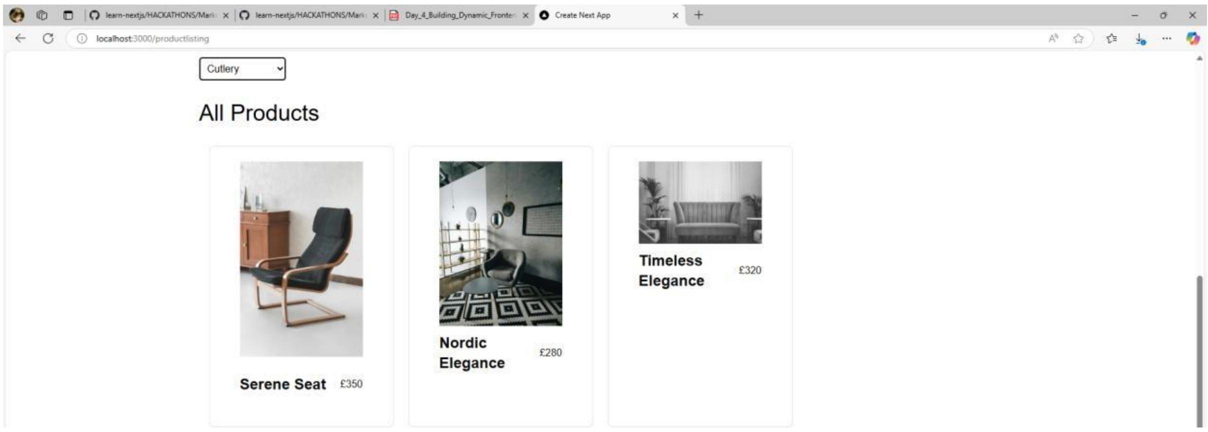

learn-nextjs/HACKATHONS/Mark
learn-nextjs/HACKATHONS/Mark
Day_4_Building_Dynamic_Fronte
Create Next App
localhost:3000/productlisting

Crockery

All Products

Marble Ease £419

5- Cutlery


learn-nextjs/HACKATHONS/Mark
learn-nextjs/HACKATHONS/Mark
Day_4_Building_Dynamic_Fronte
Create Next App
localhost:3000/productlisting

Cutlery

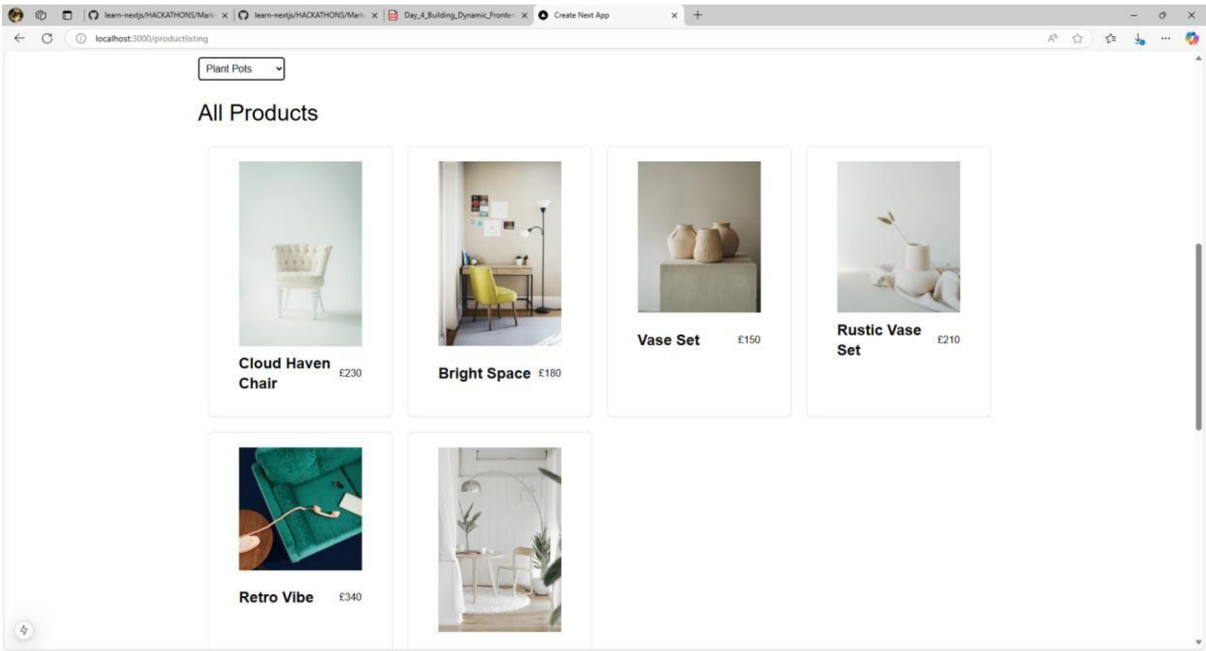
All Products

Serene Seat £350

Nordic Elegance £280

Timeless Elegance £320

6- Plant pots


learn-nextjs/HACKATHONS/Mark
learn-nextjs/HACKATHONS/Mark
Day_4_Building_Dynamic_Fronten
Create Next App
localhost:3000/productlisting

Plant Pots

All Products

Cloud Haven Chair £230

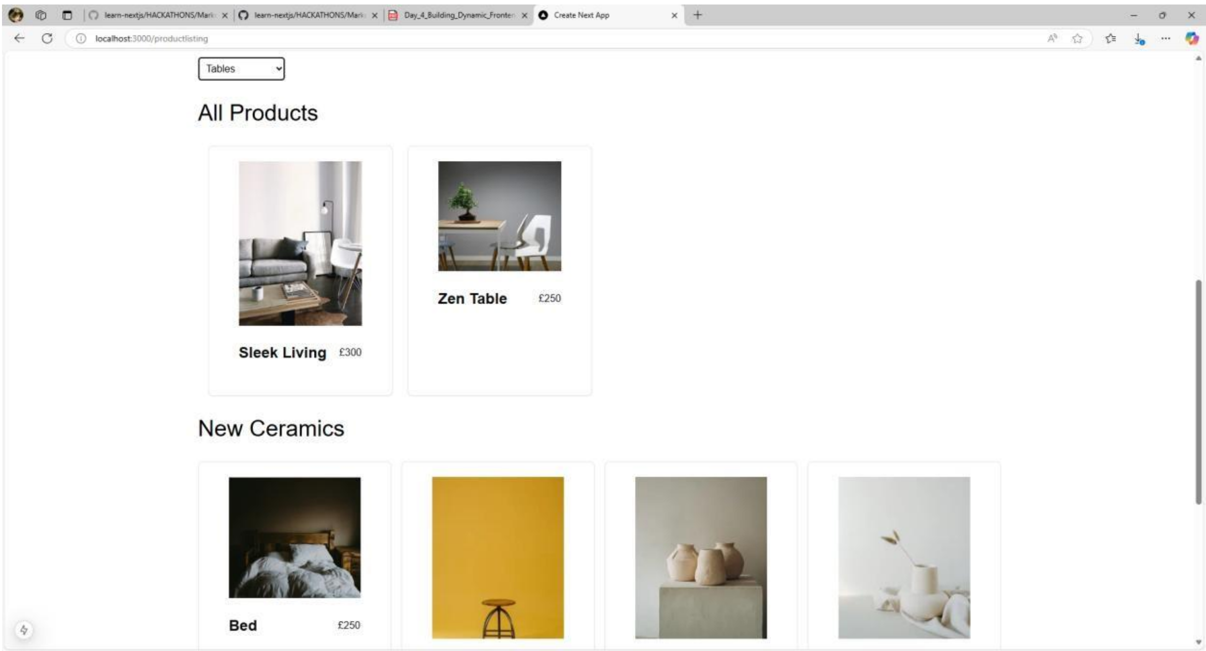
Bright Space £180

Vase Set £150

Rustic Vase Set £210

Retro Vibe £340

7- Tables


learn-nextjs/HACKATHONS/Mark
learn-nextjs/HACKATHONS/Mark
Day_4_Building_Dynamic_Fronten
Create Next App
localhost:3000/productlisting

Tables

All Products

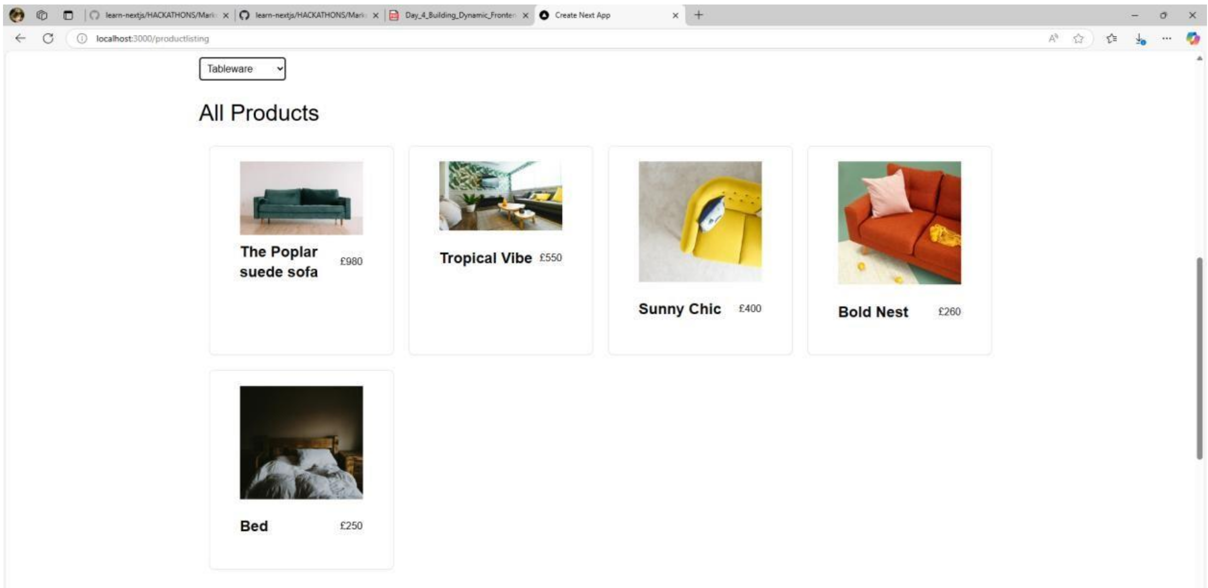
Sleek Living £300

Zen Table £250

New Ceramics

Bed £250

8- Tableware

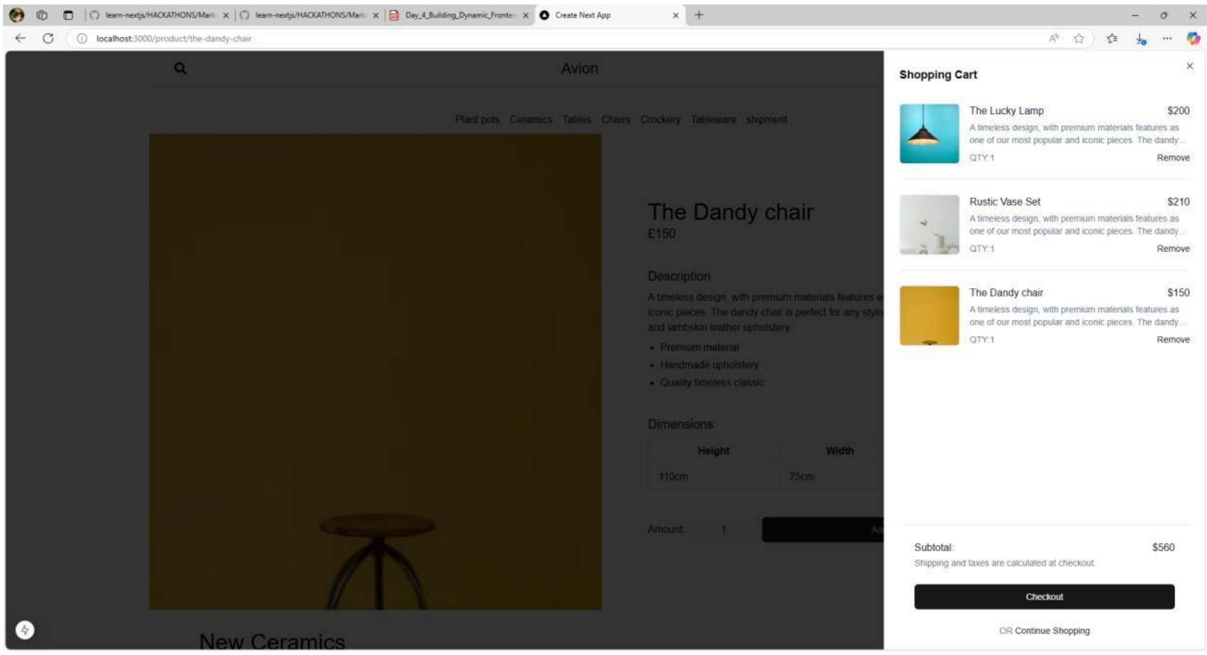

learn-nextjs/HACKATHONS/Mark
learn-nextjs/HACKATHONS/Mark
Day_4_Building_Dynamic_Fronten
Create Next App
localhost:3000/productlisting
Tableware

All Products

The Poplar suede sofa £980
Tropical Vibe £550
Sunny Chic £400
Bold Nest £260
Bed £250

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.


learn-nextjs/HACKATHONS/Mark
learn-nextjs/HACKATHONS/Mark
Day_4_Building_Dynamic_Fronten
Create Next App
localhost:3000/product/the-dandy-chair
Avion
Plant pots
Ceramics
Tables
Chairs
Crockery
Tableware
shipment

The Dandy chair
£150

Description

A timeless design, with premium materials features a
iconic pieces. The dandy chair is perfect for any stylis
and lambskin leather upholstery

- Premium material
- Handmade upholstery
- Quality timeless classic

Dimensions

Height	Width
110cm	75cm

Amount 1

New Ceramics

Shopping Cart

The Lucky Lamp $200
A timeless design, with premium materials features as one of our most popular and iconic pieces. The dandy…
QTY:1 Remove

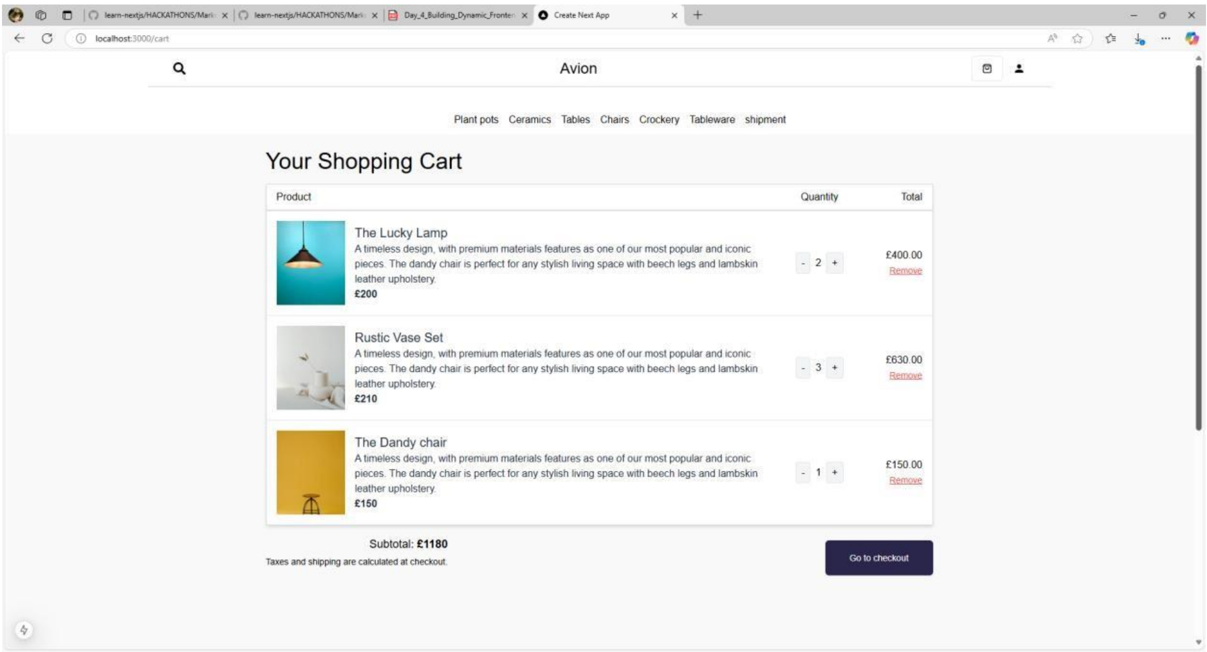
Rustic Vase Set $210
A timeless design, with premium materials features as one of our most popular and iconic pieces. The dandy…
QTY:1 Remove

The Dandy chair $150
A timeless design, with premium materials features as one of our most popular and iconic pieces. The dandy…
QTY:1 Remove

Subtotal: $560
Shipping and taxes are calculated at checkout.

Cart Page:-

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


learn-nextjs/HACKATHONS/Mark
learn-nextjs/HACKATHONS/Mark
Day_4_Building_Dynamic_Fronte
Create Next App
localhost:3000/cart
Avion
Plant pots Ceramics Tables Chairs Crockery Tableware shipment

Your Shopping Cart

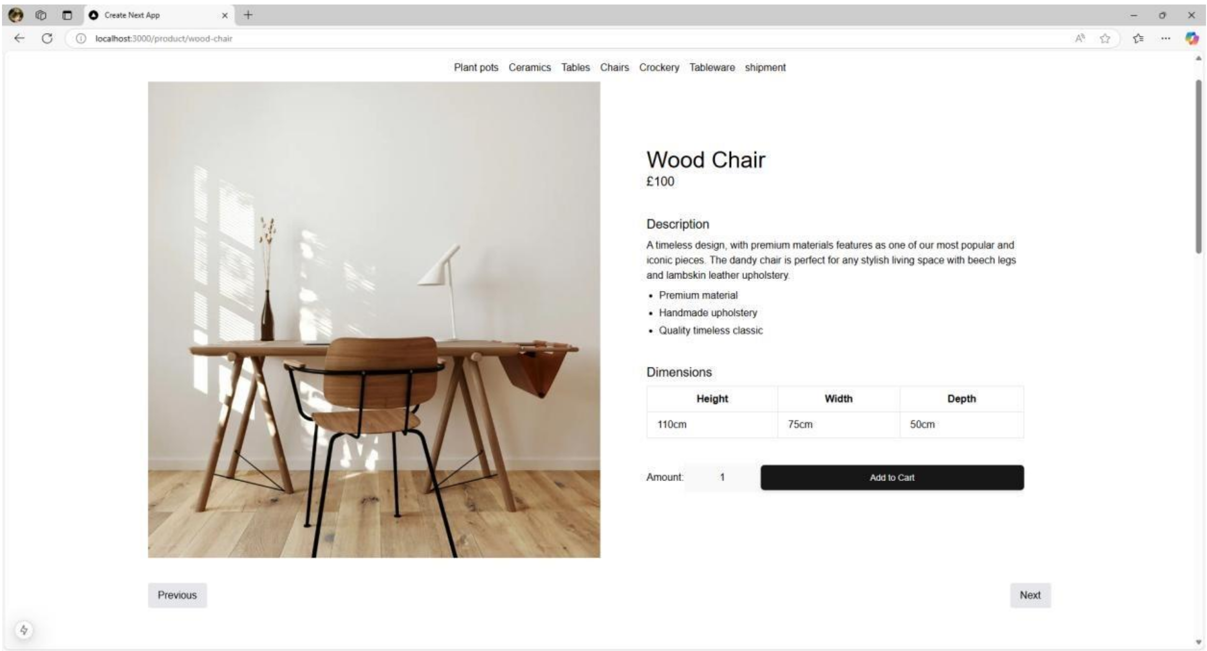
Product	Quantity	Total
The Lucky Lamp A timeless design, with premium materials features as one of our most popular and iconic pieces. The dandy chair is perfect for any stylish living space with beech legs and lambskin leather upholstery. **£200**	- 2 +	£400.00 Remove
Rustic Vase Set A timeless design, with premium materials features as one of our most popular and iconic pieces. The dandy chair is perfect for any stylish living space with beech legs and lambskin leather upholstery. **£210**	- 3 +	£630.00 Remove
The Dandy chair A timeless design, with premium materials features as one of our most popular and iconic pieces. The dandy chair is perfect for any stylish living space with beech legs and lambskin leather upholstery. **£150**	- 1 +	£150.00 Remove

Subtotal: **£1180**

Taxes and shipping are calculated at checkout.

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:


Create Next App
localhost:3000/product/wood-chair
Plant pots Ceramics Tables Chairs Crockery Tableware shipment

Wood Chair

£100

Description

A timeless design, with premium materials features as one of our most popular and iconic pieces. The dandy chair is perfect for any stylish living space with beech legs and lambskin leather upholstery.

- Premium material
- Handmade upholstery
- Quality timeless classic

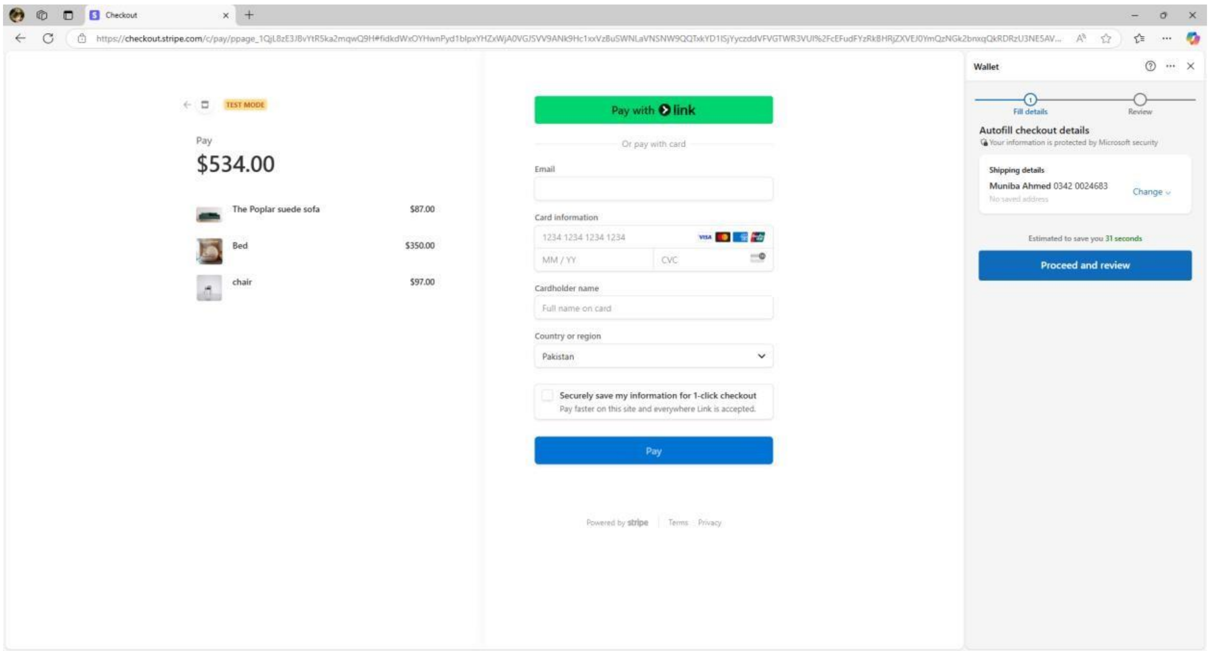
Dimensions

Height	Width	Depth
110cm	75cm	50cm

Amount: 1

Add to Cart
Previous
Next

Checkout component with stripe:



Checkout
https://checkout.stripe.com/c/pay/ppage_1QjL8zE3JBvYtR5ka2mqwQ9H#fidkdWxOYHwnPyd1blpxYHZxWjA0VGJSVV9ANk9Hc1xxVzBuSWNLaVNSNW9QQTxkYD1ISjYyczddVFVGTWR3VUl%2FcEFudFYzRkBHRjZXVEJ0YmQzNGk2bnxqQkRDRzU3NE5AV...
TEST MODE

Pay
$534.00

The Poplar suede sofa	$87.00
Bed	$350.00
chair	$97.00

Pay with link
Or pay with card

Email

Card information
1234 1234 1234 1234
MM / YY
CVC

Cardholder name
Full name on card

Country or region
Pakistan

□ Securely save my information for 1-click checkout
Pay faster on this site and everywhere Link is accepted.

Pay
Powered by stripe
Terms
Privacy
Wallet
Fill details
Review

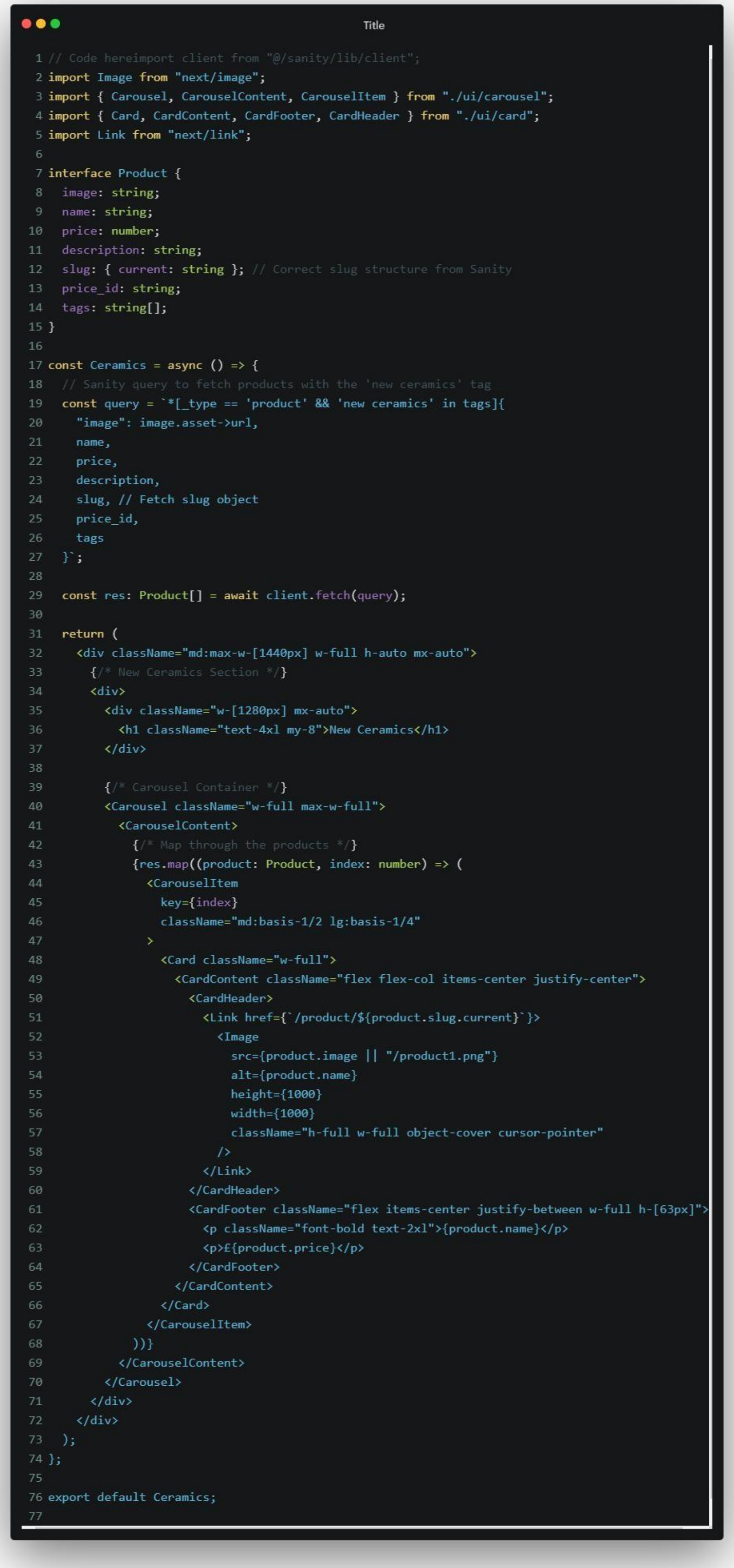
Autofill checkout details
Your information is protected by Microsoft security

Shipping details
Muniba Ahmed 0342 0024683
No saved address

Estimated to save you 31 seconds

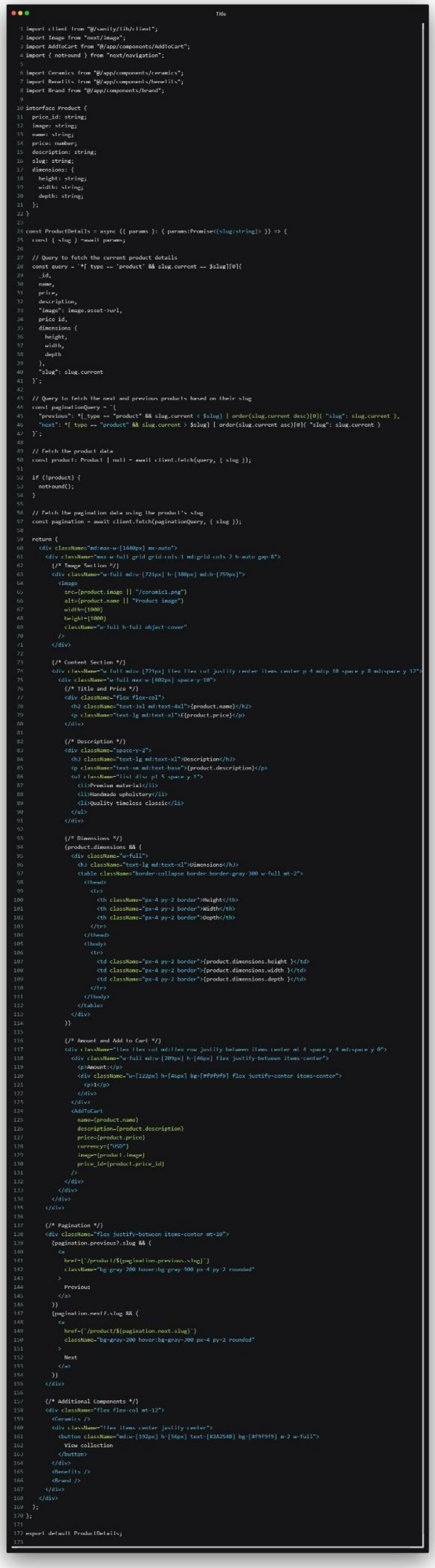
Code Snippet for important components and pages:

1- Product listing with new ceramic tag:



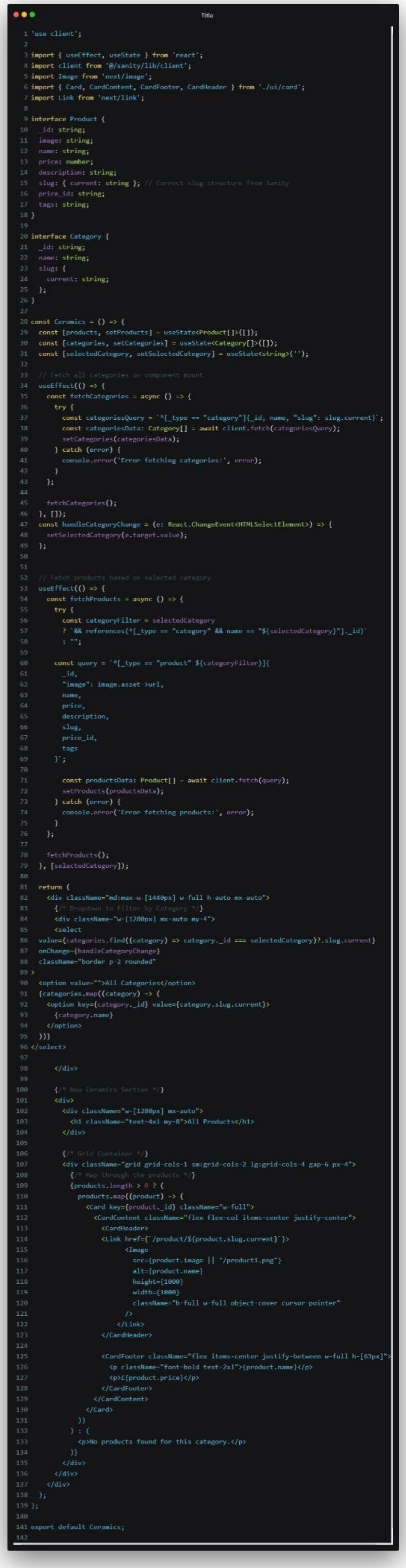
```
// Code hereimport client from "@/sanity/lib/client";
import Image from "next/image";
import { Carousel, CarouselContent, CarouselItem } from "./ui/carousel";
import { Card, CardContent, CardFooter, CardHeader } from "./ui/card";
import Link from "next/link";

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:



```tsx
import client from "@/sanity/lib/client";
import Image from "next/image";
import AddToCart from "@/app/components/AddToCart";
import { notFound } from "next/navigation";

import Ceramics from "@/app/components/ceramics";
import Benefits from "@/app/components/benefits";
import Brand from "@/app/components/brand";

interface Product {
  price_id: string;
  image: string;
  name: string;
  price: number;
  description: string;
  slug: string;
  dimensions: {
    height: string;
    width: string;
    depth: string;
  };
}

const ProductDetails = async ({ params }: { params:Promise<{slug:string}> }) => {
  const { slug } =await params;

  // Query to fetch the current product details
  const query = `*[_type == 'product' && slug.current == $slug][0]{
    _id,
    name,
    price,
    description,
    "image": image.asset->url,
    price_id,
    dimensions {
      height,
      width,
      depth
    },
    "slug": slug.current
  }`;

  // Query to fetch the next and previous products based on their slug
  const paginationQuery = `{
    "previous": *[_type == "product" && slug.current < $slug] | order(slug.current desc)[0]{ "slug": slug.current },
    "next": *[_type == "product" && slug.current > $slug] | order(slug.current asc)[0]{ "slug": slug.current }
  }`;

  // Fetch the product data
  const product: Product | null = await client.fetch(query, { slug });

  if (!product) {
    notFound();
  }

  // Fetch the pagination data using the product's slug
  const pagination = await client.fetch(paginationQuery, { slug });

  return (
    <div className="md:max-w-[1440px] mx-auto">
      <div className="max-w-full grid grid-cols-1 md:grid-cols-2 h-auto gap-8">
        {/* Image Section */}
        <div className="w-full md:w-[721px] h-[380px] md:h-[759px]">
          <Image
            src={product.image || "/ceramic1.png"}
            alt={product.name || "Product image"}
            width={1000}
            height={1000}
            className="w-full h-full object-cover"
          />
        </div>

        {/* Content Section */}
        <div className="w-full md:w-[721px] flex flex-col justify-center items-center p-4 md:p-10 space-y-8 md:space-y-12">
          <div className="w-full max-w-[602px] space-y-10">
            {/* Title and Price */}
            <div className="flex flex-col">
              <h2 className="text-3xl md:text-4xl">{product.name}</h2>
              <p className="text-lg md:text-xl">£{product.price}</p>
            </div>

            {/* Description */}
            <div className="space-y-2">
              <h3 className="text-lg md:text-xl">Description</h3>
              <p className="text-sm md:text-base">{product.description}</p>
              <ul className="list-disc pl-5 space-y-1">
                <li>Premium material</li>
                <li>Handmade upholstery</li>
                <li>Quality timeless classic</li>
              </ul>
            </div>

            {/* Dimensions */}
            {product.dimensions && (
              <div className="w-full">
                <h3 className="text-lg md:text-xl">Dimensions</h3>
                <table className="border-collapse border border-gray-300 w-full mt-2">
                  <thead>
                    <tr>
                      <th className="px-4 py-2 border">Height</th>
                      <th className="px-4 py-2 border">Width</th>
                      <th className="px-4 py-2 border">Depth</th>
                    </tr>
                  </thead>
                  <tbody>
                    <tr>
                      <td className="px-4 py-2 border">{product.dimensions.height }</td>
                      <td className="px-4 py-2 border">{product.dimensions.width }</td>
                      <td className="px-4 py-2 border">{product.dimensions.depth }</td>
                    </tr>
                  </tbody>
                </table>
              </div>
            )}

            {/* Amount and Add to Cart */}
            <div className="flex flex-col md:flex-row justify-between items-center mt-4 space-y-4 md:space-y-0">
              <div className="w-full md:w-[209px] h-[46px] flex justify-between items-center">
                <p>Amount:</p>
                <div className="w-[122px] h-[46px] bg-[#f9f9f9] flex justify-center items-center">
                  <p>1</p>
                </div>
              </div>
              <AddToCart
                name={product.name}
                description={product.description}
                price={product.price}
                currency={"USD"}
                image={product.image}
                price_id={product.price_id}
              />
            </div>
          </div>
        </div>
      </div>

      {/* Pagination */}
      <div className="flex justify-between items-center mt-10">
        {pagination.previous?.slug && (
          <a
            href={`/product/${pagination.previous.slug}`}
            className="bg-gray-200 hover:bg-gray-300 px-4 py-2 rounded"
          >
            Previous
          </a>
        )}
        {pagination.next?.slug && (
          <a
            href={`/product/${pagination.next.slug}`}
            className="bg-gray-200 hover:bg-gray-300 px-4 py-2 rounded"
          >
            Next
          </a>
        )}
      </div>

      {/* Additional Components */}
      <div className="flex flex-col mt-12">
        <Ceramics />
        <div className="flex items-center justify-center">
          <button className="md:w-[192px] h-[56px] text-[#2A254B] bg-[#f9f9f9] m-2 w-full">
            View collection
          </button>
        </div>
        <Benefits />
        <Brand />
      </div>
    </div>
  );
};

export default ProductDetails;
```

3- Category component that filters category:



```tsx
'use client';

import { useEffect, useState } from 'react';
import client from '@/sanity/lib/client';
import Image from 'next/image';
import { Card, CardContent, CardFooter, CardHeader } from './ui/card';
import Link from 'next/link';

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

interface Category {
  _id: string;
  name: string;
  slug: {
    current: string;
  };
}

const Ceramics = () => {
  const [products, setProducts] = useState<Product[]>([]);
  const [categories, setCategories] = useState<Category[]>([]);
  const [selectedCategory, setSelectedCategory] = useState<string>('');

  // Fetch all categories on component mount
  useEffect(() => {
    const fetchCategories = async () => {
      try {
        const categoriesQuery = `*[_type == "category"]{_id, name, "slug": slug.current}`;
        const categoriesData: Category[] = await client.fetch(categoriesQuery);
        setCategories(categoriesData);
      } catch (error) {
        console.error('Error fetching categories:', error);
      }
    };

    fetchCategories();
  }, []);
  const handleCategoryChange = (e: React.ChangeEvent<HTMLSelectElement>) => {
    setSelectedCategory(e.target.value);
  };


  // Fetch products based on selected category
  useEffect(() => {
    const fetchProducts = async () => {
      try {
        const categoryFilter = selectedCategory
        ? `&& references(*[_type == "category" && name == "${selectedCategory}"]._id)`
        : "";

      const query = `*[_type == "product" ${categoryFilter}]{
        _id,
        "image": image.asset->url,
        name,
        price,
        description,
        slug,
        price_id,
        tags
      }`;

        const productsData: Product[] = await client.fetch(query);
        setProducts(productsData);
      } catch (error) {
        console.error('Error fetching products:', error);
      }
    };

    fetchProducts();
  }, [selectedCategory]);

  return (
    <div className="md:max-w-[1440px] w-full h-auto mx-auto">
      {/* Dropdown to Filter by Category */}
      <div className="w-[1280px] mx-auto my-4">
      <select
  value={categories.find((category) => category._id === selectedCategory)?.slug.current}
  onChange={handleCategoryChange}
  className="border p-2 rounded"
>
  <option value="">All Categories</option>
  {categories.map((category) => (
    <option key={category._id} value={category.slug.current}>
      {category.name}
    </option>
  ))}
</select>

      </div>

      {/* New Ceramics Section */}
      <div>
        <div className="w-[1280px] mx-auto">
          <h1 className="text-4xl my-8">All Products</h1>
        </div>

        {/* Grid Container */}
        <div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-4 gap-6 px-4">
          {/* Map through the products */}
          {products.length > 0 ? (
            products.map((product) => (
              <Card key={product._id} 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>
            ))
          ) : (
            <p>No products found for this category.</p>
          )}
        </div>
      </div>
    </div>
  );
};

export default Ceramics;

```

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.