

Day 4

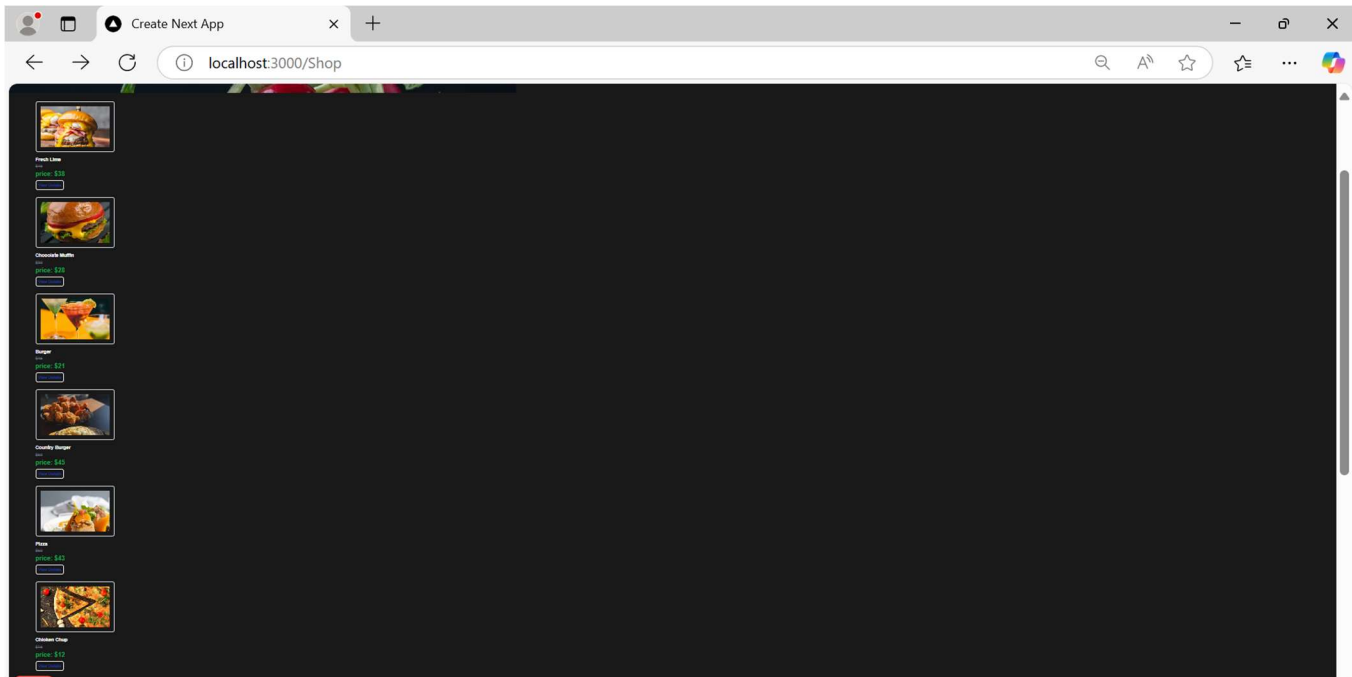
Objective:

The focus of today's task was to develop key frontend components for our marketplace application. The goal was to ensure these components are dynamic, reusable, and responsive, fetching data seamlessly from Sanity CMS or APIs.

Product Listing Component:

- Designed a grid layout to display products dynamically, including details such as:
 - **Name**
 - **Original Price**
 - **Price**
 - **Image**
 - **Product dimensions Link**

- Created a reusable ProductCard component to showcase each product. Data was retrieved using the Sanity CMS API.



Product Detail Component:

Developed dynamic product pages using Next.js dynamic routing.

These pages include:

- Product description.
- Pricing details.
- Large size product image
- Product name.



Out of Stock

Fresh Lime

Refreshing fresh lime drink made with natural ingredients.

Price: \$45

Category: Drink
Tag: HealthyPopular
Dicum/circusirises

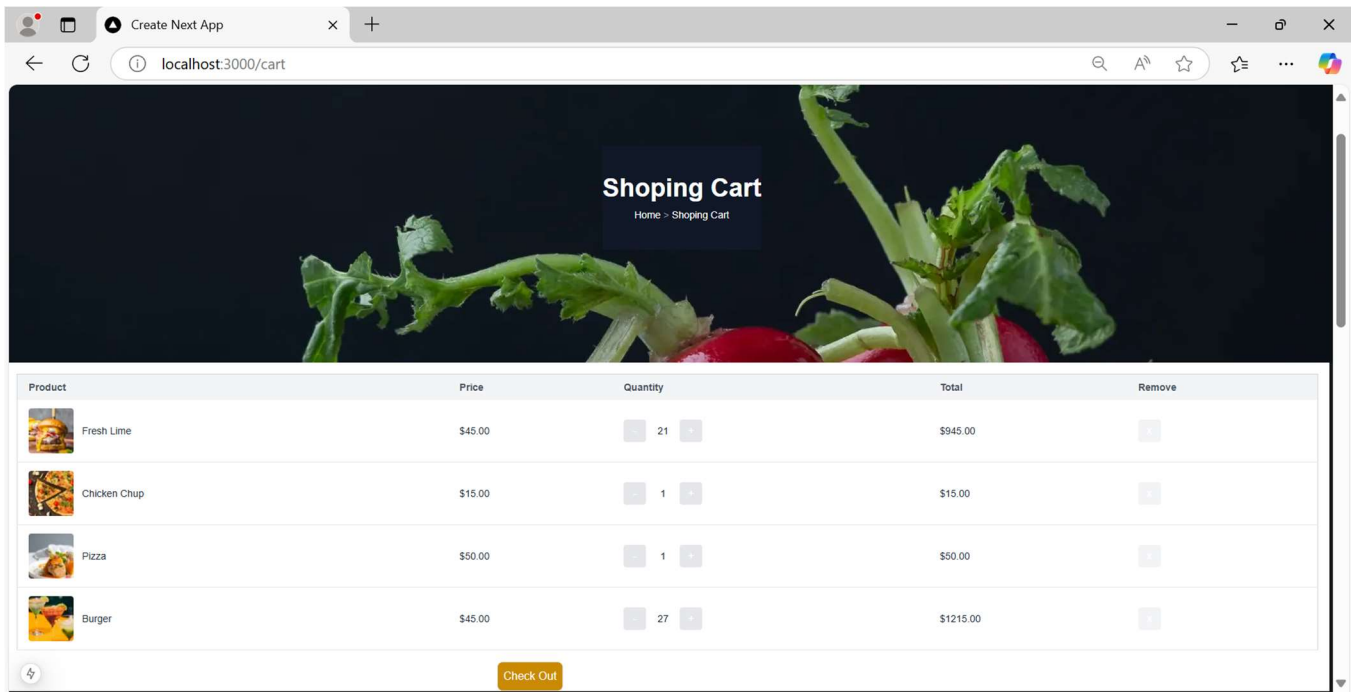
Add to cart

Cart Component:

- Built a functional cart that:

Displays added items, their quantities, and the total price.


- Used React state to track and manage cart items.



User Authentication Features:

- Developed a **Sign-Up Page** and a **Login Page** for user registration and login functionality using clerk.
- User can sign in through their **google** account.
- Created a **Profile Page** to display user-specific details.

Sign in to My app2
Welcome back! Please sign in to continue

 Continue with Google


or

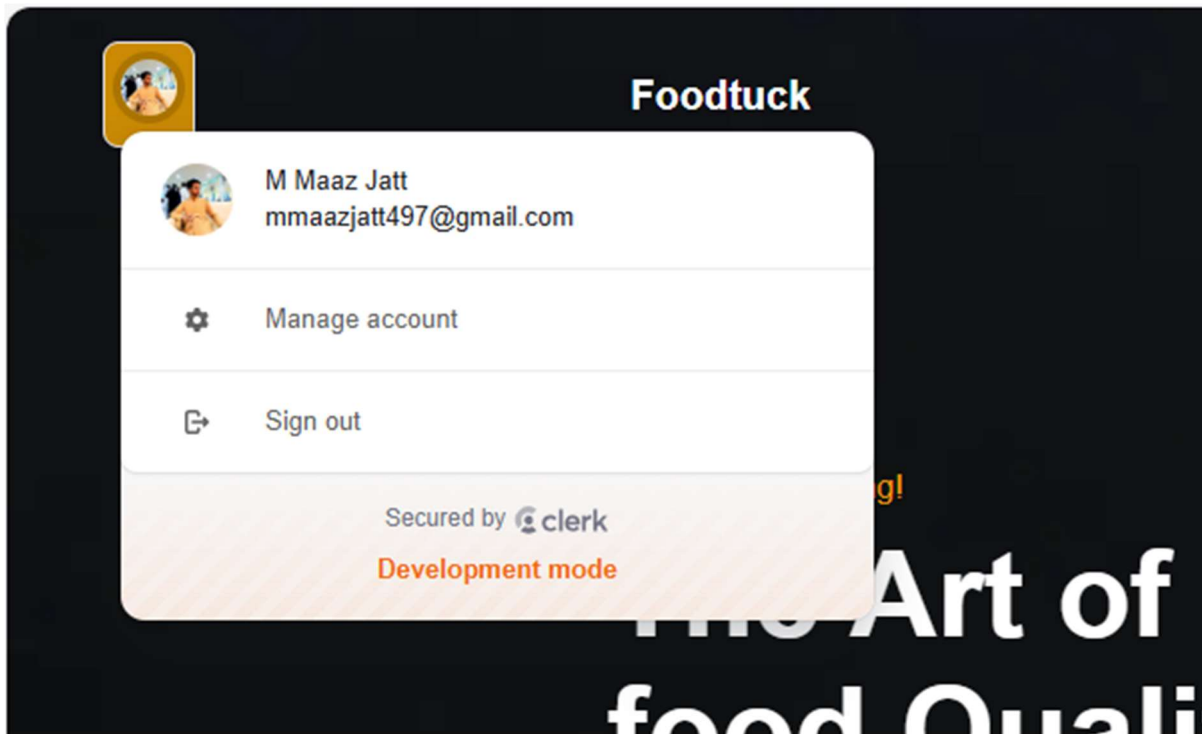
Email address or username [Use phone](#)

Enter email or username

Continue ▶

Don't have an account? [Sign up](#)

Secured by  clerk
Development mode



Checkout Flow Component:

I added a checkout flow component to the website to make the purchasing process more efficient.

The component gathers essential user information, including:

- First name
- Last name
- Address
- Cart items

All user data is securely stored on clerk to ensure reliability and scalability.

I also implemented an order confirmation page that:

- Retrieves the stored user data from Clerk.
- Shows a personalized thank-you message with the user's name.
- Displays the user's entered details and a summary of the ordered products.

This flow streamlines the checkout experience and provides a personal touch to improve the overall user experience.

Check out page:

Checkout

https://checkout.stripe.com/c/pay/cs_test_b1by4fcVWUh1bKxe0MqbRTxuYg4PXL3uDeCGL6JsUjIQtHwl21Mo5zpoF9#fidkdWxOYHwnPyd1b...

TEST MODE

Pay

\$920.00

Fresh Lime

Qty 7

\$315.00

\$45.00 each

Chicken Chup

Qty 1

\$15.00

Pizza

Qty 1

\$50.00

Burger

Qty 12

\$540.00

\$45.00 each

Pay with link

Or pay with card

Email

Card information

1234 1234 1234 1234

VISA

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

Payment confirmation page:

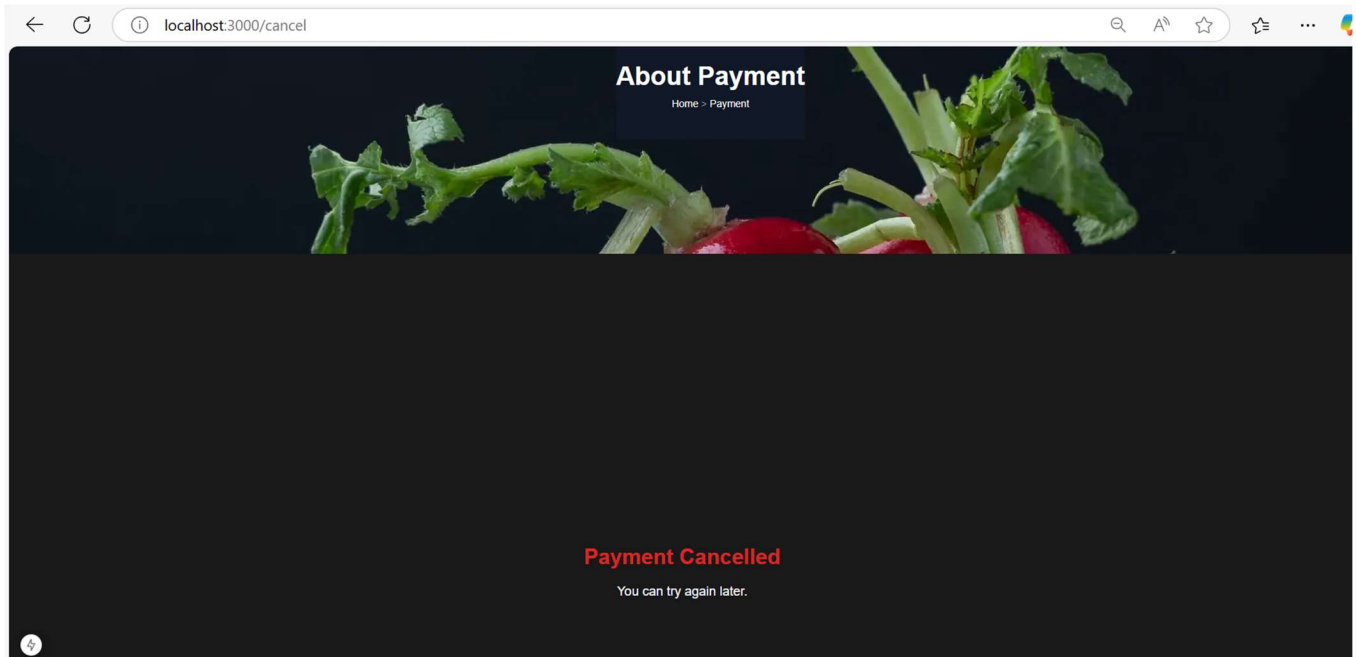
About Payment

Home > Payment

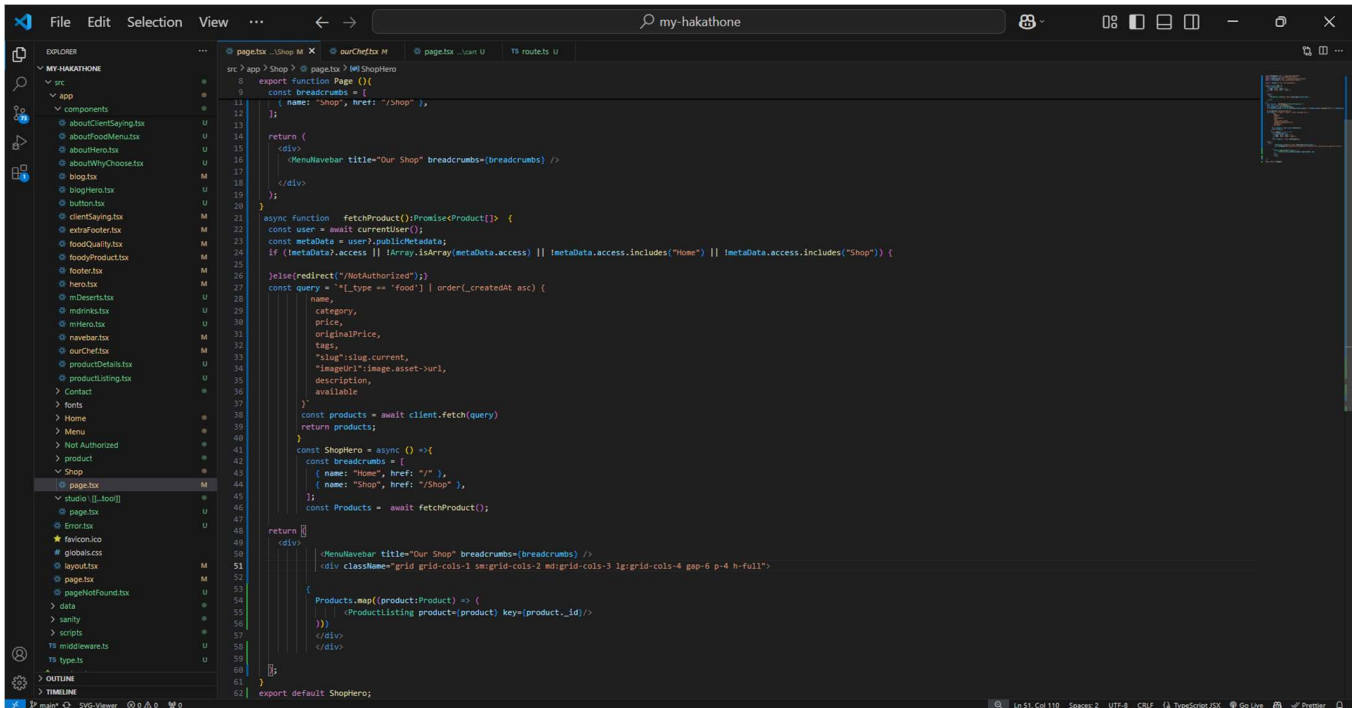
Payment Successful!

Thank you for your purchase.

Payment Cancellation page:



Product list component code snippet



```
8 export function Page() {
9   const breadcrumbs = [
10     { name: "Shop", href: "/shop" },
11   ];
12 }
13
14 return (
15   <div>
16     <Navbar title="Our Shop" breadcrumbs={breadcrumbs} />
17   </div>
18 </div>
19 )
20 }
21
22 async function fetchProduct(): Promise<Product[]> {
23   const user = await currentUser();
24   const metaData = user?.publicMetadata;
25   if (!metaData?.access || !Array.isArray(metaData.access) || !metaData.access.includes("Home") || !metaData.access.includes("Shop")) {
26     return redirect("/NotAuthorized");
27   }
28   const query = `[_type == "food" | order(_createdAt asc) {
29     name,
30     category,
31     price,
32     originalPrice,
33     tag,
34     "tagging":current,
35     "imageurl":image.asset->url,
36     description,
37     available
38   }`
39   const products = await client.fetch(query)
40   return products;
41 }
42
43 const ShopHero = async () => {
44   const breadcrumbs = [
45     { name: "Home", href: "/" },
46     { name: "Shop", href: "/shop" },
47   ];
48   const Products = await fetchProduct();
49 }
50
51 return (
52   <div>
53     <Navbar title="Our Shop" breadcrumbs={breadcrumbs} />
54     <div className="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-6 p-4 h-full">
55       {
56         Products.map((product: Product) => {
57           <ProductListing product={product} key={product._id} />
58         })
59       }
60     </div>
61   </div>
62 )
63 }
64
65 export default ShopHero;
```

Product details component code snippet

```
EDITOR
  page1x ~Shop M
  page1x ~Sales U
  page1x ~Sales U
  productDetails U X

  src app > components > @ productDetails U ...
  18
  19 const ProductDetails = ({ product }: { product: Product }) => {
  20
  21
  22
  23
  24
  25
  26
  27 const handleClick = (product: Product) => {
  28   // Parse the current cart from localStorage
  29   const cart = JSON.parse(localStorage.getItem('cart') || '{}');
  30
  31   // If the product already exists in the cart, update the quantity
  32   if (cart[product.name]) {
  33     cart[product.name].quantity += 1; // Correctly increment quantity by 1
  34   } else {
  35     // If the product doesn't exist in the cart, add it with quantity 1
  36     cart[product.name] = { ...product, quantity: 1 };
  37   }
  38
  39   // Save the updated cart back to localStorage
  40   localStorage.setItem('cart', JSON.stringify(cart));
  41 }
  42
  43 const breadcrumbs = [
  44   { name: 'Home', href: '/' },
  45   { name: 'Shop Details', href: '' },
  46 ];
  47
  48 return (
  49   <div>
  50     <MenuNavbar title="Shop Details" breadcrumbs={breadcrumbs} />
  51     <div className="w-[1920px] h-[100px] bg-white">
  52       <div className="flex flex-col md:flex-row gap-8 p-4">
  53         <div>
  54           <img alt={product.image} />
  55           <div>
  56             <h3>{product.name}</h3>
  57             <p>{product.description}</p>
  58             <p>Price: ${product.originalPrice}</p>
  59             <p>{product.stockLevel} > 0 ? 'In Stock' : 'Out of Stock'</p>
  60             <p>{product.category}</p>
  61             <p>{product.tags}</p>
  62             <p>{product.currency}</p>
  63             <p>{product.link}</p>
  64             <p>{product.error}</p>
  65             <p>{product.global}</p>
  66             <p>{product.outline}</p>
  67             <p>{product.summary}</p>
  68             <p>{product.details}</p>
  69             <p>{product.footer}</p>
  70             <p>{product.footer}</p>
  71             <p>{product.footer}</p>
  72             <p>{product.footer}</p>
  73             <p>{product.footer}</p>
  74             <p>{product.footer}</p>
  75             <p>{product.footer}</p>
  76             <p>{product.footer}</p>
  77             <p>{product.footer}</p>
  78             <p>{product.footer}</p>
  79             <p>{product.footer}</p>
  80             <p>{product.footer}</p>
  81             <p>{product.footer}</p>
  82             <p>{product.footer}</p>
  83             <p>{product.footer}</p>
  84             <p>{product.footer}</p>
  85             <p>{product.footer}</p>
  86             <p>{product.footer}</p>
  87             <p>{product.footer}</p>
  88             <p>{product.footer}</p>
  89             <p>{product.footer}</p>
  90             <p>{product.footer}</p>
  91             <p>{product.footer}</p>
  92             <p>{product.footer}</p>
  93             <p>{product.footer}</p>
  94             <p>{product.footer}</p>
  95             <p>{product.footer}</p>
  96             <p>{product.footer}</p>
  97             <p>{product.footer}</p>
  98             <p>{product.footer}</p>
  99             <p>{product.footer}</p>
  100             <p>{product.footer}</p>
  101             <p>{product.footer}</p>
  102             <p>{product.footer}</p>
  103             <p>{product.footer}</p>
  104             <p>{product.footer}</p>
  105             <p>{product.footer}</p>
  106             <p>{product.footer}</p>
  107             <p>{product.footer}</p>
  108             <p>{product.footer}</p>
  109             <p>{product.footer}</p>
  110             <p>{product.footer}</p>
  111             <p>{product.footer}</p>
  112             <p>{product.footer}</p>
  113             <p>{product.footer}</p>
  114             <p>{product.footer}</p>
  115             <p>{product.footer}</p>
  116             <p>{product.footer}</p>
  117             <p>{product.footer}</p>
  118             <p>{product.footer}</p>
  119             <p>{product.footer}</p>
  120             <p>{product.footer}</p>
  121             <p>{product.footer}</p>
  122             <p>{product.footer}</p>
  123             <p>{product.footer}</p>
  124             <p>{product.footer}</p>
  125             <p>{product.footer}</p>
  126             <p>{product.footer}</p>
  127             <p>{product.footer}</p>
  128             <p>{product.footer}</p>
  129             <p>{product.footer}</p>
  130             <p>{product.footer}</p>
  131             <p>{product.footer}</p>
  132             <p>{product.footer}</p>
  133             <p>{product.footer}</p>
  134             <p>{product.footer}</p>
  135             <p>{product.footer}</p>
  136             <p>{product.footer}</p>
  137             <p>{product.footer}</p>
  138             <p>{product.footer}</p>
  139             <p>{product.footer}</p>
  140             <p>{product.footer}</p>
  141             <p>{product.footer}</p>
  142             <p>{product.footer}</p>
  143             <p>{product.footer}</p>
  144             <p>{product.footer}</p>
  145             <p>{product.footer}</p>
  146             <p>{product.footer}</p>
  147             <p>{product.footer}</p>
  148             <p>{product.footer}</p>
  149             <p>{product.footer}</p>
  150             <p>{product.footer}</p>
  151             <p>{product.footer}</p>
  152             <p>{product.footer}</p>
  153             <p>{product.footer}</p>
  154             <p>{product.footer}</p>
  155             <p>{product.footer}</p>
  156             <p>{product.footer}</p>
  157             <p>{product.footer}</p>
  158             <p>{product.footer}</p>
  159             <p>{product.footer}</p>
  160             <p>{product.footer}</p>
  161             <p>{product.footer}</p>
  162             <p>{product.footer}</p>
  163             <p>{product.footer}</p>
  164             <p>{product.footer}</p>
  165             <p>{product.footer}</p>
  166             <p>{product.footer}</p>
  167             <p>{product.footer}</p>
  168             <p>{product.footer}</p>
  169             <p>{product.footer}</p>
  170             <p>{product.footer}</p>
  171             <p>{product.footer}</p>
  172             <p>{product.footer}</p>
  173             <p>{product.footer}</p>
  174             <p>{product.footer}</p>
  175             <p>{product.footer}</p>
  176             <p>{product.footer}</p>
  177             <p>{product.footer}</p>
  178             <p>{product.footer}</p>
  179             <p>{product.footer}</p>
  180             <p>{product.footer}</p>
  181             <p>{product.footer}</p>
  182             <p>{product.footer}</p>
  183             <p>{product.footer}</p>
  184             <p>{product.footer}</p>
  185             <p>{product.footer}</p>
  186             <p>{product.footer}</p>
  187             <p>{product.footer}</p>
  188             <p>{product.footer}</p>
  189             <p>{product.footer}</p>
  190             <p>{product.footer}</p>
  191             <p>{product.footer}</p>
  192             <p>{product.footer}</p>
  193             <p>{product.footer}</p>
  194             <p>{product.footer}</p>
  195             <p>{product.footer}</p>
  196             <p>{product.footer}</p>
  197             <p>{product.footer}</p>
  198             <p>{product.footer}</p>
  199             <p>{product.footer}</p>
  200             <p>{product.footer}</p>
  201             <p>{product.footer}</p>
  202             <p>{product.footer}</p>
  203             <p>{product.footer}</p>
  204             <p>{product.footer}</p>
  205             <p>{product.footer}</p>
  206             <p>{product.footer}</p>
  207             <p>{product.footer}</p>
  208             <p>{product.footer}</p>
  209             <p>{product.footer}</p>
  210             <p>{product.footer}</p>
  211             <p>{product.footer}</p>
  212             <p>{product.footer}</p>
  213             <p>{product.footer}</p>
  214             <p>{product.footer}</p>
  215             <p>{product.footer}</p>
  216             <p>{product.footer}</p>
  217             <p>{product.footer}</p>
  218             <p>{product.footer}</p>
  219             <p>{product.footer}</p>
  220             <p>{product.footer}</p>
  221             <p>{product.footer}</p>
  222             <p>{product.footer}</p>
  223             <p>{product.footer}</p>
  224             <p>{product.footer}</p>
  225             <p>{product.footer
```

The screenshot shows a Visual Studio Code window titled "my-hakathone". On the left, the Explorer sidebar displays a project structure:

- MY-HAKATHONE**
 - src**
 - app**
 - components**
 - Blog.jsx
 - blogHero.jsx
 - button.jsx
 - clientSaying.jsx
 - extraFooter.jsx
 - foodQuality.jsx
 - foodyProduct.jsx
 - footer.jsx
 - hero.jsx
 - mDeserts.jsx
 - mDrinks.jsx
 - mHeroes.jsx
 - navebar.jsx
 - ourChef.jsx
 - productDetails.jsx
 - productListing.jsx
 - Contact
 - fonts
 - Home
 - Menu
 - Not Authorized
 - product [slug]
 - page.jsx**
 - Shop
 - studio [!-tool]
 - page.jsx
 - Error.jsx
 - faviconico
 - globals.css
 - layout.jsx
 - page.jsx
 - pageNotFound.jsx
 - data
 - outline
 - timeline

The main editor area shows the code for `page.jsx`:

```
src > app > product > [slug] > page.jsx > ...
1 import ProductDetails from '@app/components/productDetails';
2 import { client } from '@sanity/lib/client';
3 import React from 'react';
4
5 async function getProduct(slug: string): Promise<Product> {
6   const query = `[_type == 'food' && slug.current == "${slug}"] | order(_createdAt asc) {
7     _id,
8     name,
9     category,
10    originalPrice,
11    "discountPrice": price, // Assuming the discount price is stored in 'price'
12    tags,
13    "slug": slug.current,
14    "imageId": image.asset->url,
15    description,
16    available
17  };
18
19  // Fetch data from Sanity and return the first product
20  const products = await client.fetch(query);
21  return products[0]; // Return only the first product since we're expecting a single result
22 }
23
24 const ProductDetail = async ({ params }: { params: { slug: string } }) => {
25   const product = await getProduct(params.slug);
26
27   return (
28     <div>
29       {/* Render the product details for the fetched product */}
30       <ProductDetails product={product} key={product._id} />
31     </div>
32   );
33 };
34
35 export default ProductDetail;
```

