

1. Functional Deliverables:

Screenshots or screen recordings showcasing product listing page, product detail pages and Working category filters, search bar.

Screen Recording Link:

<https://www.loom.com/share/77f656302ee248a3b62c4c702f4568ab?sid=5c8d7591-7f93-410b-8471-6982252a50f6>

2. Code Deliverables:

Product List Page

```
import React from "react";
import ProductImages from "../ProductImages";
import SingleProductDetail from "../SingleProductDetail";
import ProductDesc from "../ProductDesc";
import SimilarProducts from "../SimilarProducts";
import { client } from "@sanity/lib/client";

interface Iproduct {
  name: string;
  description: string;
  category: string;
  imageUrl: string;
  price: number;
  originalPrice: number;
}

const ProductDetail = async ({ slug }: { slug: string }) => {
  const query = `*[ _type == "food" && slug.current == "${slug}" ]{
    name,description,category,price, originalPrice,
    "imageUrl":image.asset->url
  }`;
  const data: Iproduct = await client.fetch(query);
  return (
    <div className="max-w-screen-xl mx-auto py-4 px-2">
      <div className="flex flex-col sm:flex-row gap-3 mb-5">
        <div className="flex-1">
          <Image alt={data.imageUrl} />
        </div>
        <div className="flex-1">
          <ProductDetail
            category={data.category}
            desc={data.description}
            name={data.name}
            price={data.price}
            originalPrice={data.originalPrice}
          />
        </div>
      </div>
      <div>
        <ProductDesc />
      </div>
      <div>
        <SimilarProducts category={data.category} />
      </div>
    </div>
  );
};

export default ProductDetail;
```

Product Detail Page

```
import React from "react";
import Card from "../Card";
import { Search } from "lucide-react";

import { Inter } from "next/font/google";
import Categories from "../Categories";

import Link from "next/link";
import { client } from "@sanity/lib/client";
const inter = Inter({ subsets: ["latin"] });
export interface IProduct {
  name: string;
  available: boolean;
  currentSlug: string;
  price: number;
  originalPrice: number;
  imageUrl: string;
}

const Products = async () => {
  const query = `*[ _type == "food" ]{
    name,available,"currentSlug": slug.current,
    "imageUrl": image.asset->url,price,originalPrice
  }`;

  const data: IProduct[] = await client.fetch(query);

  return (
    <div className="max-w-screen-xl mx-auto py-4 px-4">
      <div className="
flex flex-col lg:flex-row justify-between w-full gap-3">
        <div className="
flex flex-wrap gap-2 h-max w-full lg:w-2/3">
          {data.map((product: IProduct, i: number) => (
            <div key={i} className="">
              <Link href={` /shop/${product.currentSlug} `}>
                <Card
                  name={product.name}
                  image={product.imageUrl}
                  dprice={product.price}
                  aprice={product.originalPrice}
                />
              </Link>
            </div>
          ))}
        </div>
        <div className="w-full lg:w-1/3 border py-2 px-4">
          <form action="" className="flex justify-between">
            <input
              type="text"
              placeholder="Search Product"
              className={
                `bg-[#FF9F0D] bg-opacity-10 w-full text-[16px] font-normal ${
                  inter.className} p-2`
              }
            />
            <div className="
bg-[#FF9F0D] flex justify-center items-center px-2">
              <Search />
            </div>
          </form>
          <Categories />
        </div>
      </div>
    </div>
  );
};

export default Products;
```

Search Form Code

```
"use client"; // Enable client-side interactivity
import { Search } from "lucide-react";
import { useRouter } from "next/navigation";
import React, { useState } from "react";

const SearchForm = () => {
  const [query, setQuery] = useState("");
  const router = useRouter();

  const handleSubmit = (e: React.FormEvent) => {
    e.preventDefault();
    if (query.trim()) {
      router.push(`/${query}`);
    }
  };

  return (
    <Form
      onSubmit={handleSubmit}
      className="flex items-center space-x-2 border rounded-full"
    >
      <input
        type="text"
        placeholder="Search products..."
        value={query}
        onChange={(e) => setQuery(e.target.value)}
        className="p-2 bg-transparent rounded-md outline-none"
      />
      <button type="submit" className="px-2">
        <Search />
      </button>
    </form>
  );
};

export default SearchForm;
```

Search Page and Logic for fetch Data

```
import Card from "@components/Card";
import Header from "@components/Header";
import Hero from "@components/Hero";
import { IProduct } from "@components/Products";
import { client } from "@sanity/lib/client";
import banner from "@../public/unsplash_4ycv3Ky1ZZU.png";
import Link from "next/link";

// Corrected function signature
export default async function Page({
  params,
}: {
  params: Promise<{ name: string }>;
}) {
  const { name } = await params; // Get name from params

  // Fetch search results from Sanity using query with variable
  const products: IProduct[] = await client.fetch(
    `*[_type == "food" && name match $name]{
      name, description, category, price, originalPrice,
      "currentSlug": slug.current,
      "imageUrl": image.asset->url
    }`,
    { name } // Safe way to pass query variable
  );

  return (
    <main>
      <section className="max-w-screen-xl mx-auto">
        <Header />
        <Hero image={banner} heading="Search" path="search" />
        <h1 className="text-2xl font-bold mt-5">Search Results for: {name}</h1>
        {products.length > 0 ? (
          <div className="grid grid-cols-1 md:grid-cols-3 gap-4 mt-4">
            {products.map((product) => (
              <Link
                href={` /shop/${product.currentSlug}`}
                key={product.currentSlug}
              >
                <Card
                  dprice={product.originalPrice}
                  image={product.imageUrl}
                  name={product.name}
                  aprice={product.price}
                />
              </Link>
            ))}
          </div>
        ) : (
          <p className="text-gray-500 mt-4">No products found.</p>
        )}
      </section>
    </main>
  );
}
```

3. Documentation:

Steps taken to build and integrate components.

- 1) Product List Page:** The product in the list page has been fetched from sanity by writing the query which is mentioned in the above code of ProductList Page.

- 2) Product Detail Page:** The Single product is fetched by using the slug. The params slug is checked and compare with the product slug and when these two are matched, then that product is fetched from sanity.
- 3) Search Form and Search Result:** This Search is working the same as product detail page. Input field value is gotten through params and match that with product name and when it is matched with those products in the sanity, the sanity return all those matched products.

Challenges faced and solutions implemented.

I have faced one challenge implementing these components. Which is because of params. The error is of Type “Type Error”. This error comes because I am using Latest next.js version which is Next.js 15. Define type of param is a promise which is resolved by wrapping type in a promise such as: `Promise<{slug: string}>`