Hackathon 3

Day 3

My Project Overview: Dabbe Me Dabba

I’m building an e-commerce website called "Dabbe Me Dabba," where I can sell customizable gift boxes and surprise items. For this project, I’m using **Next.js** as my web framework, **Sanity** as my content management system (CMS), and **Tailwind CSS** for styling.

What I’ve Created in Sanity

In Sanity, I set up a product schema, which serves as a blueprint for all the products I want to sell. Here’s what it includes:

1. **Product Name**: Each product has a name so that customers know what they’re buying.
2. **Product Price**: I can set a price for each product, which will be displayed in Pakistani Rupees (PKR).
3. **Custom ID**: I have the option to give each product a unique custom ID if I need to track them in a special way.
4. **Product Image**: I can upload an image for each product, and Sanity allows me to crop and focus on the best part of the image.

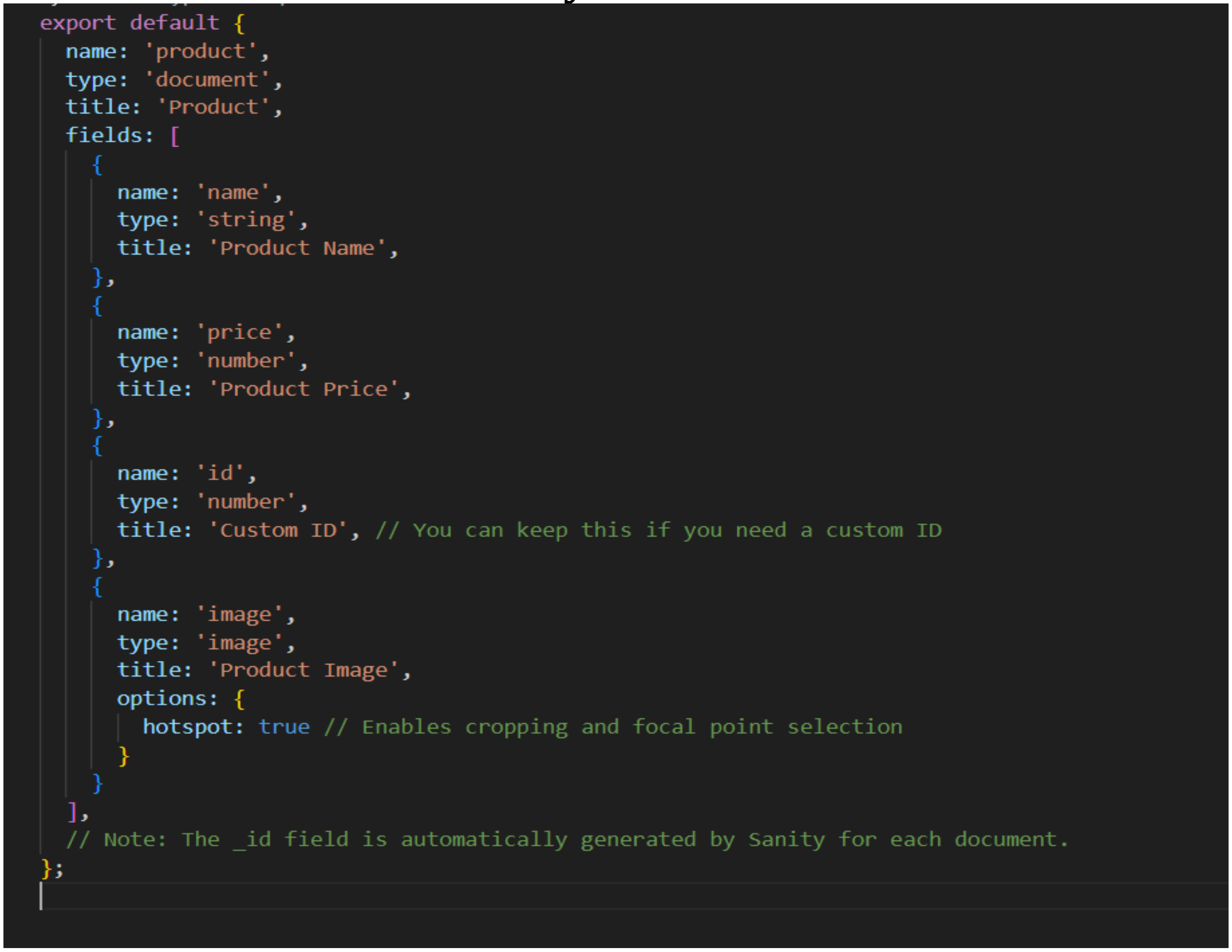
How It Works

- **Product Management**: I can easily add, edit, or remove products in Sanity, and any changes I make will show up on my website without needing to redeploy.
- **Dynamic Display**: The site fetches product data from Sanity, keeping everything up-to-date and reflecting what I have available for sale.
- **Responsive Design**: I designed the website to look good on both desktop and mobile devices, making it easy for customers to browse and buy products.

In Summary

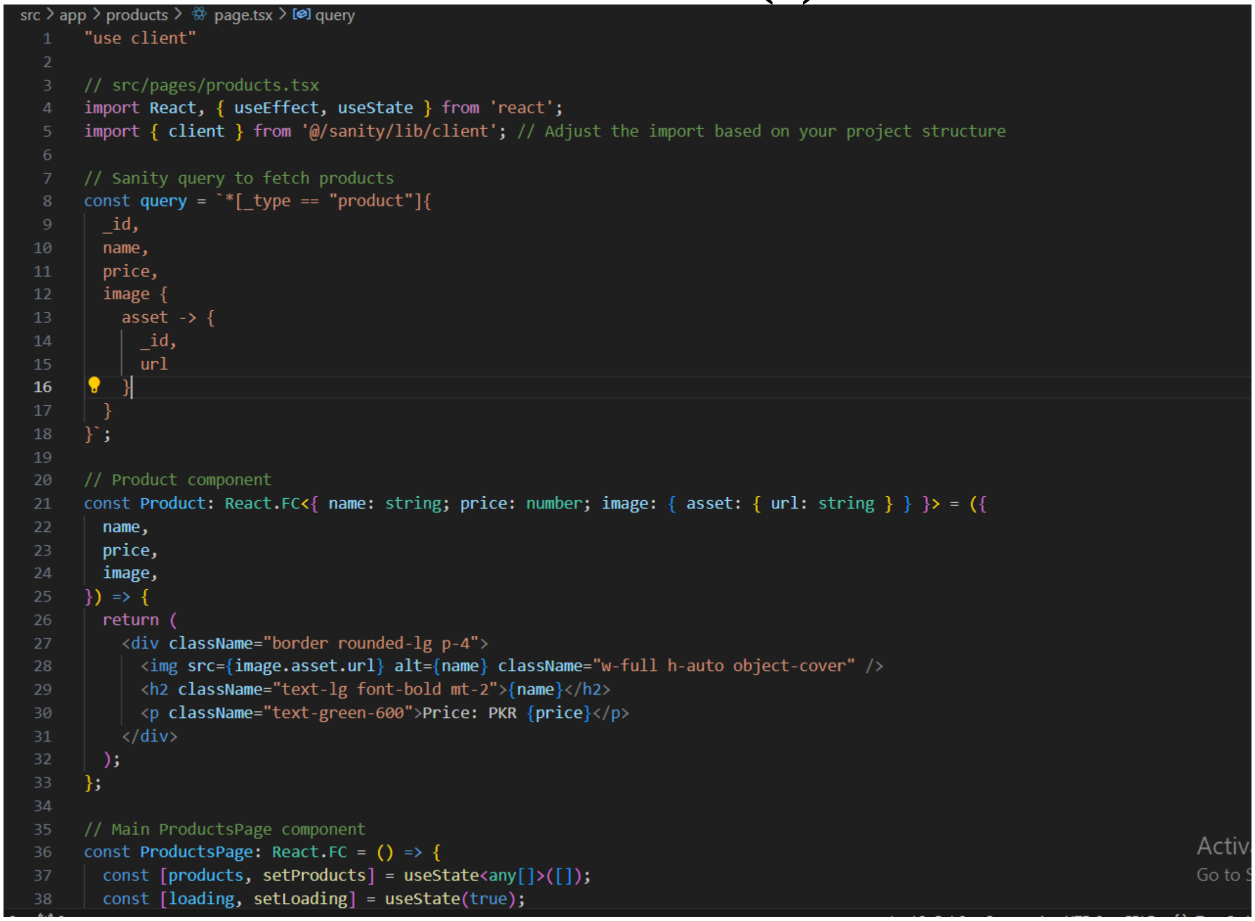
My "Dabbe Me Dabba" project is all about creating a fun, colorful marketplace for unique gifts. By using Sanity to manage my products, I’ve set up a system that makes it easy to keep my site fresh and engaging for customers. The combination of Next.js and Tailwind CSS ensures that the site is fast and visually appealing.

Sanity Schema



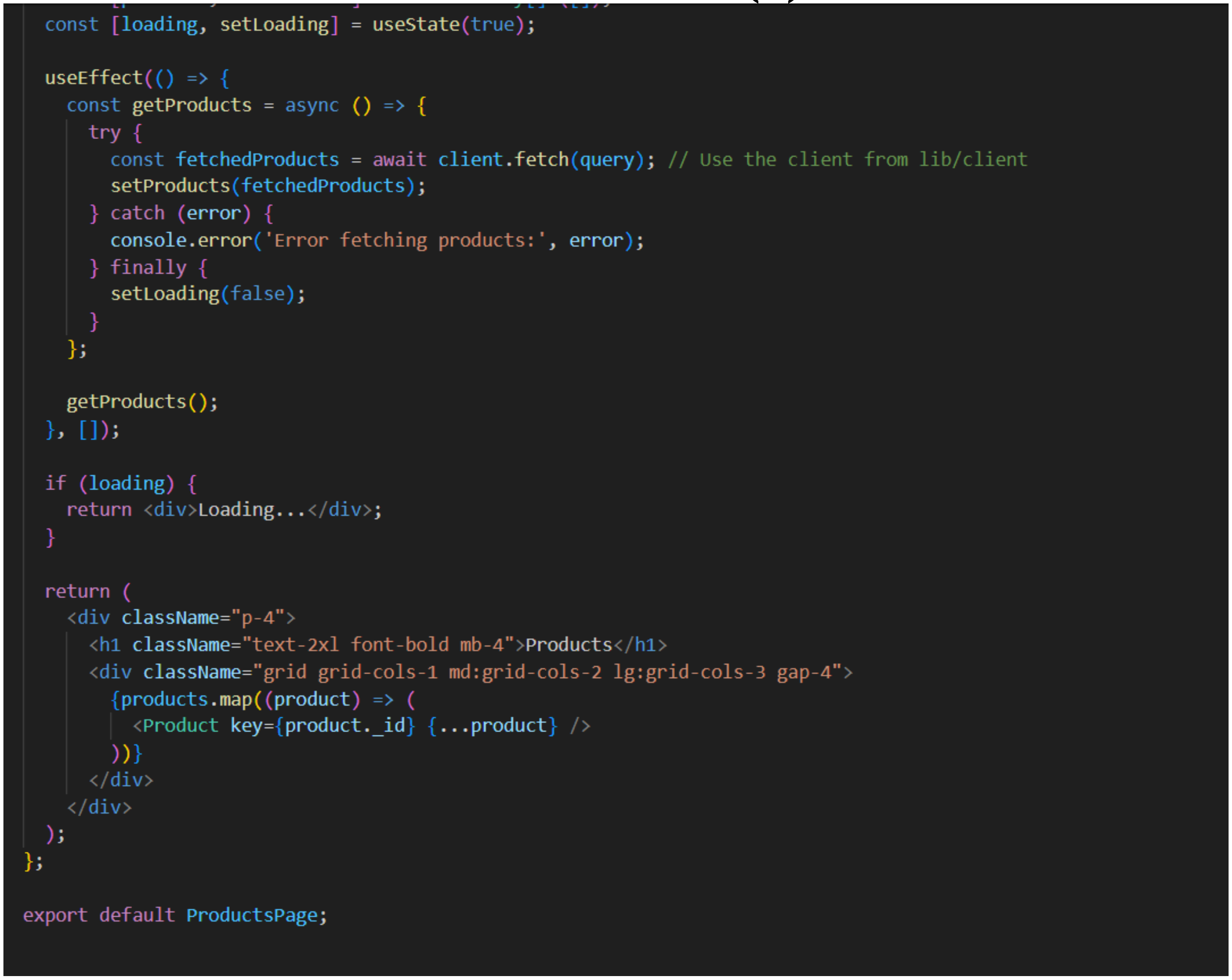
```
export default {
  name: 'product',
  type: 'document',
  title: 'Product',
  fields: [
    {
      name: 'name',
      type: 'string',
      title: 'Product Name',
    },
    {
      name: 'price',
      type: 'number',
      title: 'Product Price',
    },
    {
      name: 'id',
      type: 'number',
      title: 'Custom ID', // You can keep this if you need a custom ID
    },
    {
      name: 'image',
      type: 'image',
      title: 'Product Image',
      options: {
        hotspot: true // Enables cropping and focal point selection
      }
    }
  ],
  // Note: The _id field is automatically generated by Sanity for each document.
};
```

Product Code (1)



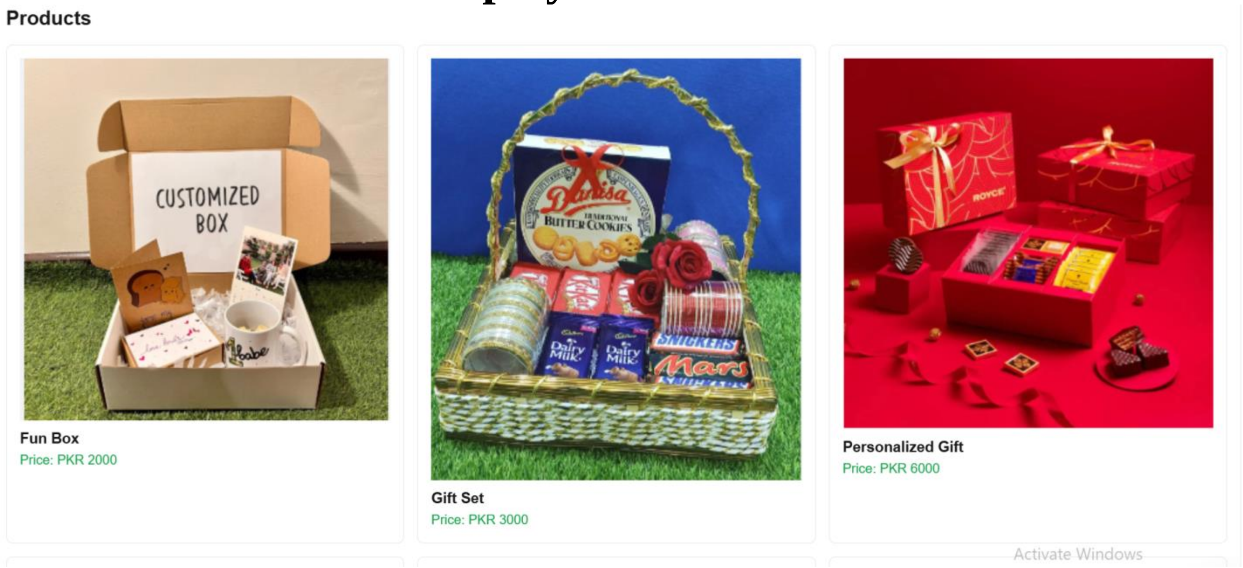
```tsx
"use client"

// src/pages/products.tsx
import React, { useEffect, useState } from 'react';
import { client } from '@/sanity/lib/client'; // Adjust the import based on your project structure

// Sanity query to fetch products
const query = `*[_type == "product"]{
  _id,
  name,
  price,
  image {
    asset -> {
      _id,
      url
    }
  }
}`;

// Product component
const Product: React.FC<{ name: string; price: number; image: { asset: { url: string } } }> = ({
  name,
  price,
  image,
}) => {
  return (
    <div className="border rounded-lg p-4">
      <img src={image.asset.url} alt={name} className="w-full h-auto object-cover" />
      <h2 className="text-lg font-bold mt-2">{name}</h2>
      <p className="text-green-600">Price: PKR {price}</p>
    </div>
  );
};

// Main ProductsPage component
const ProductsPage: React.FC = () => {
  const [products, setProducts] = useState<any[]>([]);
  const [loading, setLoading] = useState(true);
```

Product Code (2)



```
  const [loading, setLoading] = useState(true);

  useEffect(() => {
    const getProducts = async () => {
      try {
        const fetchedProducts = await client.fetch(query); // Use the client from lib/client
        setProducts(fetchedProducts);
      } catch (error) {
        console.error('Error fetching products:', error);
      } finally {
        setLoading(false);
      }
    };

    getProducts();
  }, []);

  if (loading) {
    return <div>Loading...</div>;
  }

  return (
    <div className="p-4">
      <h1 className="text-2xl font-bold mb-4">Products</h1>
      <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-4">
        {products.map((product) => (
          <Product key={product._id} {...product} />
        ))}
      </div>
    </div>
  );
};

export default ProductsPage;
```

Display on Website



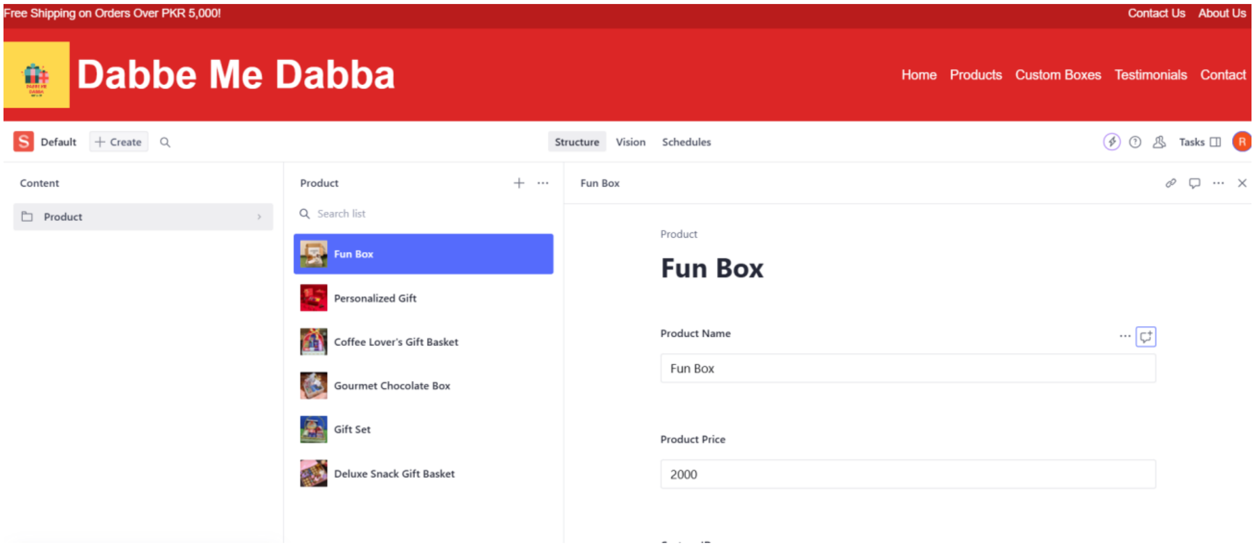
Products

Fun Box
Price: PKR 2000

Gift Set
Price: PKR 3000

Personalized Gift
Price: PKR 6000

Sanity Product Fetching



Free Shipping on Orders Over PKR 5,000!

Dabbe Me Dabba

Fun Box
Personalized Gift
Coffee Lover's Gift Basket
Gourmet Chocolate Box
Gift Set
Deluxe Snack Gift Basket

Fun Box

Product

Fun Box

Product Name: Fun Box

Product Price: 2000