Hackathon Day 3

API Integration Process Using Sanity

Sanity, a headless CMS, was employed to streamline content management and ensure smooth API integration. Below is an overview of the process:

1. Project Setup:

- o Created a new Sanity project via the Sanity CLI.
- Configured the sanity. Json file with project-specific details.
- This project involves developing an e-commerce store.
- The administrator has the capability to add products using the Sanity content management system.
- The /shop/[item] Route, situated within the app directory, is specifically designed to handle the management and display of product data.

2. Adjustments Made to Schemas

 Schema modifications were implemented to align with the website's content structure and ensure optimal performance. The Structure is showed in screenshots

Screenshots

Schema: product page

Schema Shop page:

```
export const shop={
    name: 'Shop',
    title: 'Shop Products',
    type: 'document',
    fields: [
            name: 'ProductID',
            type: 'string',
            title: 'Shop Product ID',
            name: 'ProductName',
            type: 'string',
            title: 'Shop Product Name',
            name: 'ProductImage',
            type: 'image',
            title: 'Shop Product Image',
            name: 'ProductPrice',
            type: 'string',
            title: 'Shop Product Price',
            name: 'ProductDescription',
            type: 'string',
            title: 'Shop Product Description',
            name: 'ProductDiscount',
            type: 'string',
            title: 'Shop Product Discount',
            name: 'ProductFeatured',
            type: 'boolean',
            title: 'Shop Product Is Featured',
            name: 'Productcategory',
            type: 'string',
            title: 'Shop Product Category',
```

3. Migration Steps and Tools Used

- Content migration was a critical phase to ensure the smooth transition of existing data into Sanity.
- o Making Schema in Sanity
- Creating Documents in Sanity Including Product Category and Shop products
- Using sanity client and declare some functions to fetch data from the Sanity Studio

API Calls:

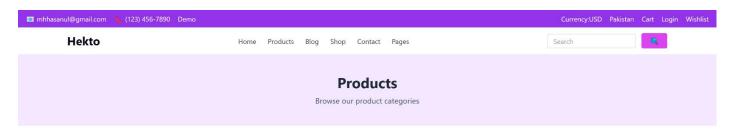
Product page:

API Call Shop Page

```
import Link from 'next/link';
lmport Header from '../Components/component1';
import Foot from '../Components/component2';
import Bottom1 from '../Components/component3';
import Image from 'next/image';
import ( client ) from '@/sanity/lib/client';
                         cdiv classhame="mb-6">
    ch3 classhame="text-md font-medium text-gray-600 mb-2">Categories</h3>
    cul classhame="space>y-1">
        cul classhame=
                                             {/* Products Section */}
<main className="flex-1 md:ml-8">
<ht className="text-x1 font-semibold text-gray-800 mb-4">Shop Products</h1></h1>
                                                  <div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-6">
{products.map((product: any) => (
                                                                      className="bg-white rounded-ig statute |
claim href=('/shop/$(product.ProductID)') passHref>
clay
clay
class
srce(product.imageUrl)
alt-(product.ProductName)
className="rounded-md w-full object-cover"
width=(300)
height=(200)
//
                                                                                ///
//dtv>
//Link>
/div className="mt-4">
//A className="mt-4">
//A className="mt-4">
//A className="mt-4">
//A className="text-lg font-semibold text-gray-700">
//A3
**Semibold text-gray-500 text-sm">(product
**InssNamen"mt-2 flex justify-between itc
**Or classNamen"text-gray-500 font-bold">
**Orproduct.Product)Iscount || product.Product)
**Or classNamen"text-gray-500 fine-through
**Or classNamen"text-gray-500 line-through
**Orproduct.ProductPrice)
**Orproduct.ProductPrice)
**Orproduct.ProductPrice)
**Orproduct.ProductPrice)
**Orproduct.ProductPrice)
**Ordroduct.ProductPrice)
**Ordroduct.Product.ProductPrice)
**Ordroduct.Product.Product.Product.Product.Product.Product.Product.Product.Product.Product.Product.Product.Product.Product.Product.Product.Product.Product.Product.Product.Product.Product.Product.Product.Product.Produ

className="text-gray-500 text-sm">{product.ProductDescription}
<div className="mt-2 flox justify-between items-center">
className="text-gray-800 font-bold">
$(product.ProductDiscount || product.ProductPrice)
```

Frontend Display:



Ecommerce Items

8 Product Categories









Phones

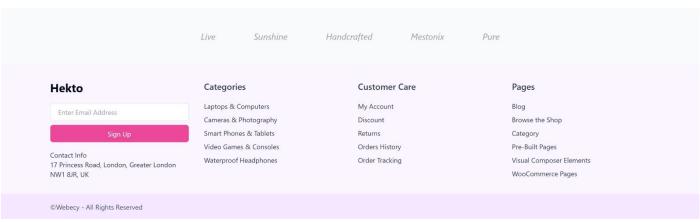






Furniture







Customer Care
My Account
Discount
Returns
Curley History
Onlier Tracking