GOLBAL MART ONLINE SHOP

DEV: MAHNOOR GHAFFAR

Hackathon Day 01

Business Idea

Marketplace Type
General E-Commerce

- A General e-commerce market place its is a digital shopping mall where each shop represents seller.
- However, many websites offer similar services discover what sets my website apart.

Unique Idea

- A Unique feature of my platform is that user can choose and hire riders to deliver their goods to nearby location.
- Additionally, it provides an excellent opportunity for bike riders to connect with and find delivery boy jobs easily.
- Reliable and timely delivery services.
- Competitive pricing , A diverse range of products

Target audience

Busy individuals who prefer online shopping over physical stores.

Individuals looking for delivery boy jobs.

Products or services we offer

Products :

Electronics, clothing, groceries, home, kitchen supplies, toys, beauty and wellness items.

Services:

Home delivery, Easy Return.

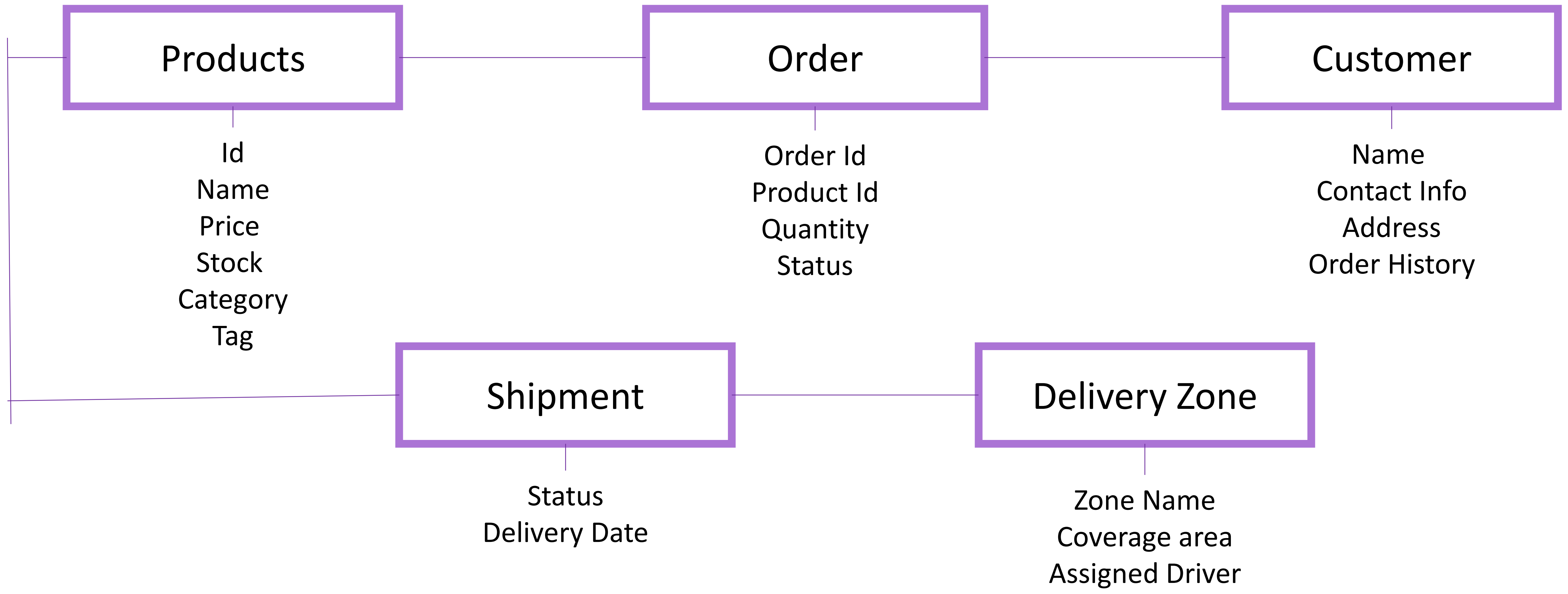
What will set Our Market place Apart?

Our platform offers faster delivery especially for local orders.

Customer support: 24/7 assistance for queries and returns.

As I mentioned earlier, the uniqueness of project is that user can hire local bike riders.

Schema


Products
Id
Name
Price
Stock
Category
Tag
Order
Order Id
Product Id
Quantity
Status
Customer
Name
Contact Info
Address
Order History
Shipment
Status
Delivery Date
Delivery Zone
Zone Name
Coverage area
Assigned Driver

Hackathon Day 02

- **Frontend :**
- Next.js Tailwind.css.
- Handel UI , dynamic routing state management (e.g Redux for cart).
- Integrates middleware and validation for secure operation.

- **Backend:**
- Manages products order and user data Handel curd operations and schema definition.

- **Third Party Api :**
- Stripe payment processing .
- Shipment api : tracking order delivery.
- Fetch data from third party api to populate sanity.

- **Functionalities Implemented :**

- User fill out the registration form.
- User data stored in sanity CMS.
- Products are fetched and displayed properly using a custom API with data stored in Sanity.
- Secure authentication for users.
- Ensures smooth navigation between pages.
- Secure and seamless payment processing.
- Users can track their orders.
- Allows users to leave reviews in real time.

Key Workflows

User Registration:

- User signIn data stored in sanity CMS.

- **Product Browsing**

- Users explore various product categories via the frontend.
- The frontend dynamically displays product listings.
- Sanity CMS API retrieves product information, including name, price, stock availability, description, and images.
- The frontend dynamically displays product listings.

- **Order Placement**

- Users add desired products to their cart and proceed to checkout.

- Payments are processed securely through Stripe.

- **Shipment Tracking**

- After the order is placed, shipment details are updated using ShipEngine.
- Real-time tracking information is made available to the user on the frontend.

Day 3 - API Integration Report

"This document outlines the process of integrating the API with Sanity CMS, migrating data, and displaying the imported data on the frontend using Next.js."

- **Steps Undertaken**

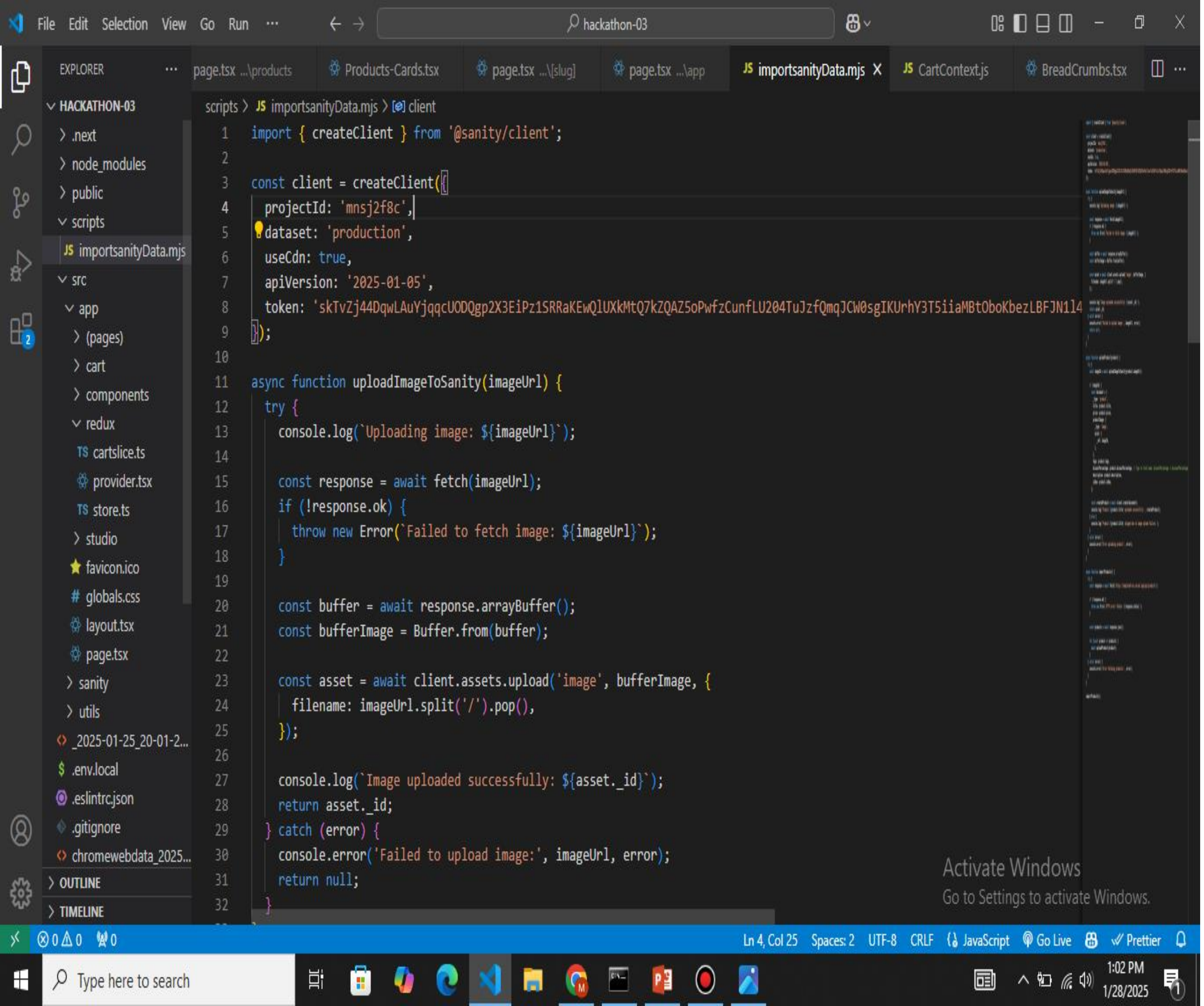
- **1. API Understanding**
- Api Endpoints :The product data was fetched from https://template6-six.vercel.app/api/products
- **Key fields identified**
- Item
- Price
- imageUrl
- Tags
- Discountpercentage
- Description
- Is new

Sanity Schema Example

```
import { defineType } from "sanity"

export const product = defineType({
  name: "product",
  title: "Product",
  type: "document",
  fields: [
    {
      name: "title",
      title: "Title",
      validation: (rule) => rule.required(),
      type: "string"
    },
    {
      name:"description",
      type:"text",
      validation: (rule) => rule.required(),
      title:"Description",
    },
```

```
{
      name: "productImage",
      type: "image",
      validation: (rule) => rule.required(),
      title: "Product Image"
    },
    {
      name: "price",
      type: "number",
      validation: (rule) => rule.required(),
      title: "Price",
    },
    {
      name: "tags",
      type: "array",
      title: "Tags",
      of: [{ type: "string" }]
    },
```

```
{
      name:"dicountPercentage",
      type:"number",
      title:"Discount Percentage",
    },
    {
      name:"isNew",
      type:"boolean",
      title:"New Badge",
    }
  ]
})
```



```
import { createClient } from '@sanity/client';

const client = createClient({
  projectId: 'mnsj2f8c',
  dataset: 'production',
  useCdn: true,
  apiVersion: '2025-01-05',
  token: 'skTvZj44DqwLAuYjqqcUODQgp2X3EiPz1SRRaKEwQlUXkMtQ7kZQAZ5oPwfzCunfLU204TuJzfQmqJCW0sgIKUrhY3T5iiaMBtOboKbezLBFJN1l4
});

async function uploadImageToSanity(imageUrl) {
  try {
    console.log(`Uploading image: ${imageUrl}`);

    const response = await fetch(imageUrl);
    if (!response.ok) {
      throw new Error(`Failed to fetch image: ${imageUrl}`);
    }

    const buffer = await response.arrayBuffer();
    const bufferImage = Buffer.from(buffer);

    const asset = await client.assets.upload('image', bufferImage, {
      filename: imageUrl.split('/').pop(),
    });

    console.log(`Image uploaded successfully: ${asset._id}`);
    return asset._id;
  } catch (error) {
    console.error('Failed to upload image:', imageUrl, error);
    return null;
  }
```

➢ Data Migration

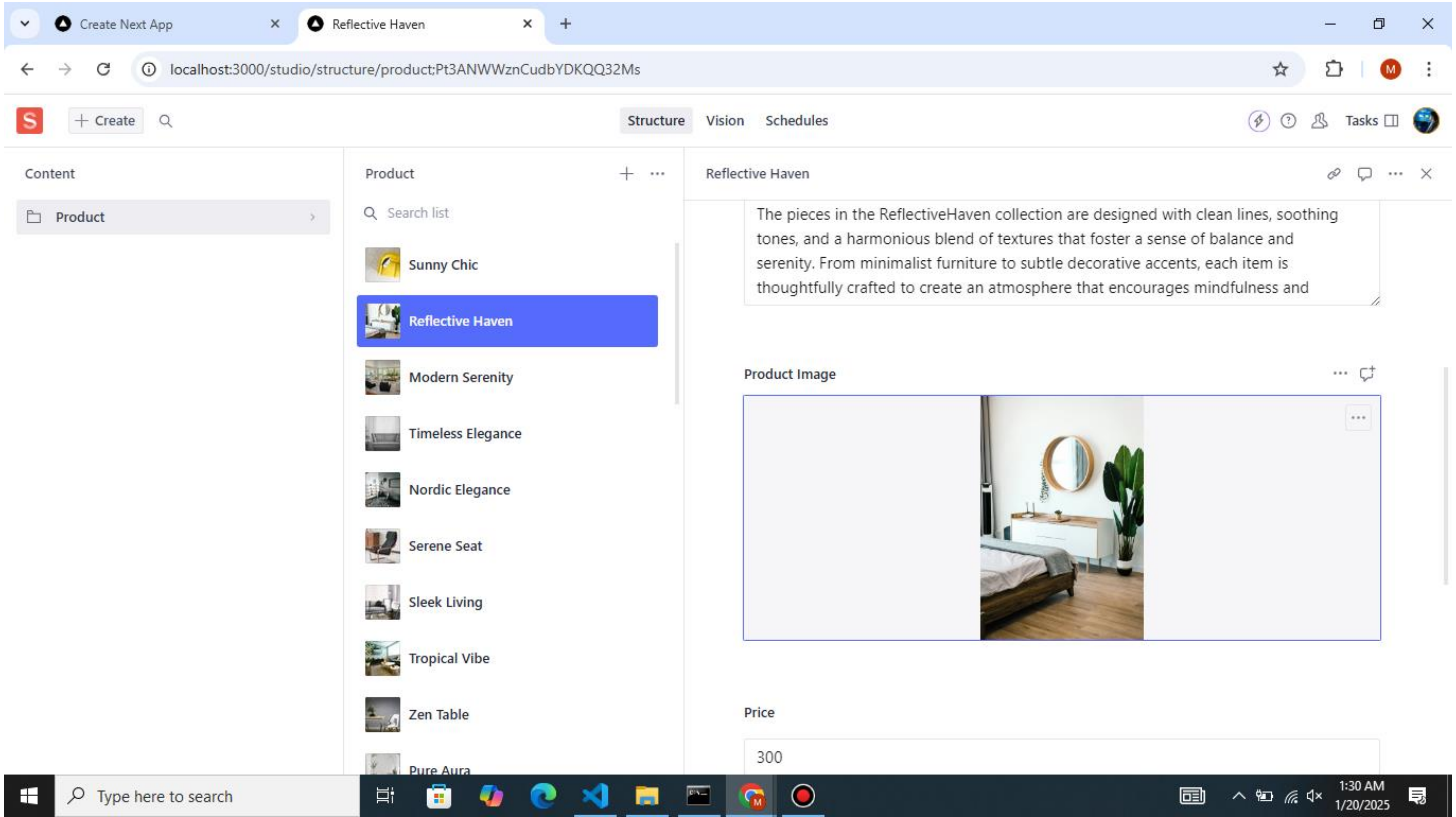
Script-Based Migration

Script: A custom script was developed to:
Fetch product data from the API.

Upload product images to Sanity CMS.

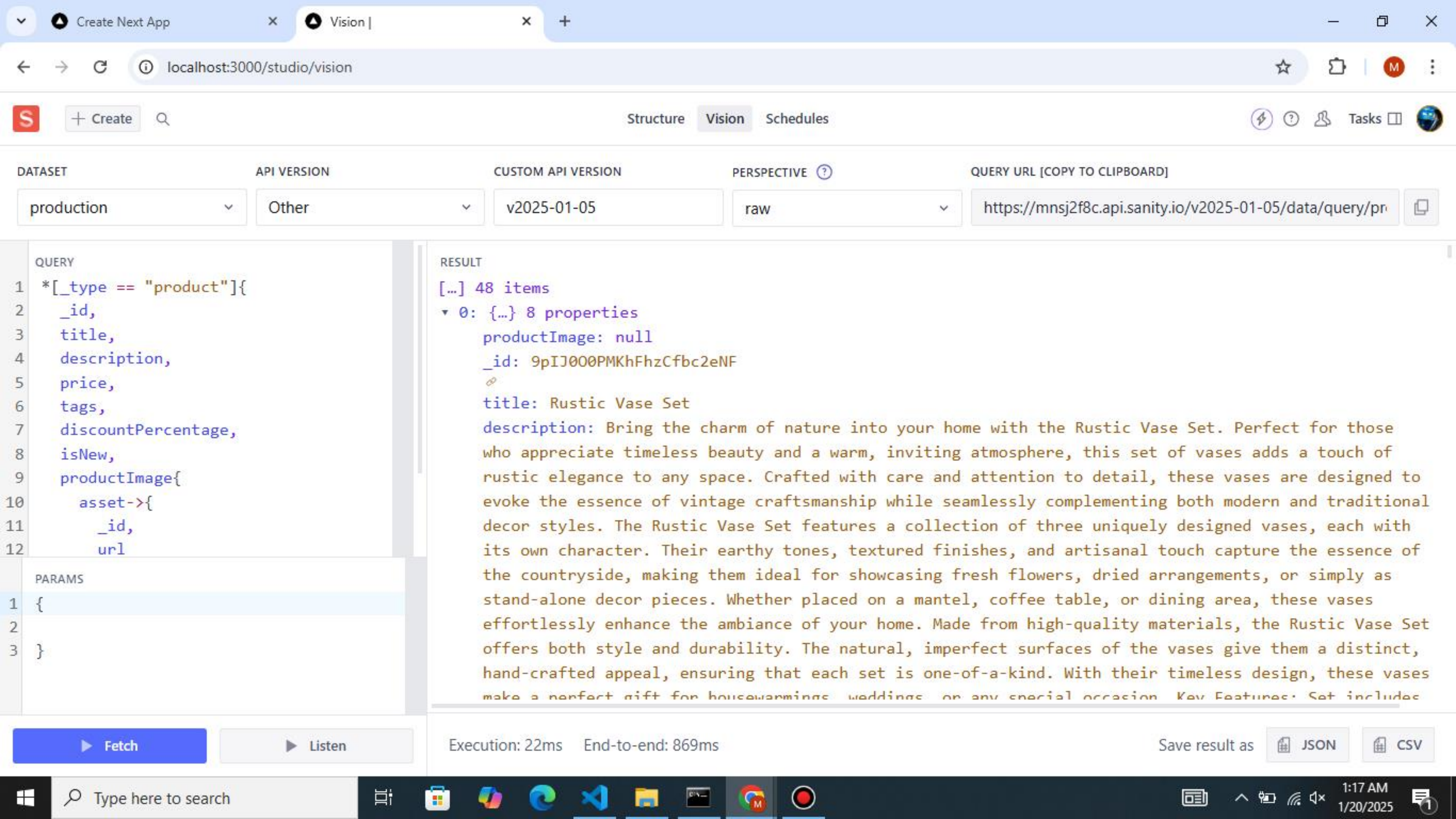
Create product documents in Sanity CMS with the corresponding data.

API Data Was Successfully imported into sanity



The pieces in the ReflectiveHaven collection are designed with clean lines, soothing tones, and a harmonious blend of textures that foster a sense of balance and serenity. From minimalist furniture to subtle decorative accents, each item is thoughtfully crafted to create an atmosphere that encourages mindfulness and

300



localhost:3000/studio/vision

DATASET: production
API VERSION: Other
CUSTOM API VERSION: v2025-01-05
PERSPECTIVE: raw
QUERY URL [COPY TO CLIPBOARD]: https://mnsj2f8c.api.sanity.io/v2025-01-05/data/query/pr

QUERY
```
*[_type == "product"]{
  _id,
  title,
  description,
  price,
  tags,
  discountPercentage,
  isNew,
  productImage{
    asset->{
      _id,
      url
```

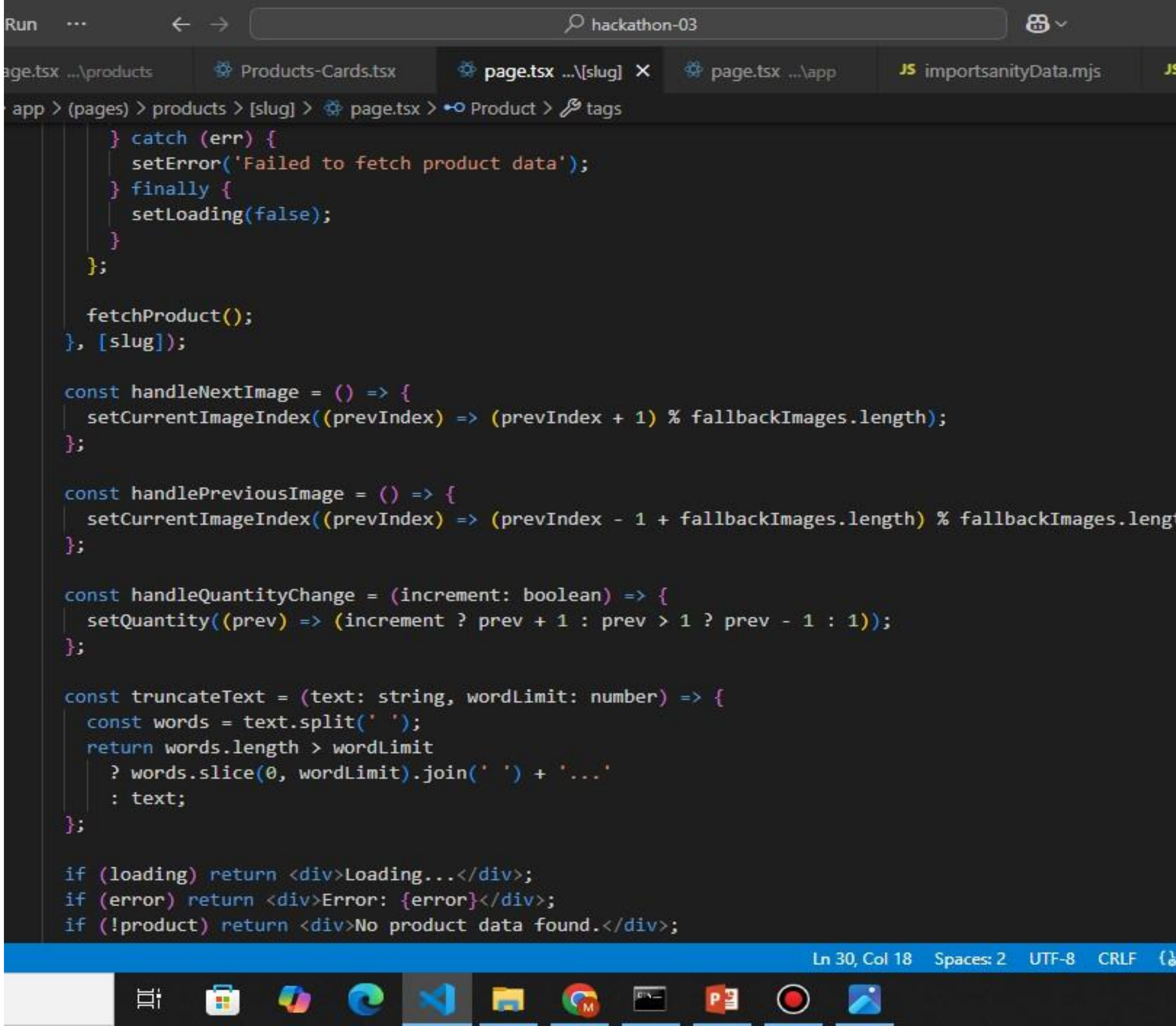
PARAMS
```
{

}
```

RESULT
```
[…] 48 items
▾ 0: {…} 8 properties
    productImage: null
    _id: 9pIJ0O0PMKhFhzCfbc2eNF
    title: Rustic Vase Set
    description: Bring the charm of nature into your home with the Rustic Vase Set. Perfect for those
    who appreciate timeless beauty and a warm, inviting atmosphere, this set of vases adds a touch of
    rustic elegance to any space. Crafted with care and attention to detail, these vases are designed to
    evoke the essence of vintage craftsmanship while seamlessly complementing both modern and traditional
    decor styles. The Rustic Vase Set features a collection of three uniquely designed vases, each with
    its own character. Their earthy tones, textured finishes, and artisanal touch capture the essence of
    the countryside, making them ideal for showcasing fresh flowers, dried arrangements, or simply as
    stand-alone decor pieces. Whether placed on a mantel, coffee table, or dining area, these vases
    effortlessly enhance the ambiance of your home. Made from high-quality materials, the Rustic Vase Set
    offers both style and durability. The natural, imperfect surfaces of the vases give them a distinct,
    hand-crafted appeal, ensuring that each set is one-of-a-kind. With their timeless design, these vases
```

Execution: 22ms   End-to-end: 869ms   Save result as JSON CSV

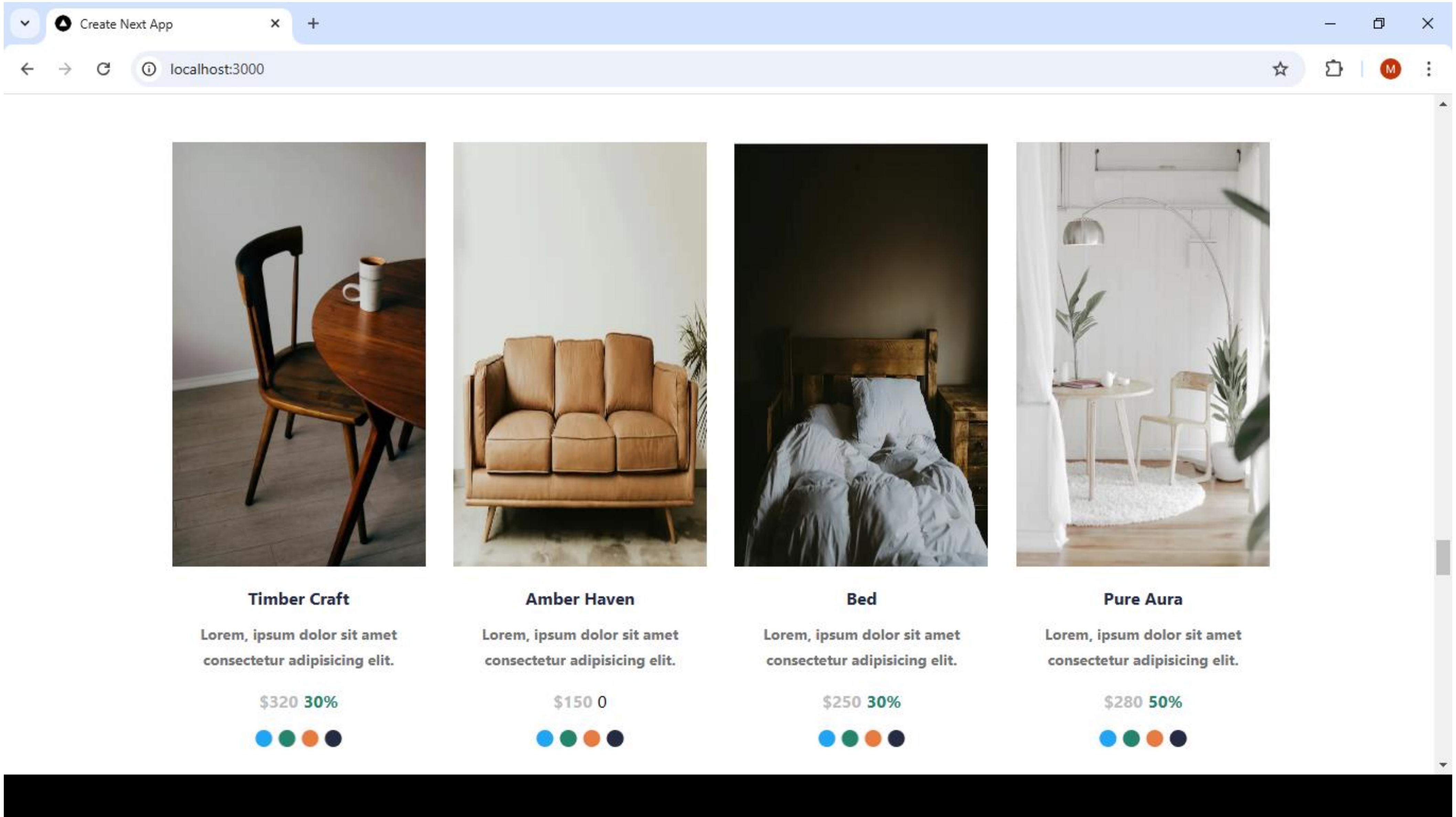
- **API Integration in Next.js**
- Utility Function: A utility function was implemented to fetch data from Sanity CMS.

- Error Handling:
- Fallback images were used for missing imageUrl values.
- Included user friendly error messages.



```
      } catch (err) {
        setError('Failed to fetch product data');
      } finally {
        setLoading(false);
      }
    };

    fetchProduct();
  }, [slug]);

  const handleNextImage = () => {
    setCurrentImageIndex((prevIndex) => (prevIndex + 1) % fallbackImages.length);
  };

  const handlePreviousImage = () => {
    setCurrentImageIndex((prevIndex) => (prevIndex - 1 + fallbackImages.length) % fallbackImages.leng
  };

  const handleQuantityChange = (increment: boolean) => {
    setQuantity((prev) => (increment ? prev + 1 : prev > 1 ? prev - 1 : 1));
  };

  const truncateText = (text: string, wordLimit: number) => {
    const words = text.split(' ');
    return words.length > wordLimit
      ? words.slice(0, wordLimit).join(' ') + '...'
      : text;
  };

  if (loading) return <div>Loading...</div>;
  if (error) return <div>Error: {error}</div>;
  if (!product) return <div>No product data found.</div>;
```

Frontend Rendering



Create Next App

localhost:3000

Timber Craft

Lorem, ipsum dolor sit amet consectetur adipisicing elit.

~~$320~~ 30%

Amber Haven

Lorem, ipsum dolor sit amet consectetur adipisicing elit.

$150 0

Bed

Lorem, ipsum dolor sit amet consectetur adipisicing elit.

$250 30%

Pure Aura

Lorem, ipsum dolor sit amet consectetur adipisicing elit.

$280 50%

MAHNOOR GHAFFAR

mahnoorghaffar9@gmail.com

WEB DEVELOPER