Day 3 - API Integration and Data Migration

Nike store

Date: [17-1-25]

Name: [Aiman Rehman]

Project: [API Integration and Data Migration]

Overview:

The primary focus of Day 3 was on integrating APIs, specifically with **Sanity CMS**, using **GROQ queries**, and updating the schema to support dynamic data requirements. Here's a summary of the tasks accomplished:

Key Objectives:

1. Establish a connection with Sanity CMS for managing product data efficiently.
2. Implement **GROQ queries** to fetch structured data dynamically from Sanity.
3. Update the **Sanity schema** to align with the evolving requirements of the api.
4. Migrate product data from an external API to Sanity CMS, including image uploads.
5. Test and validate the integration to ensure smooth data flow between the backend and frontend.

---

1. API Data Migration to Sanity:

- **Objective:** ○ **Migrate product data, including images, from an external API to Sanity CMS.**
- **Steps Taken:**

1. Fetch Product Data:

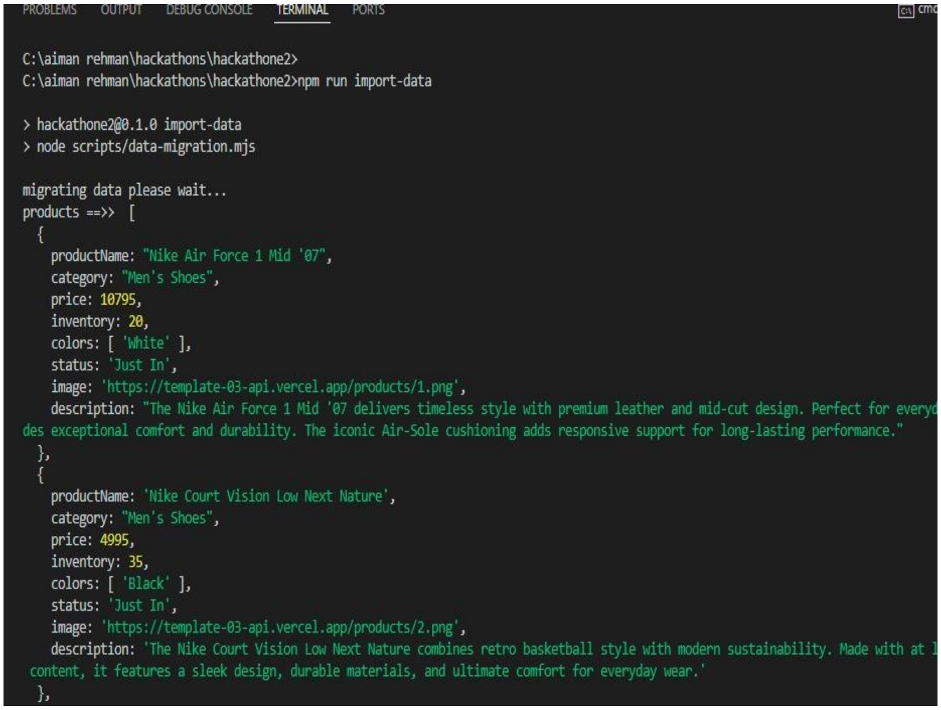
- **Used Axios to retrieve product data from an external API.**
- **Example API Endpoint: https://template-03api.vercel.app/api/products.**

2. Upload Images to Sanity:

- **Uploaded product images using the client.assets.upload method.**
- **Ensured that each image was linked to its respective product.**

3. Create Product Documents in Sanity:

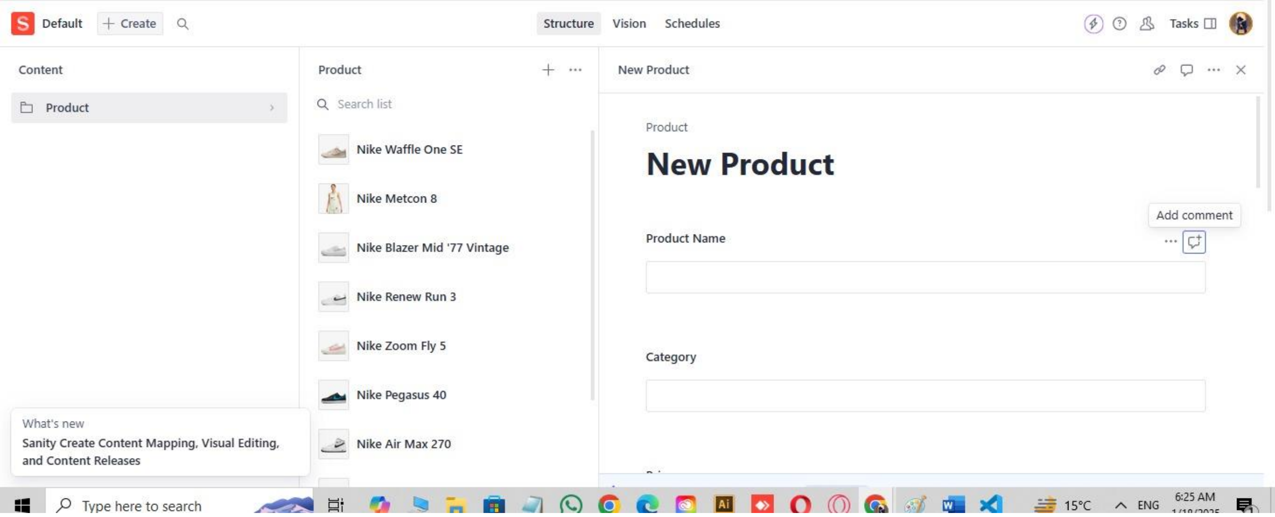
- **Iterated over the fetched data and created a corresponding product document in Sanity using the client.create() method.**

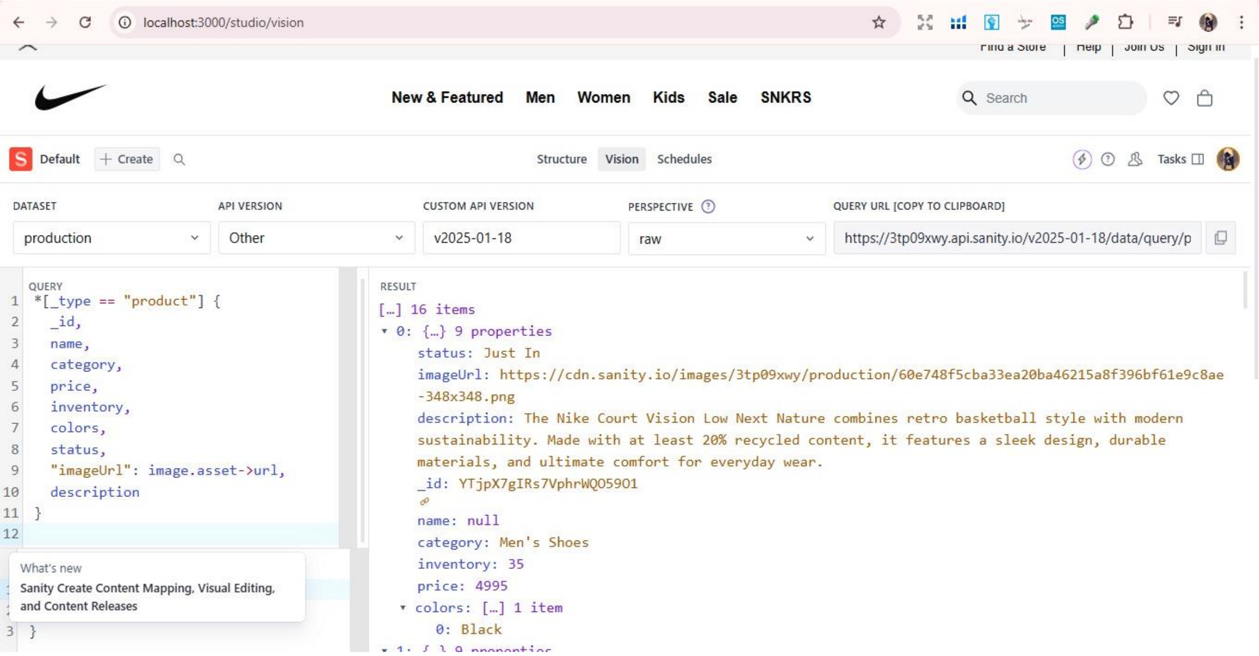


```
C:\aiman rehman\hackathons\hackathone2>
C:\aiman rehman\hackathons\hackathone2>npm run import-data

> hackathone2@0.1.0 import-data
> node scripts/data-migration.mjs

migrating data please wait...
products ==>>  [
  {
    productName: "Nike Air Force 1 Mid '07",
    category: "Men's Shoes",
    price: 10795,
    inventory: 20,
    colors: [ 'White' ],
    status: 'Just In',
    image: 'https://template-03-api.vercel.app/products/1.png',
    description: "The Nike Air Force 1 Mid '07 delivers timeless style with premium leather and mid-cut design. Perfect for everyd
des exceptional comfort and durability. The iconic Air-Sole cushioning adds responsive support for long-lasting performance."
  },
  {
    productName: 'Nike Court Vision Low Next Nature',
    category: "Men's Shoes",
    price: 4995,
    inventory: 35,
    colors: [ 'Black' ],
    status: 'Just In',
    image: 'https://template-03-api.vercel.app/products/2.png',
    description: 'The Nike Court Vision Low Next Nature combines retro basketball style with modern sustainability. Made with at l
 content, it features a sleek design, durable materials, and ultimate comfort for everyday wear.'
  },
```

2. Sanity CMS Integration:

- **Connection Setup:**
  - Integrated the @sanity/client library into the project to enable communication with the Sanity CMS.
  - Configured the client using project-specific credentials, including the projectId, dataset, and apiVersion.
- **Purpose:** To centralize product management and leverage Sanity's features, such as realtime updates, robust content modeling, and optimized image handling




localhost:3000/studio/vision
Find a Store | Help | Join Us | Sign In
New & Featured
Men
Women
Kids
Sale
SNKRS
Search
Default
+ Create
Structure
Vision
Schedules
Tasks
What's new
Sanity Create Content Mapping, Visual Editing, and Content Releases

DATASET: production

API VERSION: Other

CUSTOM API VERSION: v2025-01-18

PERSPECTIVE: raw

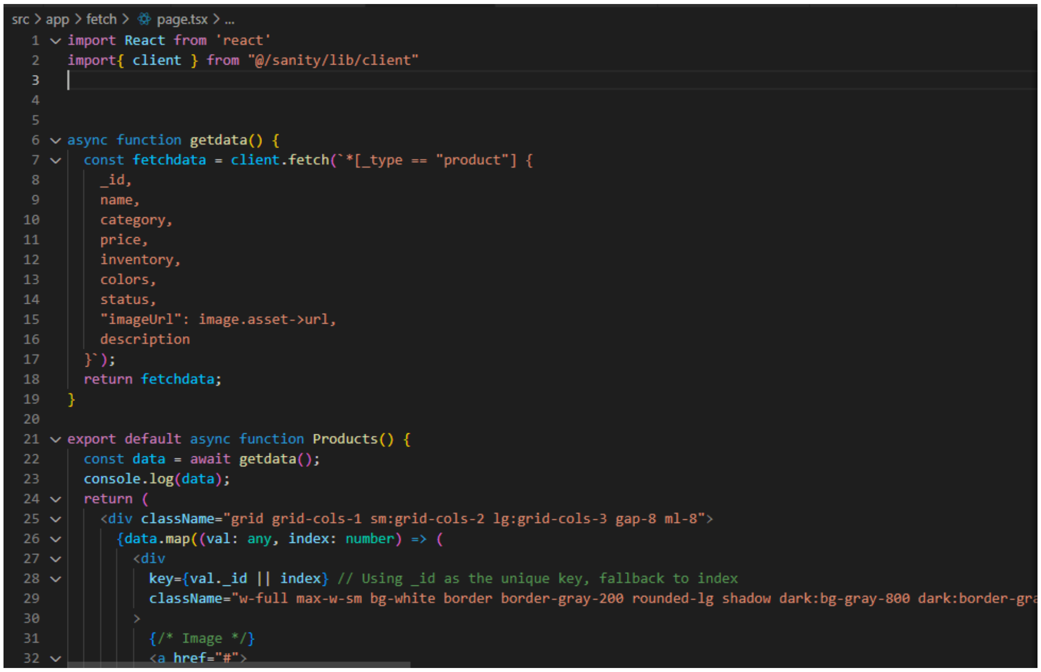
QUERY URL [COPY TO CLIPBOARD]: https://3tp09xwy.api.sanity.io/v2025-01-18/data/query/p

QUERY

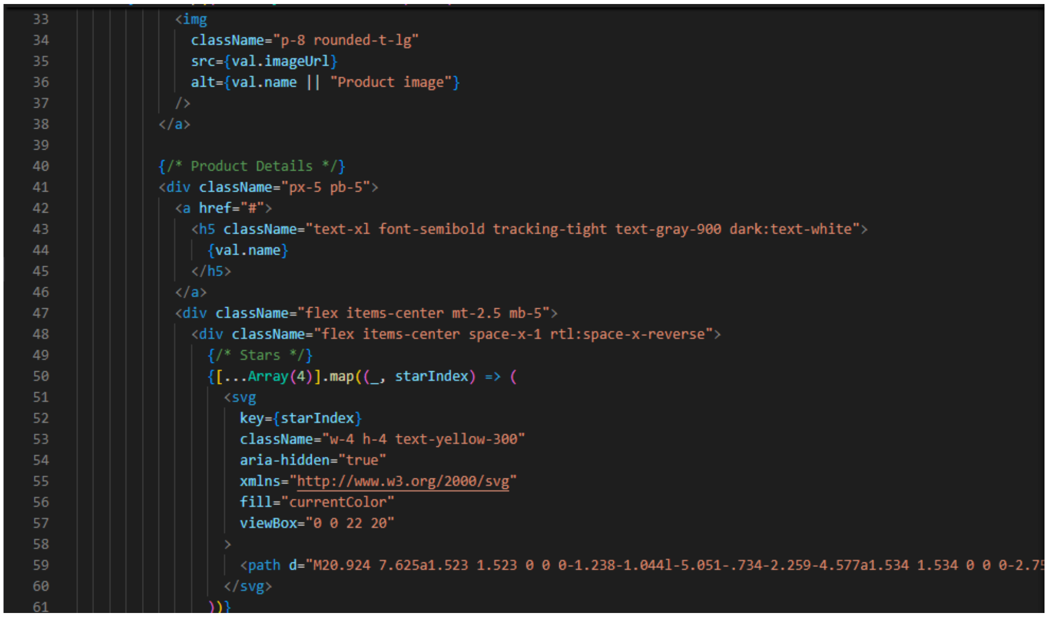
```
*[_type == "product"] {
  _id,
  name,
  category,
  price,
  inventory,
  colors,
  status,
  "imageUrl": image.asset->url,
  description
}
```

RESULT

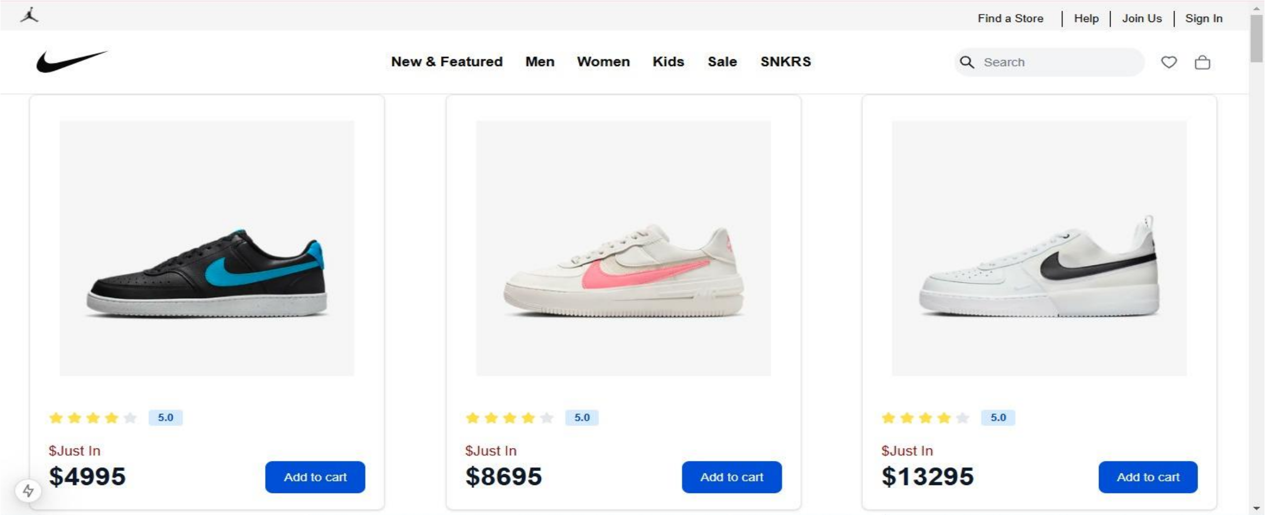
```
[…] 16 items
▾ 0: {…} 9 properties
    status: Just In
    imageUrl: https://cdn.sanity.io/images/3tp09xwy/production/60e748f5cba33ea20ba46215a8f396bf61e9c8ae
    -348x348.png
    description: The Nike Court Vision Low Next Nature combines retro basketball style with modern
    sustainability. Made with at least 20% recycled content, it features a sleek design, durable
    materials, and ultimate comfort for everyday wear.
    _id: YTjpX7gIRs7VphrWQO59O1
    name: null
    category: Men's Shoes
    inventory: 35
    price: 4995
  ▾ colors: […] 1 item
      0: Black
```



```
import React from 'react'
import{ client } from "@/sanity/lib/client"



async function getdata() {
  const fetchdata = client.fetch(`*[_type == "product"] {
    _id,
    name,
    category,
    price,
    inventory,
    colors,
    status,
    "imageUrl": image.asset->url,
    description
  }`);
  return fetchdata;
}

export default async function Products() {
  const data = await getdata();
  console.log(data);
  return (
    <div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-8 ml-8">
      {data.map((val: any, index: number) => (
        <div
          key={val._id || index} // Using _id as the unique key, fallback to index
          className="w-full max-w-sm bg-white border border-gray-200 rounded-lg shadow dark:bg-gray-800 dark:border-gra
        >
          {/* Image */}
          <a href="#">
```



```
            <img
              className="p-8 rounded-t-lg"
              src={val.imageUrl}
              alt={val.name || "Product image"}
            />
          </a>

          {/* Product Details */}
          <div className="px-5 pb-5">
            <a href="#">
              <h5 className="text-xl font-semibold tracking-tight text-gray-900 dark:text-white">
                {val.name}
              </h5>
            </a>
            <div className="flex items-center mt-2.5 mb-5">
              <div className="flex items-center space-x-1 rtl:space-x-reverse">
                {/* Stars */}
                {[...Array(4)].map((_, starIndex) => (
                  <svg
                    key={starIndex}
                    className="w-4 h-4 text-yellow-300"
                    aria-hidden="true"
                    xmlns="http://www.w3.org/2000/svg"
                    fill="currentColor"
                    viewBox="0 0 22 20"
                  >
                    <path d="M20.924 7.625a1.523 1.523 0 0 0-1.238-1.044l-5.051-.734-2.259-4.577a1.534 1.534 0 0 0-2.7
                  </svg>
                ))}
```



5.0

$Just In

$4995

5.0

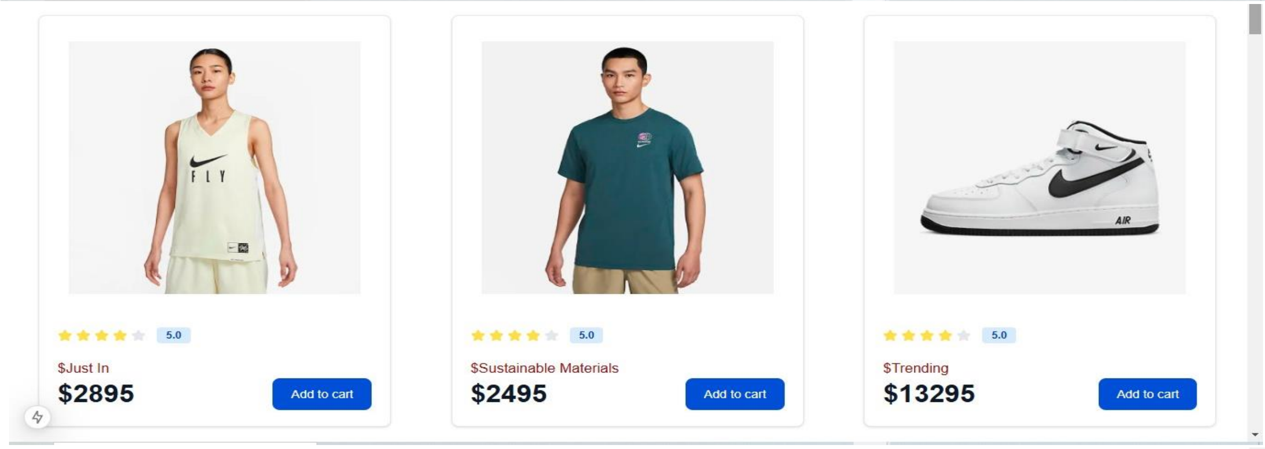
$Just In

$8695

5.0

$Just In

$13295


5.0

$Just In

$2895

5.0

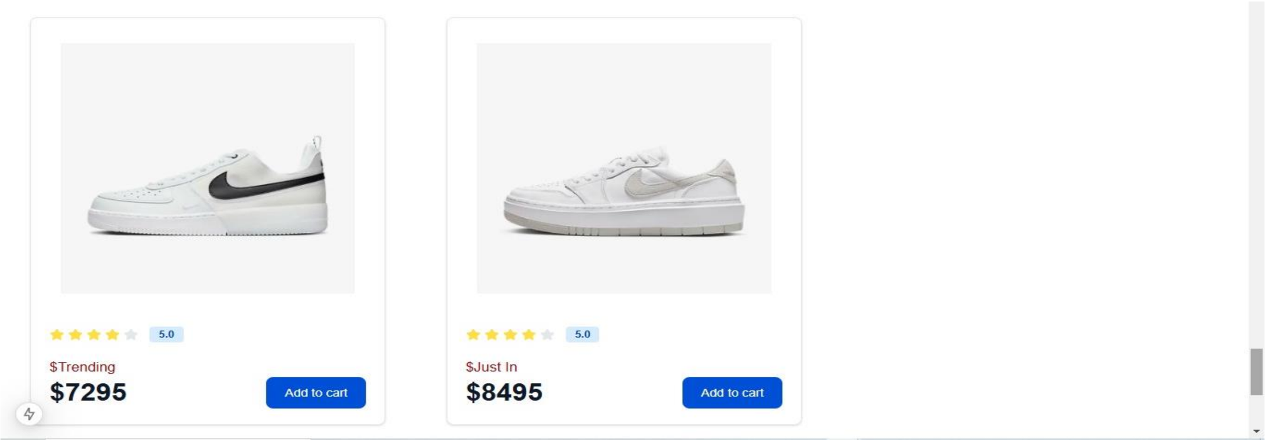
$Sustainable Materials

$2495

5.0

$Trending

$13295


5.0

$Trending

$7295

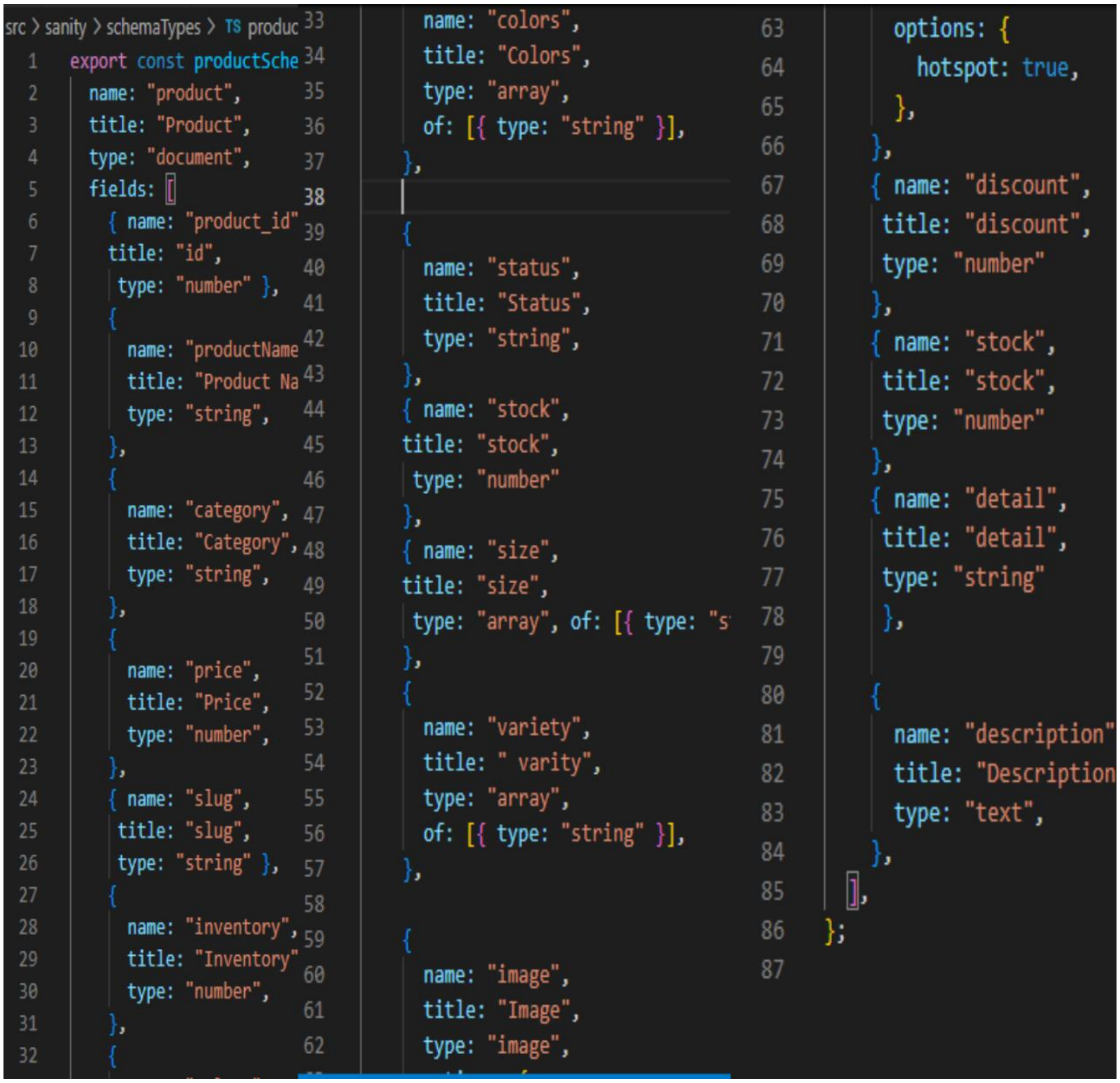
5.0

$Just In

$8495

3. Schema Enhancements in Sanity CMS:

- Updated the product schema to align with the requirements of the application, ensuring it could handle new fields and validations.
- **Fields Added/Updated:**
  - **inventory**: Tracks stock levels for products.
  - **colors**: Stores available color options as an array.



```
src > sanity > schemaTypes > TS produc
export const productSche
  name: "product",
  title: "Product",
  type: "document",
  fields: [
    { name: "product_id"
    title: "id",
     type: "number" },
    {
      name: "productName
      title: "Product Na
      type: "string",
    },
    {
      name: "category",
      title: "Category",
      type: "string",
    },
    {
      name: "price",
      title: "Price",
      type: "number",
    },
    { name: "slug",
     title: "slug",
     type: "string" },
    {
      name: "inventory",
      title: "Inventory"
      type: "number",
    },
    {
      name: "colors",
      title: "Colors",
      type: "array",
      of: [{ type: "string" }],
    },

    {
      name: "status",
      title: "Status",
      type: "string",
    },
    { name: "stock",
    title: "stock",
     type: "number"
    },
    { name: "size",
    title: "size",
     type: "array", of: [{ type: "s
    },
    {
      name: "variety",
      title: " varity",
      type: "array",
      of: [{ type: "string" }],
    },

    {
      name: "image",
      title: "Image",
      type: "image",
      options: {
        hotspot: true,
      },
    },
    { name: "discount",
    title: "discount",
    type: "number"
    },
    { name: "stock",
    title: "stock",
    type: "number"
    },
    { name: "detail",
    title: "detail",
    type: "string"
     },

    {
      name: "description"
      title: "Description
      type: "text",
    },
  ],
};
```

Self-Validation Checklist:

API Understanding:	Schema Validation:	Data Migration:	API Integration in Next.js	Submission Preparation:
✓	✓	✓	✓	✓

Outcomes of Day 3:

- Successfully integrated **Sanity CMS** with the application.
- Implemented dynamic **GROQ queries** to fetch and display product data.
- Enhanced the Sanity schema to support extended product attributes.
- Migrated data and images from an external API to Sanity CMS.
- Validated and tested the integration to ensure seamless backend-to-frontend data flow.

This concludes my Day 3 Hackathon work by **Aiman Rehman**. I will continue to work diligently to turn these features into reality for my e-commerce website