

Day3: API Integration and Data Migration for Nike Products Marketplace

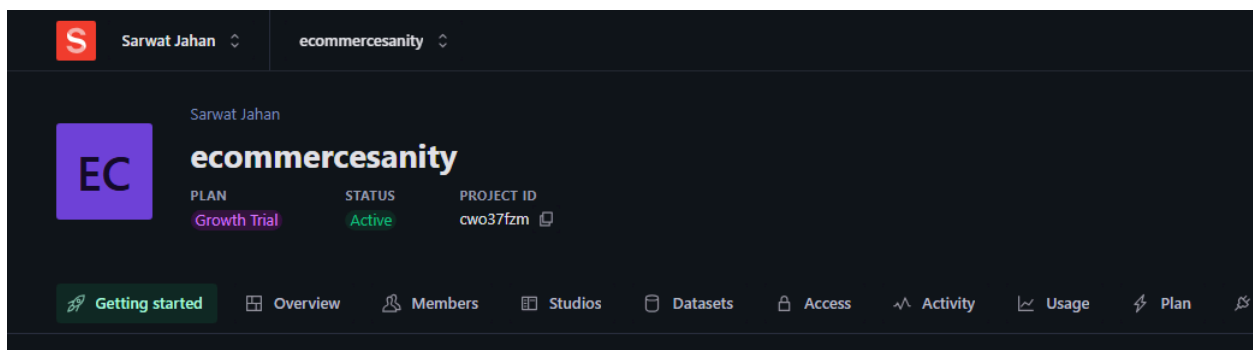
This document details the activities undertaken on the third day of the Nike Products Marketplace hackathon.

It includes custom migration, integrating data from Sanity, creating schemas, and presenting data using GROQ queries within a Next.js application.

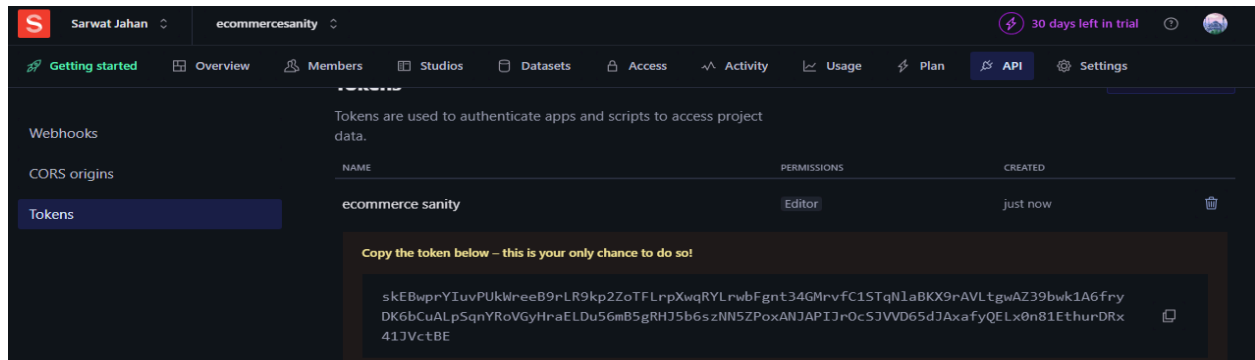
Each Segment is accompanied by annotated code images with comprehensive explanations of their functions.

API Integration process:

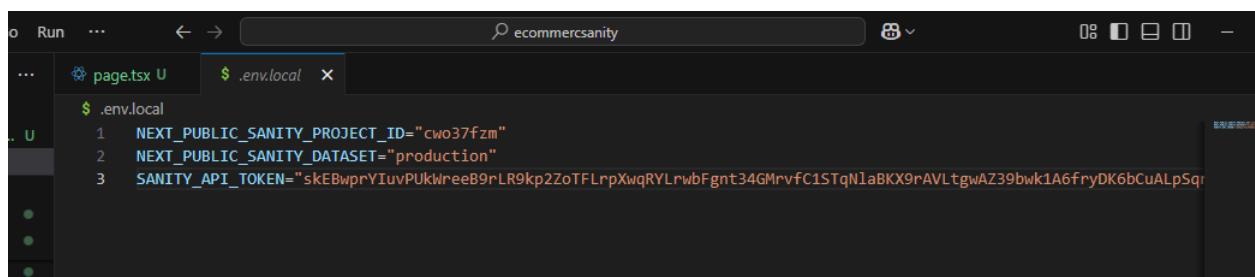
- 1.First, created a project in my Sanity account ,named 'ecommercesanity'.
- 2.Installed Sanity in my Next.js Project folder.



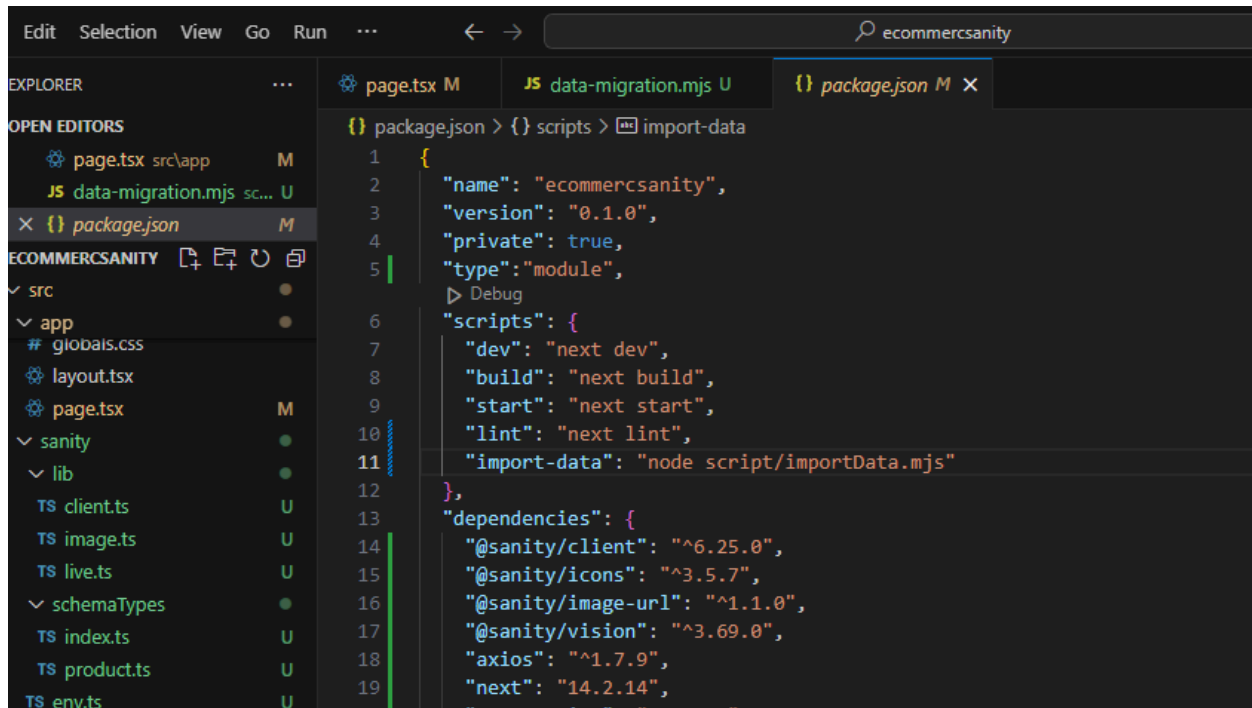
- 3.Then , I created a Token.



4.Token created in sanity, then pasted in my project.



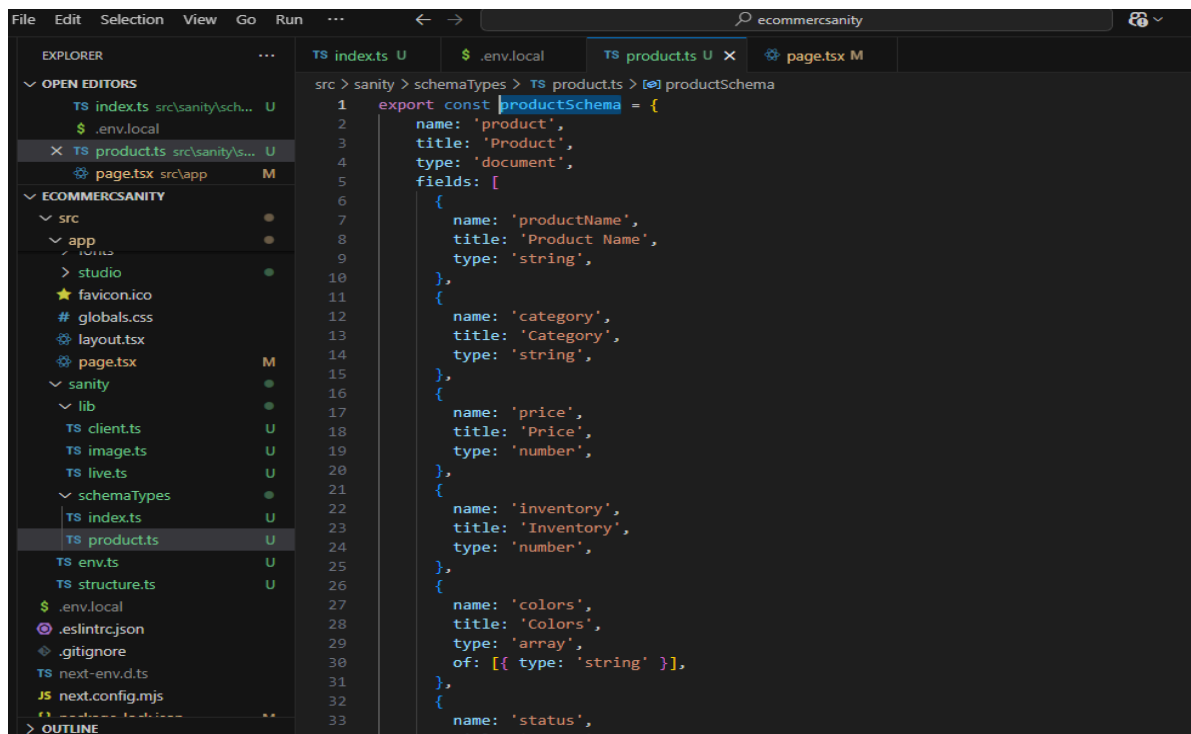
5. Import-Data is added in the package.json file.



The screenshot shows the VS Code interface with the Explorer on the left and the Editor on the right. The Explorer shows the project structure with folders like 'src', 'app', 'sanity', and 'lib'. The Editor displays the 'package.json' file with the following content:

```
1 {
2   "name": "ecommercsanity",
3   "version": "0.1.0",
4   "private": true,
5   "type": "module",
6   "scripts": {
7     "dev": "next dev",
8     "build": "next build",
9     "start": "next start",
10    "lint": "next lint",
11    "import-data": "node script/importData.mjs"
12  },
13  "dependencies": {
14    "@sanity/client": "^6.25.0",
15    "@sanity/icons": "^3.5.7",
16    "@sanity/image-url": "^1.1.0",
17    "@sanity/vision": "^3.69.0",
18    "axios": "^1.7.9",
19    "next": "14.2.14",
20    "react": "18.3.1",
21    "react-dom": "18.3.1"
22  }
23 }
```

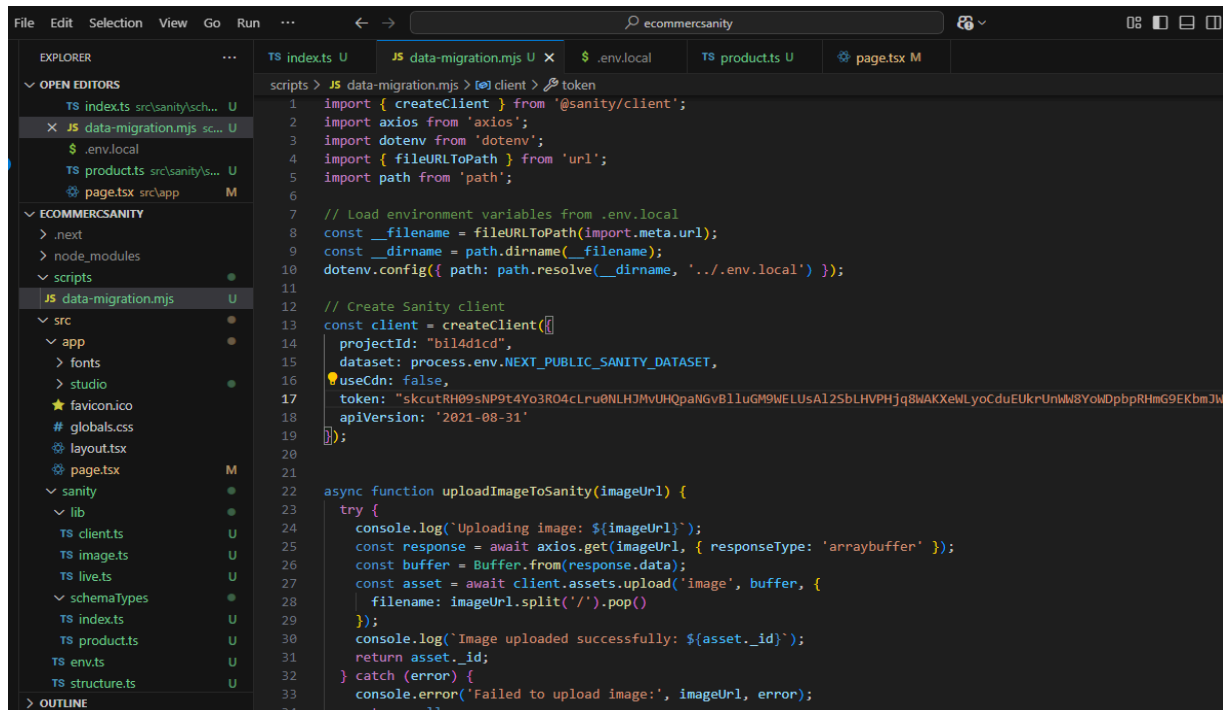
6. In SchemaTypes, a file named product.ts is created which contains Product Schema ,provided by the teacher.



The screenshot shows the VS Code interface with the Explorer on the left and the Editor on the right. The Explorer shows the project structure with folders like 'src', 'app', 'sanity', and 'lib'. The Editor displays the 'productSchema.ts' file with the following content:

```
1 export const productSchema = {
2   name: 'product',
3   title: 'Product',
4   type: 'document',
5   fields: [
6     {
7       name: 'productName',
8       title: 'Product Name',
9       type: 'string',
10    },
11    {
12      name: 'category',
13      title: 'Category',
14      type: 'string',
15    },
16    {
17      name: 'price',
18      title: 'Price',
19      type: 'number',
20    },
21    {
22      name: 'inventory',
23      title: 'Inventory',
24      type: 'number',
25    },
26    {
27      name: 'colors',
28      title: 'Colors',
29      type: 'array',
30      of: [{ type: 'string' }],
31    },
32    {
33      name: 'status',
34      title: 'Status',
35      type: 'string',
36    }
37  ]
38 }
```

7. Created a data-migration.mjs file in scripts folder and add function to integrate API in sanity project.



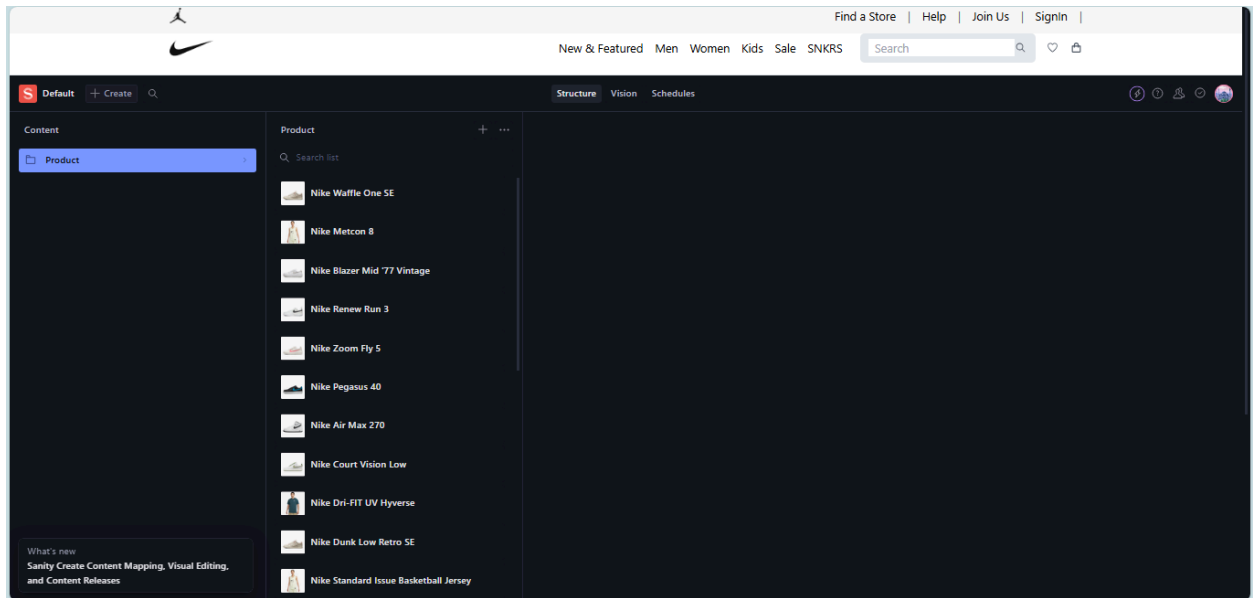
```
1 import { createClient } from '@sanity/client';
2 import axios from 'axios';
3 import dotenv from 'dotenv';
4 import { fileURLToPath } from 'url';
5 import path from 'path';
6
7 // Load environment variables from .env.local
8 const __filename = fileURLToPath(import.meta.url);
9 const __dirname = path.dirname(__filename);
10 dotenv.config({ path: path.resolve(__dirname, '../.env.local') });
11
12 // Create Sanity client
13 const client = createClient({
14   projectId: "bil4d1cd",
15   dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
16   useCdn: false,
17   token: "skcutRH09sNP9t4Yo3R04cLru0NLHJmVUHQpaNgvB1luGM9WELUsA125bLHVPHjq8WAKXewLYoCduEUkrUnWMSYomDpbpRHmG9EKbmJWM",
18   apiVersion: '2021-08-31'
19 });
20
21 async function uploadImageToSanity(imageUrl) {
22   try {
23     console.log(`Uploading image: ${imageUrl}`);
24     const response = await axios.get(imageUrl, { responseType: 'arraybuffer' });
25     const buffer = Buffer.from(response.data);
26     const asset = await client.assets.upload('image', buffer, {
27       filename: imageUrl.split('/').pop()
28     });
29     console.log(`Image uploaded successfully: ${asset._id}`);
30     return asset._id;
31   } catch (error) {
32     console.error('Failed to upload image:', imageUrl, error);
33     return null;
34   }
35 }
```

8. After all the important steps for migrating API data in my projects, this command ran in terminal
npm run import -data.

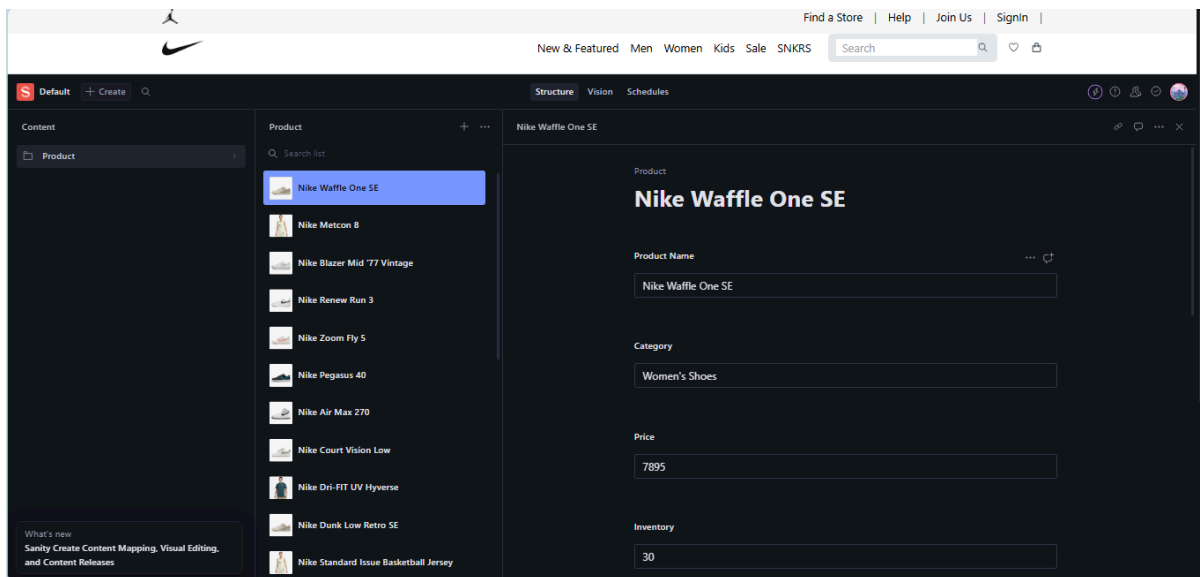


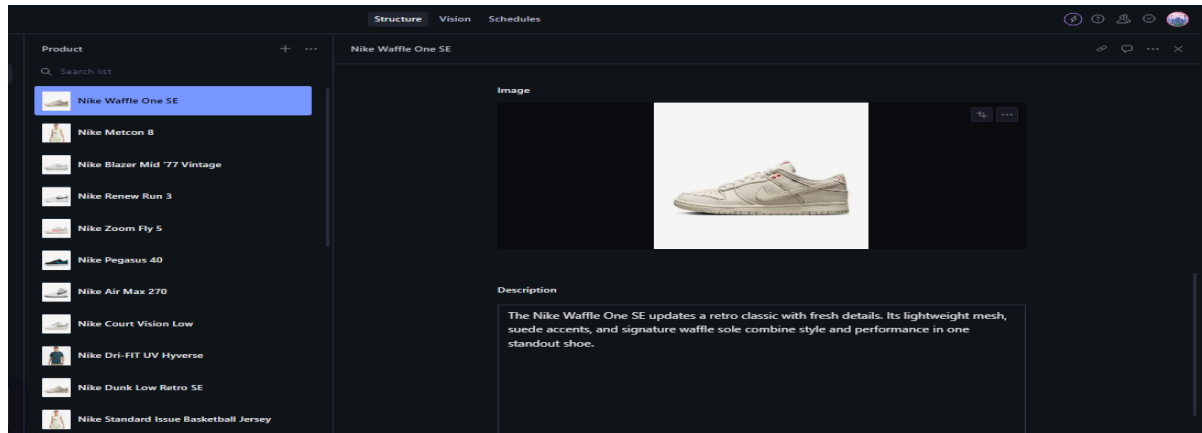
```
> node scripts/importSanityData.mjs
migrating data please wait...
products ==>> [
  {
    productName: "Nike Air Force 1 Mid '07",
    category: "Men's Shoes",
    price: 10795,
    inventory: 20,
    colors: [ 'white' ],
    status: 'Just In',
    image: 'https://template-03-api.vercel.app/products/1.png',
    description: "The Nike Air Force 1 Mid '07 delivers timeless style with premium leather and mid-cut design. Perfect for everyday wear, it provides exceptional comfort and durability. The iconic Air-Sole cushioning adds responsive support for long-lasting performance."
  }
],
```

9. All the Data through API was integrated in Sanity Studio.



Each Product can see in Studio separately.





Self-Validation Checklist:

Tasks:

API Understanding

Schema Validation

Data Migration

API Integration in Next.js

Submission Preparation:

- Data Fetched and Displayed successfully in Sanity CMS.
- With manually entering data, frontend validation.
- Screenshots are added in file to show my work.

Conclusion:

This report documents the manual data migration process and its integration with Sanity CMS for Nike products, ensuring an optimal user experience for local dropshipping.