

MUHAMMAD UMER

304953

(Student Leader Friday 9-12)

Day 3 - API Integration Report – General Ecommerce Nike Website

1. API Integration Process

- Used **Mock API Tool** to create a mock API for testing.
- Designed the API structure, ensuring it aligned with the required data model.
- Set field types in the schema based on API response.
- Used **Thunder Client** to test the API's functionality and validate responses.

2. Schema Adjustments

- Defined a schema in **Sanity** that matched the mock API structure.
- Ensured proper data types for each field to avoid inconsistencies.
- Adjusted schema fields based on API responses to maintain data integrity.

3. Migration Steps & Tools Used

- Created a **migration file** to import API data into **Sanity**.
- Executed the migration script to transfer API data to the Sanity database.
- Used **Sanity Vision** to write **GROQ queries** for data retrieval.
- Developed a **utility function** to fetch data from Sanity to the frontend.
- Rendered the fetched **products** dynamically on the frontend.

Following are Some Screenshots of Day 3 Task

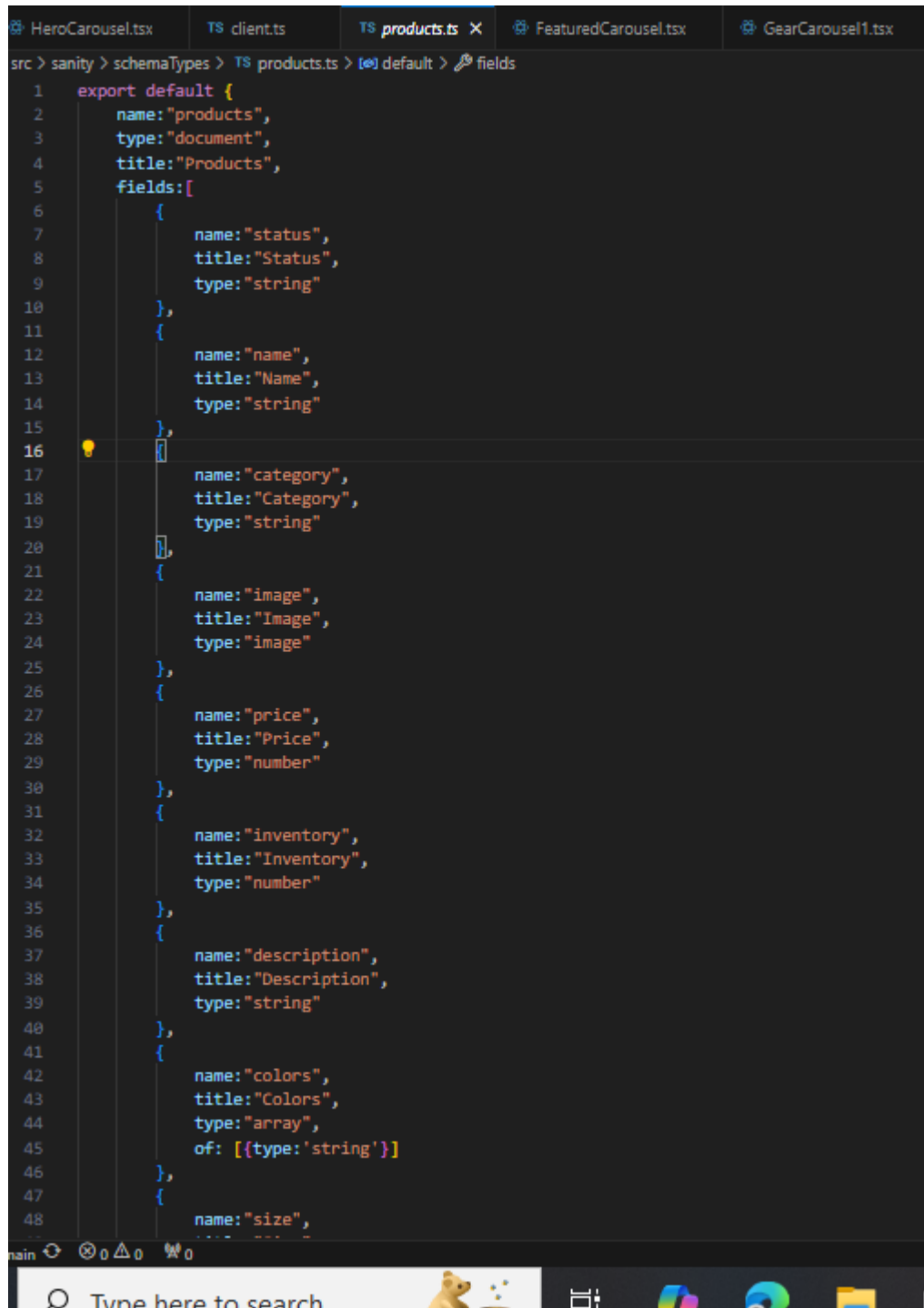
1) Create Mock API

```
[
  {
    "name": "Nike Air Max 90",
    "image": "https://static.nike.com/a/images/c_limit,w_592,f_auto/t_product_v1/8c673990-2e05-447c-b20a-b5ed57fbb007/NIKE+AIR+MAX+SC.png",
    "description": "Classic design with premium cushioning.",
    "price": 12045,
    "size": [
      "UK 7",
      "UK 8",
      "UK 9",
      "UK 10",
      "UK 11"
    ],
    "colors": [
      "Black",
      "White",
      "Red"
    ],
    "category": "Men's Shoes",
    "inventory": 32,
    "status": "Just In",
    "id": "1"
  },
  {
    "name": "Nike Air Jordan 1",
    "image": "https://static.nike.com/a/images/c_limit,w_592,f_auto/t_product_v1/d1641887-f508-4442-89ab-c1bfa5033cbf/NIKE+P-6000.png",
    "description": "Iconic high-top basketball sneaker.",
    "price": 15370,
    "size": [
      "UK 8",
      "UK 9",
      "UK 10",
      "UK 11",

```

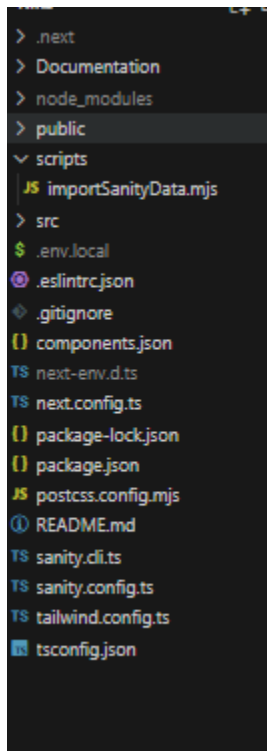
Activate V
Go to Setting

2)Create Sanity Schema



```
1  export default {
2    name: "products",
3    type: "document",
4    title: "Products",
5    fields: [
6      {
7        name: "status",
8        title: "Status",
9        type: "string"
10     },
11     {
12       name: "name",
13       title: "Name",
14       type: "string"
15     },
16     {
17       name: "category",
18       title: "Category",
19       type: "string"
20     },
21     {
22       name: "image",
23       title: "Image",
24       type: "image"
25     },
26     {
27       name: "price",
28       title: "Price",
29       type: "number"
30     },
31     {
32       name: "inventory",
33       title: "Inventory",
34       type: "number"
35     },
36     {
37       name: "description",
38       title: "Description",
39       type: "string"
40     },
41     {
42       name: "colors",
43       title: "Colors",
44       type: "array",
45       of: [{ type: 'string' }]
46     },
47     {
48       name: "size",
49       type: "string"
50     }
51   ]
52 }
```

3) Create Scripts folder in root directory and then create migration file inside scripts folder



4)Set Script in package.json

```
() package.json > {} dependencies
1  {
2    "name": "nike",
3    "version": "0.1.0",
4    "private": true,
   ▶ Debug
5    "scripts": {
6      "dev": "next dev --turbo",
7      "build": "next build",
8      "start": "next start",
9      "lint": "next lint",
10     "import-data": "node scripts/importSanityData.mjs"
11   },
12   "dependencies": {
13     "@radix-ui/react-checkbox": "^1.1.2",
```

5) Migration file help to import API data into Sanity

```
scripts > js importSanityData.mjs > importData > sanityProduct > _type
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    token: process.env.NEXT_PUBLIC_SANITY_API_TOKEN,
15    projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
16    dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
17    apiVersion: process.env.NEXT_PUBLIC_SANITY_Api_Version,
18    useCdn: false, // Use 'false' to avoid cached responses during uploads
19  });
20
21  // Function to upload an image to Sanity
22  async function uploadImageToSanity(imageUrl) {
23    try {
24      console.log('Uploading image: ${imageUrl}');
25
26      // Validate if imageUrl is not null or empty
27      if (!imageUrl) {
28        console.error('Image URL is missing!');
29        return null;
30      }
31
32      // Fetch the image from the provided URL
33      const response = await axios.get(imageUrl, { responseType: 'arraybuffer' });
34      console.log('Image fetched successfully:', response.status);
35
36      // Convert the response data to a buffer
37      const buffer = Buffer.from(response.data);
38
39      // Upload the image to Sanity
40      const asset = await client.assets.upload('image', buffer, {
41        filename: imageUrl.split('/').pop(),
42      });
43
44      console.log('Image uploaded successfully: ${asset._id}');
45      return asset._id;
46    } catch (error) {
47      console.error('Failed to upload image:', error.message, error.response?.data || '');
48      return null;
49    }
  }
```

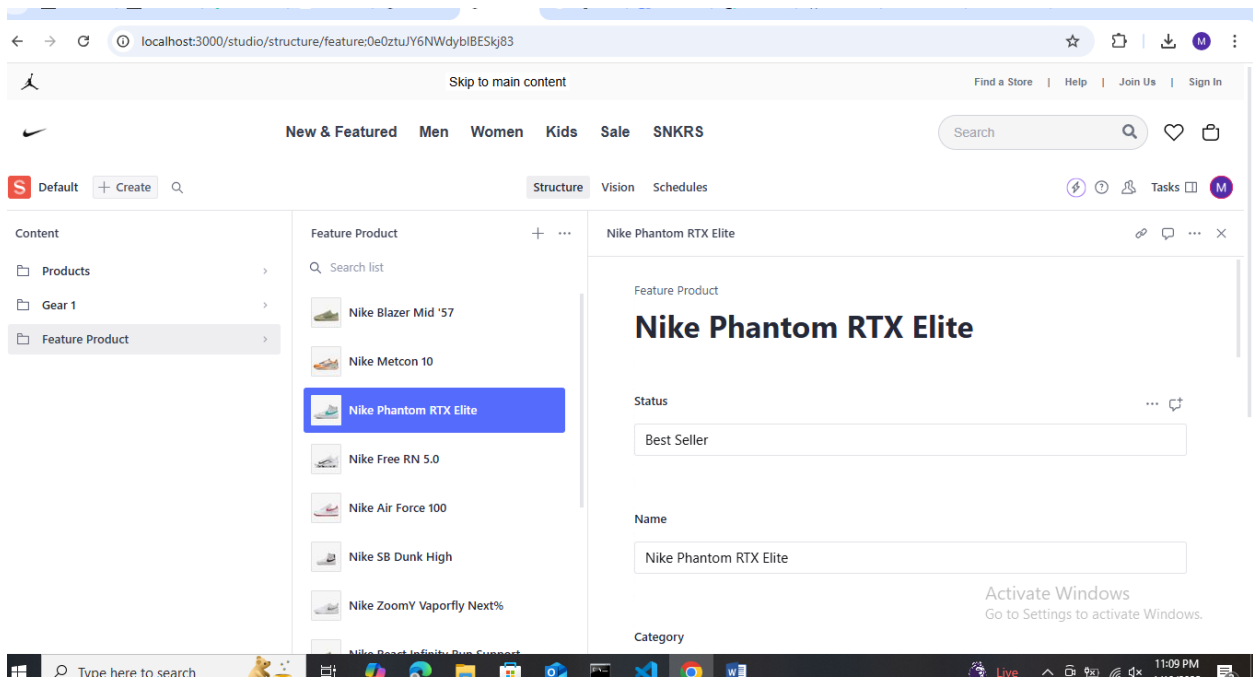
importSanityData.mjs and its dependencies

6) Import Data Successfully

The screenshot displays a VS Code editor with a file explorer on the left and a terminal window on the right. The file explorer shows a project structure with files like `.next`, `Documentation`, `node_modules`, `public`, `scripts`, `importSanityData.mjs`, `src`, `.env.local`, `.eslintrc.json`, `.gitignore`, `components.json`, `next-env.d.ts`, `next.config.ts`, `package-lock.json`, `package.json`, `postcss.config.mjs`, `README.md`, `sanity.d.ts`, `sanity.config.ts`, `tailwind.config.ts`, and `tsconfig.json`. The `package.json` file is selected. The terminal window shows the output of a script, with messages indicating the successful fetching and uploading of various Nike product images to a Sanity database. The status bar at the bottom indicates the current line and column (Ln 19, Col 22) and the file encoding (UTF-8).

```
Image fetched successfully: 200
Image uploaded successfully: image-9d4ef465d44f155c37b4d9cad188852e35c95eb7-592x592-jpg
Uploading product to Sanity: Nike ZoomV Vaporfly Next%
Product uploaded successfully: BQj9k9wMf2GlcUg3GyIt
Processing product: Nike SB Dunk High
Uploading image: https://static.nike.com/a/images/c_limit,w_592,f_auto/t_product_v1/786c669c-d43f-4e18-b29f-e0296bed8f61/NIKE+BLAZER+MID+X2777+WNIG.png
Image fetched successfully: 200
Image uploaded successfully: image-a3b7169a0ce9108faa945ef41919b4636113a3da-592x592-jpg
Uploading product to Sanity: Nike SB Dunk High
Product uploaded successfully: 0e0rtuY6NMyb1BESkibX
Processing product: Nike Air Force 100
Uploading image: https://static.nike.com/a/images/c_limit,w_592,f_auto/t_product_v1/c7989c35-f33a-4485-8864-2bb088612f32/NIKE+AIR+FORCE+1+X2787+NEXT+WNIG.png
Image fetched successfully: 200
Image uploaded successfully: image-6ad4ba3b7befb70127a70454e609efae348ddc62-592x592-jpg
Uploading product to Sanity: Nike Air Force 100
Product uploaded successfully: BQj9k9wMf2GlcUg3Gz59
Processing product: Nike Free RN 5.0
Uploading image: https://static.nike.com/a/images/c_limit,w_592,f_auto/t_product_v1/aeb54c7f-51da-4405-82c9-a4a970a54633/NIKE+ALPHA+MEMACE+4+SHARK.png
Image fetched successfully: 200
Image uploaded successfully: image-07d0c62c0e673e7a3c71c483b7b9f38ac581475c-592x592-jpg
Uploading product to Sanity: Nike Free RN 5.0
Product uploaded successfully: YjFidg1LOD0J2gZ7Pw607
Processing product: Nike Phantom RTX Elite
Uploading image: https://static.nike.com/a/images/c_limit,w_592,f_auto/t_product_v1/cf97d90d-0ef4-4866-958a-05eddb7c692c/BLAZER+MID+X2777+WNIG.png
Image fetched successfully: 200
Image uploaded successfully: image-bc64f06bda4d81682bead472646c41a876e4c2b0-592x592-jpg
Uploading product to Sanity: Nike Phantom RTX Elite
Product uploaded successfully: 0e0rtuY6NMyb1BESkibX
Processing product: Nike Metcon 10
Uploading image: https://static.nike.com/a/images/c_limit,w_592,f_auto/t_product_v1/99c91b89-302d-4080-b478-999849bd311f/AIR+MAX+SOLO+TENNESSEE.png
Image fetched successfully: 200
Image uploaded successfully: image-8df99d78505a6cc7cf634436d376a14fedcac8-592x592-jpg
Uploading product to Sanity: Nike Metcon 10
Product uploaded successfully: BQj9k9wMf2GlcUg3H098
Processing product: Nike Blazer Mid '57
Uploading image: https://static.nike.com/a/images/c_limit,w_592,f_auto/t_product_v1/6fe8c522-81fc-44ec-a878-330deb133984/NIKE+DUNK+LOW.png
Uploading product to Sanity: Nike Metcon 10
Product uploaded successfully: BQj9k9wMf2GlcUg3H098
Processing product: Nike Blazer Mid '57
Uploading image: https://static.nike.com/a/images/c_limit,w_592,f_auto/t_product_v1/6fe8c522-81fc-44ec-a878-330deb133984/NIKE+DUNK+LOW.png
Image fetched successfully: 200
Image uploaded successfully: image-c91b38c1a8b0e59e4d703855831d569f349597f8-592x592-jpg
Uploading product to Sanity: Nike Blazer Mid '57
Product uploaded successfully: BQj9k9wMf2GlcUg3H0d0
Data import completed successfully!
```

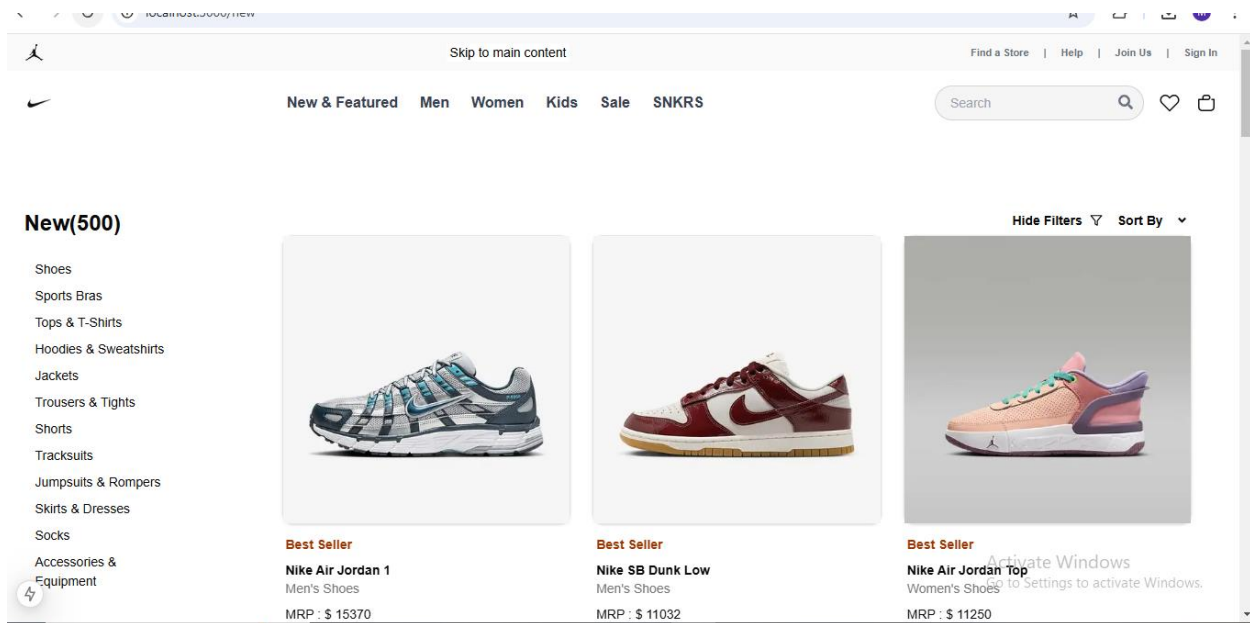
8)Sanity Structure



9)Fetch Data From Sanity to Frontend



10)Data Successfully Rendered on Frontend



TASK :

STATUS

API Uderstanding

YES

Schema Validation

YES

Data Migration

YES

API Integration in NEXT JS

YES

Submission Preparation

YES