

Day 3 - API Integration Report - [Ecommerce Furniture website]

Prepared By: Ameen Alam

1. API Integration Process

Process Overview:

- Integrated the API endpoint <https://template-0-beta.vercel.app/api/product> to fetch product data.
- Utilized axios for making HTTP requests in the data-migration.mjs script.
- Ensured sensitive API keys were stored securely in the .env.local file.

Challenges and Resolutions:

- **Unauthorized Access:** Fixed by verifying the Sanity API token and setting correct permissions.
- **Network Errors:** Implemented retry logic for image uploads to handle transient network issues.
- **Data Validation:** Validated API response and handled empty or malformed data.

2. Schema Adjustments

```
src > sanity > schematypes > JS product.js > ...
2 export default {
3   // ...
4   // ...
5   // ...
6   fields: [
7     {
8       name: 'id',
9       title: 'ID',
10      type: 'string',
11    },
12    {
13      name: 'name',
14      title: 'Name',
15      type: 'string',
16    },
17    {
18      name: 'image',
19      title: 'Image',
20      type: 'image',
21    },
22    {
23      name: 'imagePath',
24      title: 'Image Path',
25      type: 'url',
26    },
27    {
28      name: 'price',
29      title: 'Price',
30      type: 'number',
31    },
32    {
33      name: 'description',
34      title: 'Description',
35      type: 'text',
36    },
37    {
38      name: 'discountPercentage',
39      title: 'Discount Percentage',
40      type: 'number',
41    },
42    {
43      name: 'isFeaturedProduct',
44      title: 'Is Featured Product',
45      type: 'boolean',
46    },
47    {
48      name: 'stockLevel',
49      title: 'Stock Level',
50      type: 'number',
51    },
52    {
53      name: 'category',
54      title: 'Category',
55      type: 'string',
56    },
57  ],
58 }
```

Changes Made:

- Added new fields in the Sanity schema for product:
 - id (String)

- `image` (URL)
 - `discountPercentage` (Number)
 - `isFeaturedProduct` (Boolean)
- Updated field types to align with the API response structure.

Changelog:

- Updated product schema in `schemas/product.js` to include `imagePath` and other required fields.

3. Migration Steps and Tools

Steps:

1. Fetched product data from the API using `axios`.
2. Validated and mapped API response fields to the Sanity schema.
3. Uploaded images to Sanity CMS using `client.assets.upload()`.
4. Created product documents in Sanity CMS using `client.create()`.

```

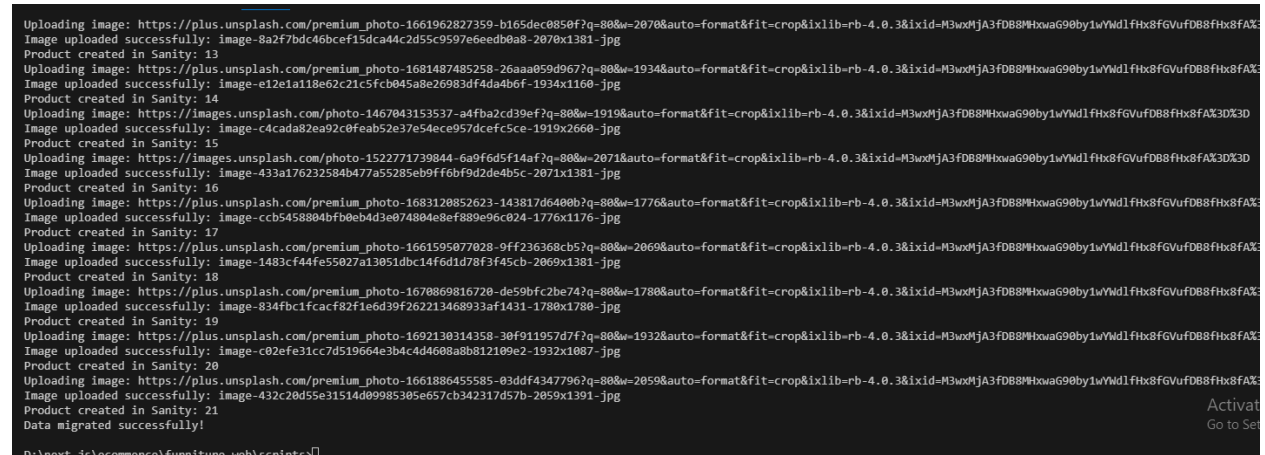
scripts > JS data-migration.mjs > importData
21 async function uploadImageToSanity(imageUrl) {
22 } catch (error) {
23   console.error('Failed to upload image:', imageUrl, error.message);
24   return null;
25 }
26
27 async function importData() {
28   try {
29     console.log('Migrating data, please wait...');
30
31     // Fetch products from the API
32     const response = await axios.get('https://template-0-beta.vercel.app/api/product');
33     const products = response.data;
34
35     console.log('Products fetched:', products);
36
37     for (const product of products) {
38       let imageRef = null;
39
40       if (product.imagePath) {
41         imageRef = await uploadImageToSanity(product.imagePath);
42       }
43
44       const sanityProduct = {
45         _type: 'product',
46         id: product.id,
47         name: product.name,
48         category: product.category,
49         description: product.description,
50         discountPercentage: product.discountPercentage,
51         isFeaturedProduct: product.isFeaturedProduct,
52         stockLevel: product.stockLevel,
53         price: parseFloat(product.price),
54         image: imageRef
55         ? {
56           _type: 'image',
57           asset: {
58             _type: 'reference',
59             _ref: imageRef,
60           },
61         }
62         : undefined,
63         imagePath: product.imagePath, // Store original image URL
64       };
65
66       await client.create(sanityProduct);
67       console.log('Product created in Sanity: ${sanityProduct.id}');
68     }
69
70     console.log('Data migrated successfully!');
71   } catch (error) {
72     console.error('Error in migrating data:', error.message);
73   }
74 }

```

Tools Used:

- **Node.js:** Runtime for executing the script.
- **Axios:** HTTP client for API calls.
- **Sanity Client:** For interacting with Sanity CMS.
- **Postman:** For testing API endpoints.

4. Screenshots



API Calls:

- Screenshot of successful API response showing product data.

Frontend Data Display:

- Screenshot of the products displayed in the frontend application.

Sanity CMS Fields:

- Screenshot of populated fields in the Sanity dashboard for the product schema.

5. Code Snippets

API Integration:

```
const response = await axios.get('https://template-0-beta.vercel.app/api/product');
const products = response.data.data;
```

Data Migration Script:

```
async function uploadImageToSanity(imageUrl) {
  try {
    console.log(`Uploading image: ${imageUrl}`);
    const response = await axios.get(imageUrl, { responseType:
'arraybuffer' });
    const buffer = Buffer.from(response.data);
    const asset = await client.assets.upload('image', buffer, {
      filename: imageUrl.split('/').pop(),
    });
    console.log(`Image uploaded successfully: ${asset._id}`);
    return asset._id;
  } catch (error) {
    console.error('Failed to upload image:', error.message);
    return null;
  }
}
```

6. Best Practices

- 1. Secure Sensitive Data:**
 - a. Used `.env` files for storing API keys and tokens.
- 2. Clean Coding Practices:**
 - a. Modularized functions for reusability.
 - b. Used descriptive variable names.
 - c. Added comments to explain logic.
- 3. Data Validation:**
 - a. Validated fields using schemas before migrating.
 - b. Logged discrepancies for further review.
- 4. Thorough Documentation:**
 - a. Included screenshots, testing notes, and changelogs.
- 5. Version Control:**
 - a. Committed changes frequently with meaningful messages.
 - b. Tagged milestones in the repository.

6. Comprehensive Testing:

- a. Tested edge cases like empty API responses and invalid data using Postman.

7. Peer Review:

- a. Shared code and documentation for feedback and incorporated suggestions.

7. Day 3 Checklist

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

Attachments:

- Screenshots of API responses, frontend data, and Sanity CMS fields.
- Full migration script and schema code.