

Day 3 - API Integration and Data Migration Report

Marketplace Name: Furniture Hub (General-Ecommerce)

Prepared by: Asifa Usman

1. API Integration Process:

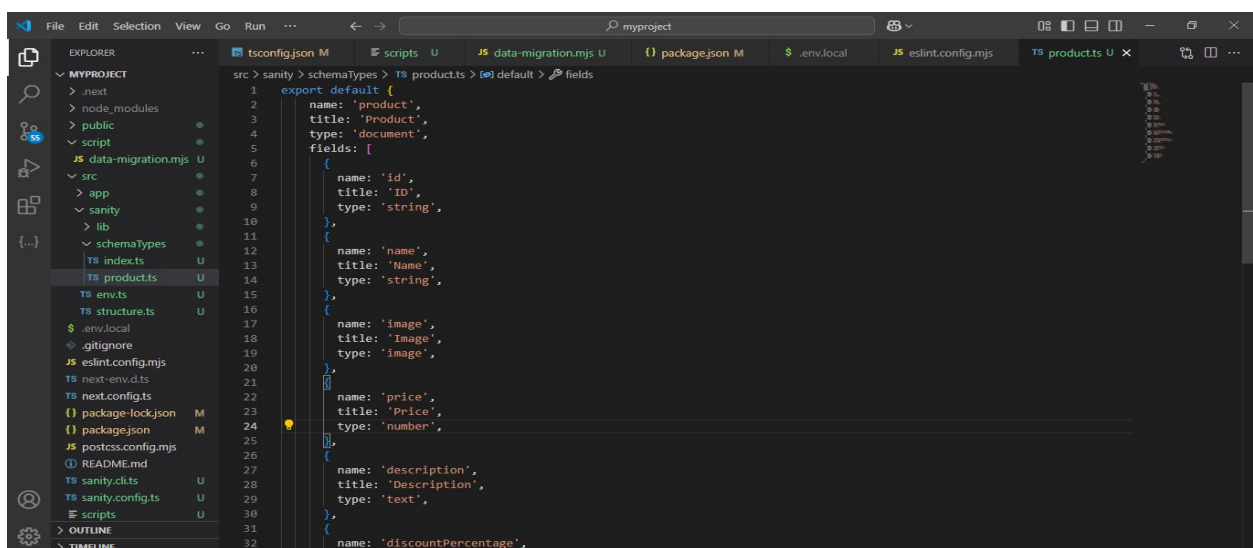
API Overview

- **API Used:** [Template 0 API]
- **API Documentation:**
- **Key Endpoints Used:**
 - /products - Fetch product listings.
 - /categories - Fetch product categories.
 - /orders - Create new orders.

Steps for API Integration

1. **Review API Documentation:**
 2. **Create Utility Functions:**
 3. **Render Data in Components:**
 4. **Error Handling:**
-

2. Schema:



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'src', 'sanity', and 'lib'. The code editor shows a TypeScript file named 'products.ts' with the following schema definition:

```
src > sanity > schemaTypes > TS products.ts > default > fields
1 export default {
2   name: 'product',
3   title: 'Product',
4   type: 'document',
5   fields: [
6     {
7       name: 'id',
8       title: 'ID',
9       type: 'string',
10    },
11    {
12      name: 'name',
13      title: 'Name',
14      type: 'string',
15    },
16    {
17      name: 'image',
18      title: 'Image',
19      type: 'image',
20    },
21    {
22      name: 'price',
23      title: 'Price',
24      type: 'number',
25    },
26    {
27      name: 'description',
28      title: 'Description',
29      type: 'text',
30    },
31    {
32      name: 'discountPercentage',
```

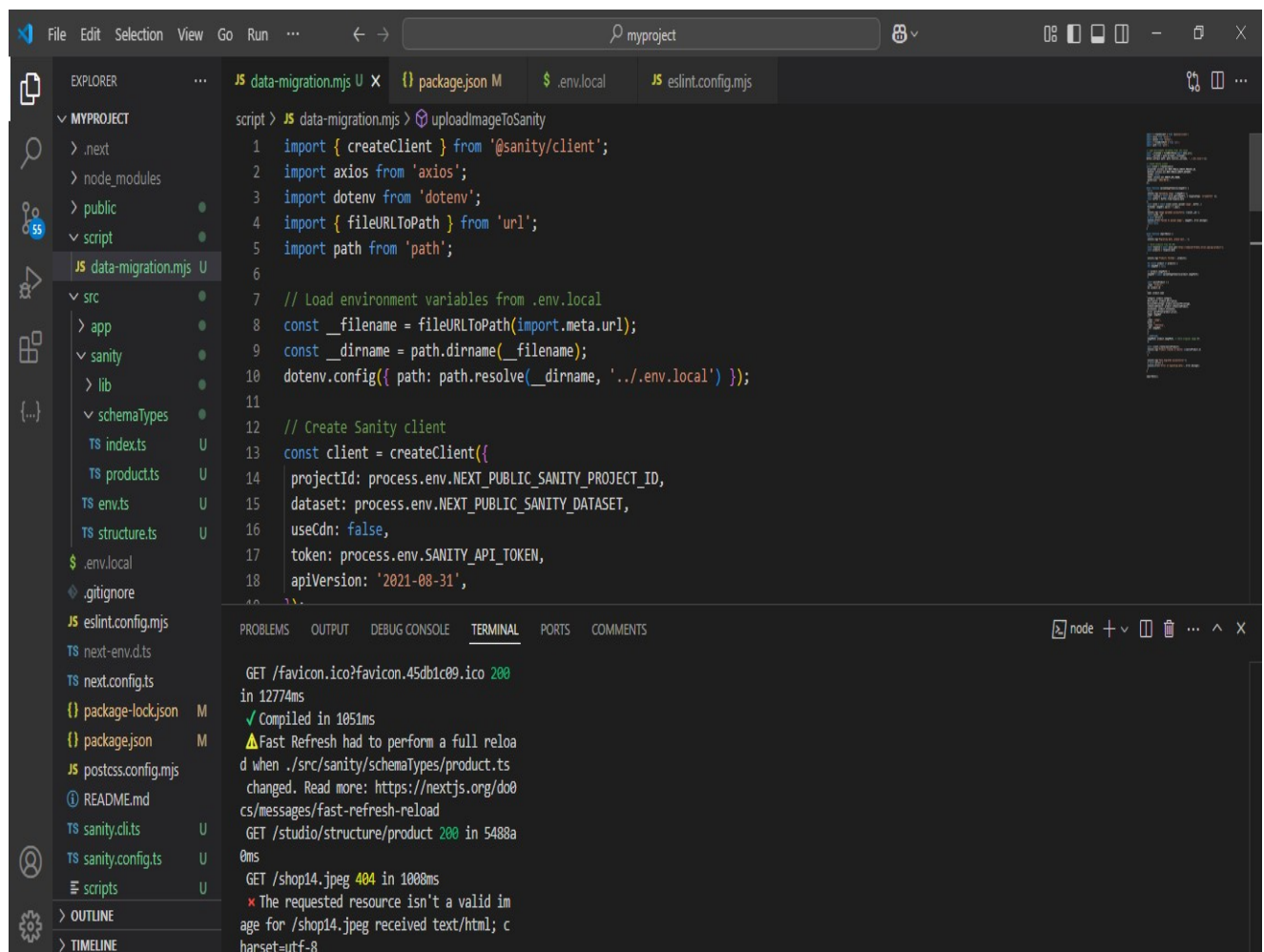
3. Data Migration Steps:

Migration Method

- **Method Used:** Script-based migration using the provided API.
- **Script Location:** [script/data-migration.mjs]

Steps for Data Migration

1. **Fetch Data from API:**
2. **Transform Data:**
3. **Upload Data to Sanity:**



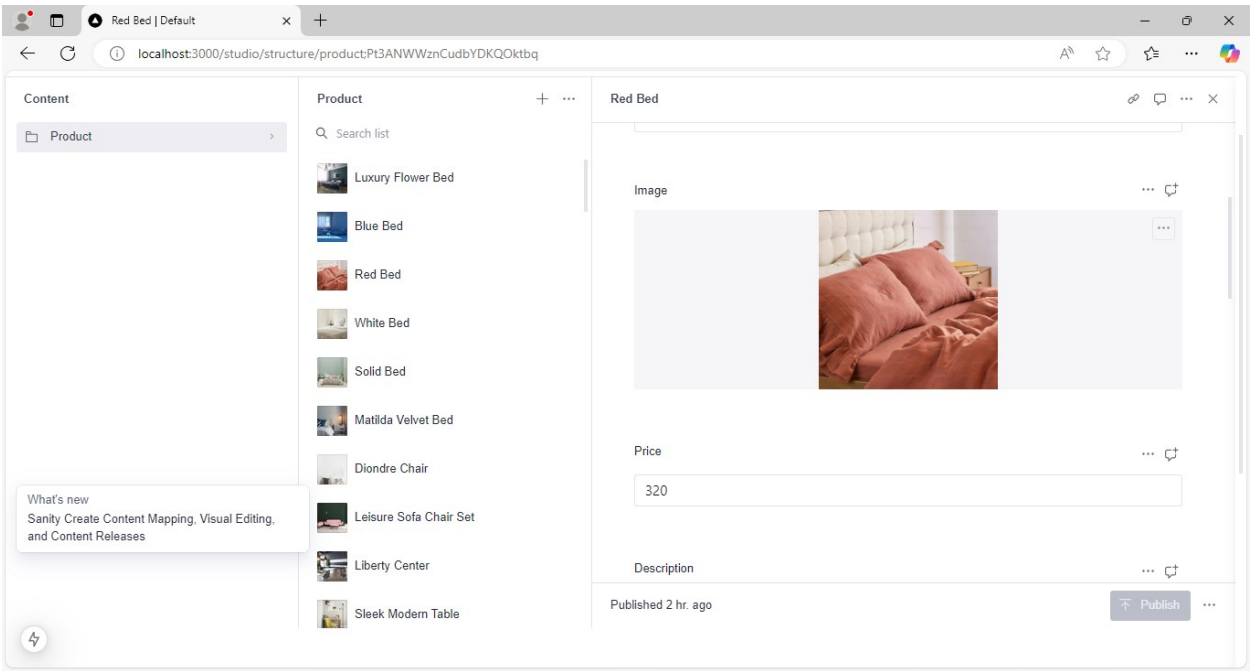
The screenshot displays a VS Code editor window for a project named 'myproject'. The Explorer sidebar on the left shows the file structure, including a 'script' directory containing 'data-migration.mjs'. The main editor area shows the content of 'data-migration.mjs', which is a JavaScript script for migrating data to Sanity. The script imports necessary modules, loads environment variables from '.env.local', creates a Sanity client, and defines functions for fetching and uploading data. The Terminal at the bottom shows the output of running the script, including successful API calls and a Fast Refresh reload message.

```
script > JS data-migration.mjs > uploadImageToSanity
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: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
15   dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
16   useCdn: false,
17   token: process.env.SANITY_API_TOKEN,
18   apiVersion: '2021-08-31',
19 });
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
GET /favicon.ico?favicon.45db1c89.ico 200
in 12774ms
✓ Compiled in 1051ms
⚠ Fast Refresh had to perform a full reload
when ./src/sanity/schemaTypes/product.ts
changed. Read more: https://nextjs.org/docs/messages/fast-refresh-reload
GET /studio/structure/product 200 in 5488a
0ms
GET /shop14.jpeg 404 in 1008ms
✗ The requested resource isn't a valid im
age for /shop14.jpeg received text/html; c
harset=utf-8
```

4. Sanity CMS



5. Self-Validation Checklist

Day 1: Business Focus Outcome Checklist:

| Task | Status | Notes |
|--------------------------------|--------|--|
| Choose Marketplace Type | ✓ | Selected [General E-Commerce/Q-Commerce/Rental E-Commerce]. |
| Define Business Goals | ✓ | Defined the problem, target audience, and unique value proposition. |
| Create Data Schema | ✓ | Drafted a schema with entities like products, orders, and customers. |
| Submit Business Goals & Schema | ✓ | Uploaded to the repository or shared on LinkedIn. |

Day 2: Technical Planning Outcome Checklist:

| Task | Status | Notes |
|-------------------------------|--------|---|
| Define Technical Requirements | ✓ | Outlined frontend, backend, and API requirements. |
| Design System | ✓ | Created a high-level diagram showing |

| | | |
|---------------------------------------|---|--|
| Architecture | | interactions between components. |
| Plan API Requirements | ✓ | Listed endpoints, methods, payloads, and responses. |
| Write Technical Documentation | ✓ | Documented system architecture, workflows, and API requirements. |
| Draft Sanity Schema | ✓ | Designed schemas for products, orders, and customers. |
| Submit Technical Documentation | ✓ | Uploaded to the repository. |

Day 3: API Integration and Data Migration Checklist:

| Task | Status | Notes |
|-----------------------------------|--------|--|
| Understand Provided API | ✓ | Reviewed API documentation and identified key endpoints. |
| Validate and Adjust Schema | ✓ | Adjusted schema to match API fields. |
| Migrate Data | ✓ | Successfully migrated data using a script. |
| Integrate API in Next.js | ✓ | Integrated API data into frontend components. |
| Prepare Submission | ✓ | Documented all steps and included screenshots. |

7. Final Notes

- **Challenges Faced:**
 - Initially struggled with mapping API fields to Sanity schema.
 - Resolved by adding transformation logic in the migration script.
 - **Key Learnings:**
 - Gained hands-on experience in API integration and data migration.
 - Improved understanding of Sanity CMS and Next.js.
 - **Next Steps:**
 - Refine the frontend UI for better user experience.
 - Add more error handling for API calls.
-