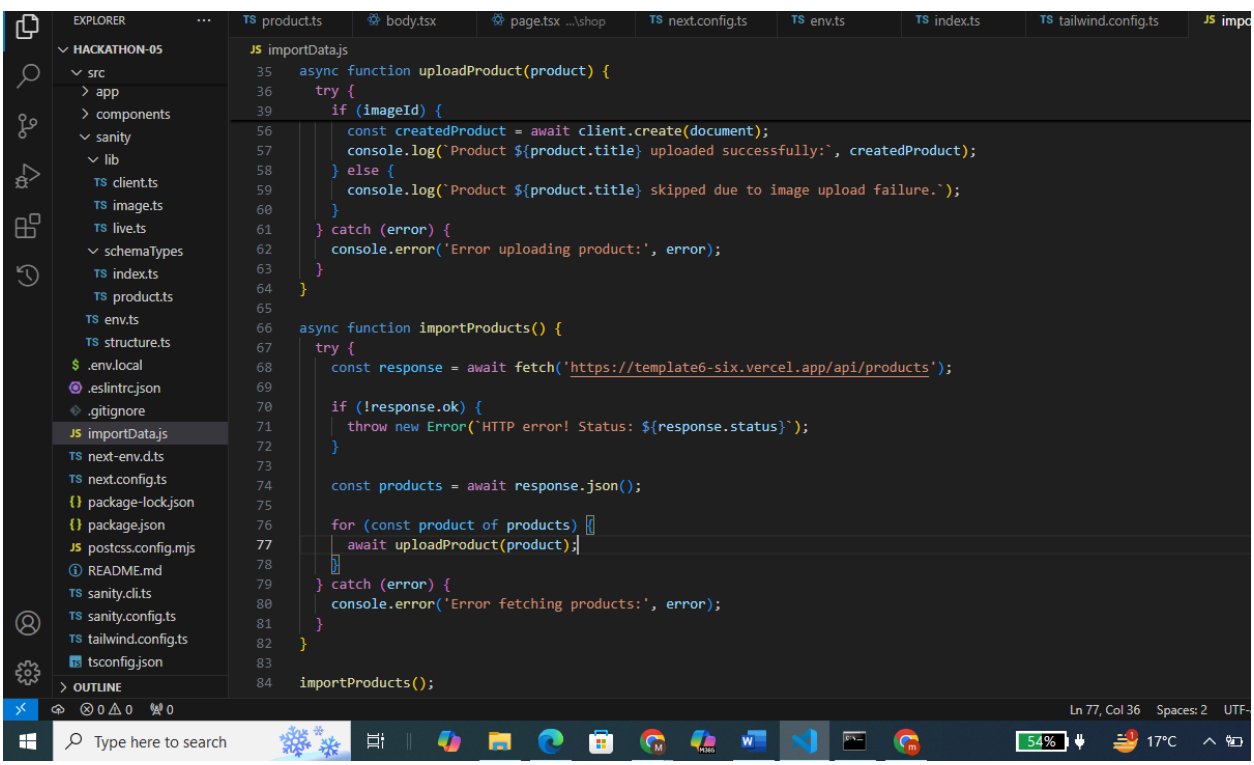
HACKATHON DAY 03
API INTEGRATION & DATA MIGRATION

Marketplace

This documentation outlines the work completed on Day 3 of the Furniro Ecommerce Marketplace hackathon. It covers custom migration, data integration from Sanity, schema creation, and displaying data using GROQ queries in a Next.js application. Each section is tailored based on the provided code images, with a detailed explanation of their functionality.

Custom Migration

The custom migration code was developed to streamline the process of transferring product data from an external API into the Sanity CMS database. This was essential for our hackathon project, as it allowed us to efficiently populate the content management system with real-world product data, complete with images and associated metadata. The migration script ensured that the data was properly structured and matched the schema defined in the project, reducing manual data entry efforts and avoiding errors.



Key Features of Custom Migration

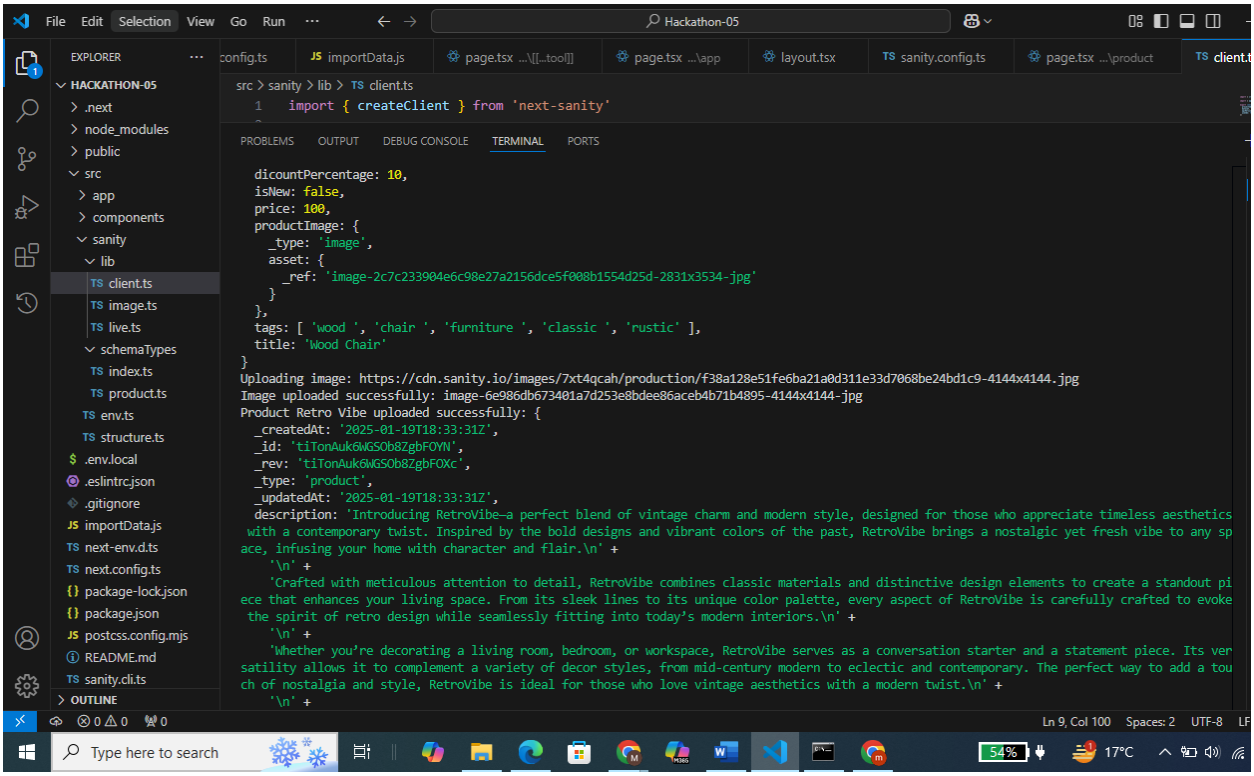
1) Sanity Client Configuration: The code initializes a client with the required projectId, dataset, and token.

2) Image Upload Handling: The uploadImageToSanity function fetches image files from external URLs and uploads them to Sanity, ensuring they are stored as assets.

3) Product Creation: The uploadProduct function creates a structured product document, linking uploaded image assets and ensuring data integrity.

4) Batch Processing: The importProducts function fetches all products from an external API and processes them sequentially.

Data Migration Become Successful



Migration Process Includes

1. Planning: Outline how the data was structured, validated, and prepared for migration.

2. Implementation: Highlight the steps taken to ensure data consistency and integrity during migration, such as:

    - Uploading high-quality product images.
    - Associating metadata with each product (e.g., tags, discount status, and descriptions).

3. Tools and Techniques: Describe the tools used for migration (e.g., Sanity CLI, APIs) and the strategies for managing large datasets.

4. Challenges and Solutions

- Challenge: Ensuring image references were properly linked to products.
    - Solution: Use of Sanity's image asset system for seamless integration.
- Challenge: Validating data accuracy for product descriptions and pricing.
    - Solution: Implementing data validation during migration.
- Challenge: Managing tags for consistent categorization across the collection.
    - Solution: Standardizing tag formats before import.

5. Outcome

- Successfully migrated four complete product entries with rich metadata and high-resolution images.
- Verified that all data is accessible and editable via Sanity's CMS.
- Ensured real-time updates are functional across all integrated platforms.

6. Features Highlighted in the Products

- Detailed descriptions and key features for each product.
- Associated high-quality images for enhanced visual representation.
- Inclusion of tags for improved discoverability.
- Differentiation of new arrivals and discounted items.

7. Impact

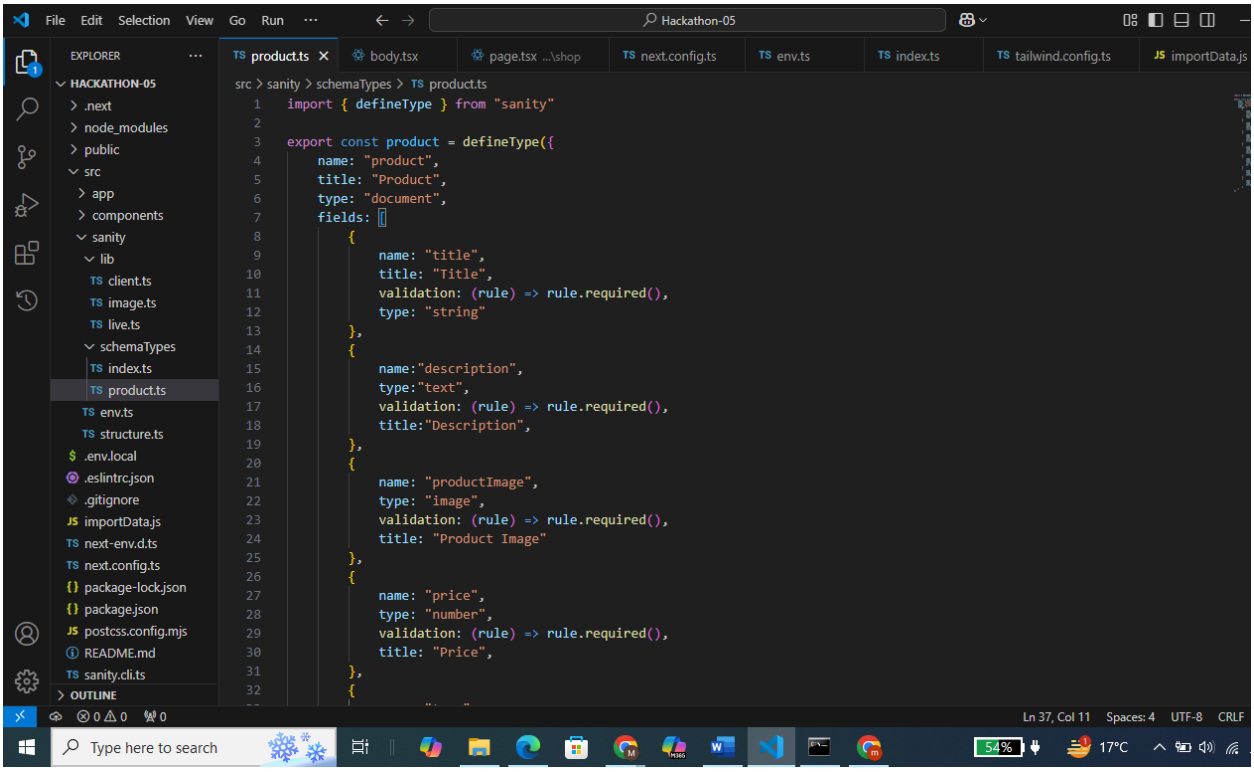
- Enhanced user experience with better content management and faster updates.
- Scalability for adding new products and collections.
- Improved data organization for future projects and features.

8. Future Plans

- Integration with e-commerce platforms for real-time product synchronization.

- Leveraging Sanity's GraphQL capabilities for advanced queries.
- Automating periodic updates and quality checks.

Schema Side



Sanity Product Schema Components

The product schema defines the structure for managing product information within the Sanity Content Studio. Each field serves a specific purpose to ensure accurate and organized data management. Below is a detailed breakdown of the schema components:

1. Schema Type Definition

- defineType: This function is used to define a Sanity document schema.
- name: The internal identifier for this schema (product). Used for referencing this schema programmatically within the Sanity ecosystem.
- title: The human-readable name of the schema (Product) that appears in the Sanity Studio.

2. Schema Fields

The fields array contains the structure and rules for each product property. Each field is customized for specific types of data:

a. Title

- name: title
- type: string
- validation: Ensures that every product must have a title.
- Purpose: Serves as the name of the product, ensuring products are easily identifiable.

b. Description

- name: description
- type: text
- validation: Ensures every product includes a description.
- Purpose: Provides detailed information about the product for users and search engines.

c. Product Image

- name: productImage
- type: image
- validation: Ensures every product is accompanied by at least one image.
- Purpose: Displays the visual representation of the product.

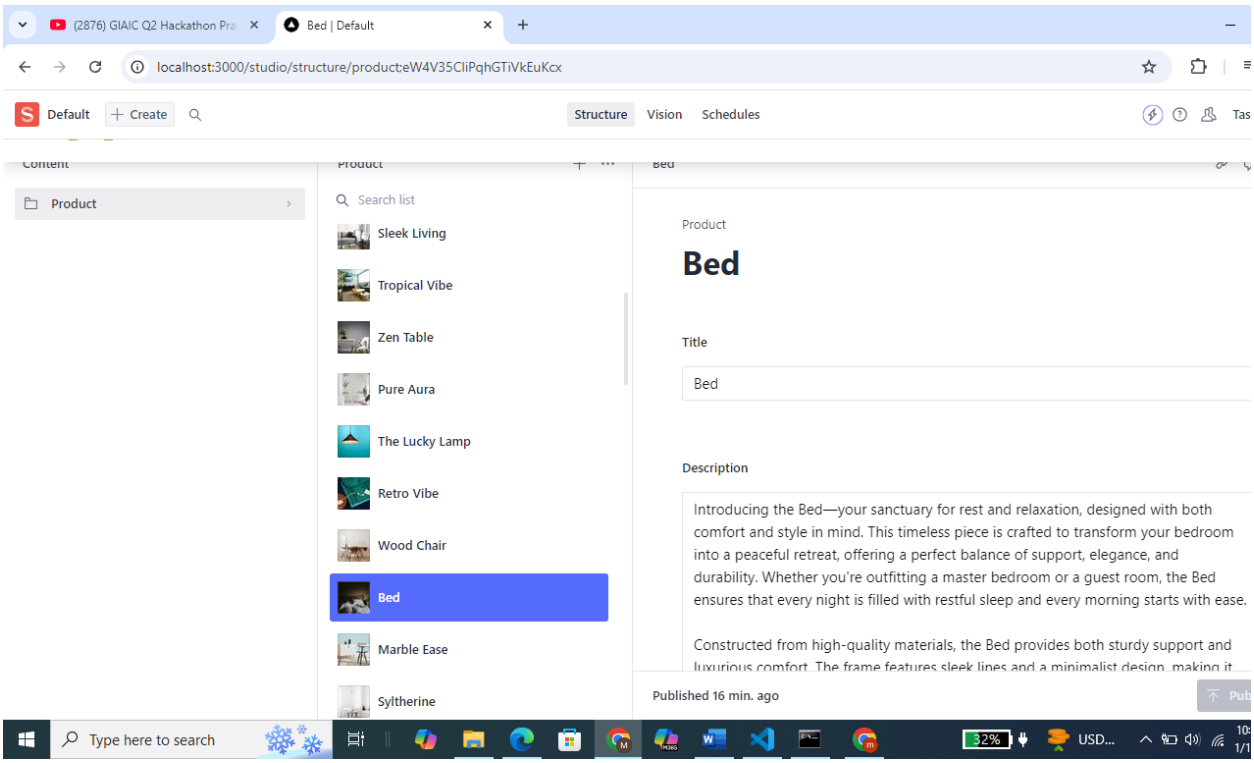
d. Price

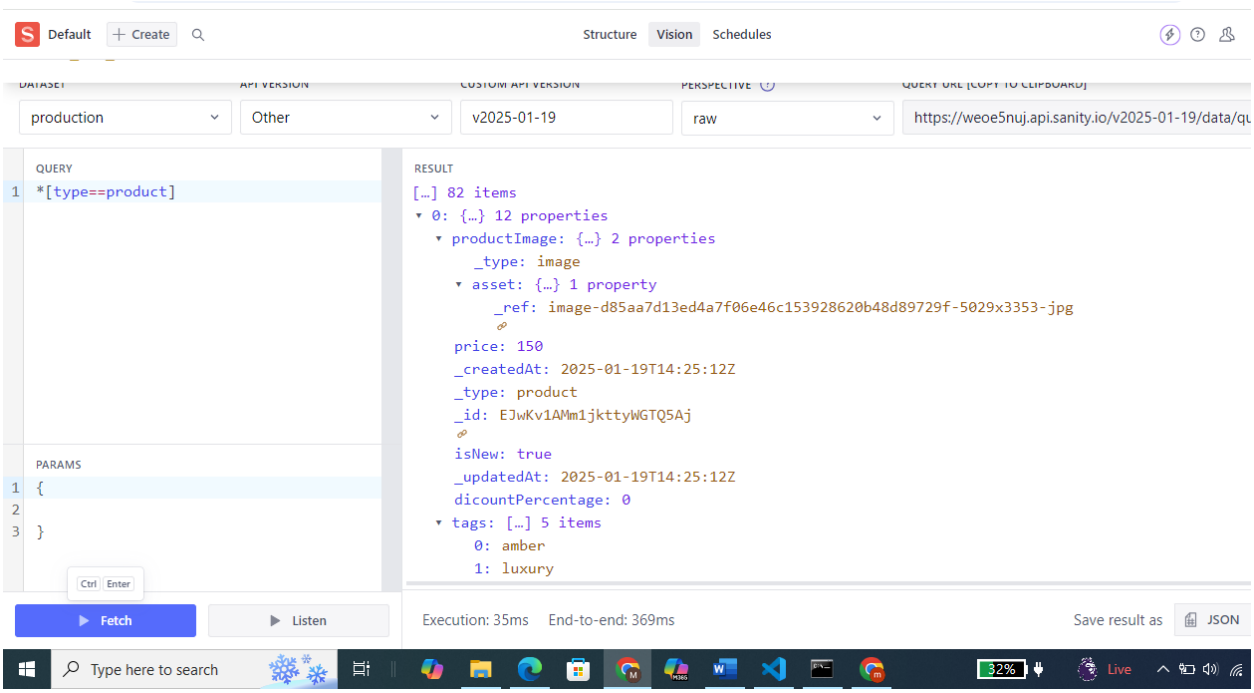
- name: price
- type: number
- validation: Ensures the price field is not left empty.
- Purpose: Defines the cost of the product, essential for e-commerce purposes.

e. Tags

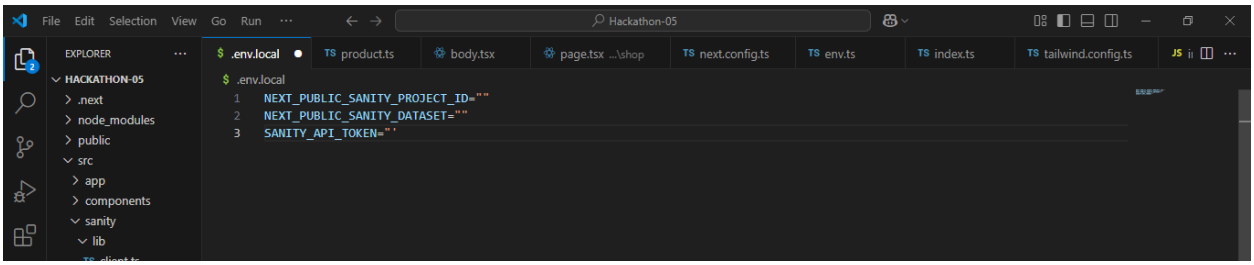
- name: tags
- type: array
- of: Specifies that the array contains strings.
- Purpose: Tags are used for categorization and improving searchability within the platform.

Sanity Product Schema





Environmental Variable:



1. **SANITY_PROJECT_ID**: The unique identifier for your Sanity project.

2. **SANITY_DATASET**: Specifies the dataset you're working with (e.g., production, development).

3. **SANITY_API_TOKEN**: Used for accessing the Sanity API in different environments, enabling secure content retrieval and management.

Benefits of Using Environmental Variables

- **Security**: Reduces the risk of leaking sensitive information.
- **Flexibility**: Easily switch between different environments (e.g., development to production) without modifying the code.
- **Scalability**: As the project grows, managing configurations via environmental variables keeps the code clean and manageable.
- **Portability**: Developers can easily replicate the project in a different environment (e.g., from one machine to another) by setting up the required environment variables.

Conclusion

On Day 3 of the Furniro Ecommerce Marketplace hackathon, we focused on API integration and data migration to ensure seamless product management within the Sanity CMS. We implemented a custom migration process to transfer real-world product data, including images and metadata, from an external API to Sanity. This streamlined data entry and ensured consistency across environments. By leveraging Sanity's schema system, we structured product data such as titles, descriptions, prices, and tags for easy management and display. Environmental variables played a key role in securing API access and ensuring flexibility across different environments. The successful migration resulted in rich product entries with high-quality images and accurate metadata, significantly improving the user experience, scalability, and data organization for future growth. The project is now poised for future enhancements, including e-commerce platform integration and automated updates.