Day 3 - API Integration Report –

Home inventory website

# 1. Introduction

# API Integration Overview: Describe the overall API integration process and its purpose in your marketplace (e.g., fetching product data, user information, etc.).

# Adjustments Made to Schemas: Briefly explain any changes or additions made to the schemas in your project to accommodate API data.

# 2. API Integration Process

##### Step-by-Step Process:

##### Describe how the API was integrated with your marketplace (e.g., using fetch or axios).

##### Include any configurations like API base URLs, headers, and authentication methods (using .env for secure storage of keys).

##### Explain how you structured the API responses and mapped them to your frontend components.

#### // Example of API call in Next.js const fetchData = async () => { const res = await fetch('https://api.example.com/products'); const data = await res.json(); return data; };

## 3. ****Adjustments Made to Schemas****

##### **Schema Updates**: Describe what changes were made to the database schema, if any, to accommodate the API data (e.g., adding new fields or data types).

##### **Field Validation**: Ensure the fields are validated properly during migration and API integration.

# day3 schema.png

### 4. ****Migration Steps and Tools Used****

##### **Tools**: Mention tools used for data migration (e.g., Sanity’s API or custom migration scripts).

##### **Migration Steps**:

##### Fetch data from the external API.

##### Validate and adjust data according to your schema.

##### Populate Sanity CMS fields.

##### Migrate data to your frontend.

##### **Code Snippet**:

##### migration.png

### Successful Data Display in Sanity CMS

The screenshot below shows the data successfully displayed in the Sanity CMS. After the migration, the relevant fields were populated with product information, ensuring that the content is ready for the frontend display.

##### day3.png

# **Final Product Listing Page**

The final product listing page showcases the migrated data, displaying each product’s details fetched from the API. The information is dynamically rendered, ensuring that users see real-time updates and accurate product details directly from the Sanity CMS.

