***Day 3 API Integration and Data Migration Report***

1. Sanity CMS Schema Design: To ensure the seamless handling of product data, I designed a schema called product in Sanity CMS. The schema includes the following fields:

Main Fields:

title: The product title (string type).

description: A detailed description of the product (text type).

productImage: The main product image (image type).

price: The price of the product (number type).

tags: An array of tags to categorize the product (array of strings).

discountPercentage: The discount percentage (number type).

isNew: A boolean flag indicating if the product is new (boolean type).

2.API Integration and Data Migration:

API Data Fetching: I fetched product data from an external API, which included details like images, titles, descriptions, prices, and tags. This data was then mapped to the corresponding fields in the Sanity CMS schema.

Data Population in Sanity CMS: After fetching the API data, I populated the product fields in Sanity CMS dynamically. This allowed for the automated population of product information, ensuring consistency and accuracy across the platform.

Data Migration: Using the Sanity CLI, I exported the dataset from Sanity CMS for backup purposes and later re-imported it for testing. This migration ensured that all data was properly structured and displayed as intended on the frontend.

3. Steps Taken for Data Migration:

Exporting Data: The first step was exporting the data from Sanity CMS using the Sanity CLI. This ensured that all product data was safely backed up before any further operations.

Verification of Data: The exported JSON structure was reviewed to ensure that all fields were populated correctly. This step ensured that the data would be accurately displayed when fetched and rendered on the frontend.

Re-importing Data: After verification, the dataset was re-imported into Sanity CMS. This confirmed that the data migration was successful and the system was working as expected.

4. Tools Used:

Sanity Studio: Used for schema creation, content management, and displaying product data.

Sanity CLI: Utilized for exporting and importing the dataset, ensuring data consistency and backup.

Conclusion: The API integration and data migration were successfully completed, enhancing the efficiency and scalability of the Bandage project. This integration streamlined the process of adding and updating product data in the marketplace, while the migration steps ensured data consistency and accuracy across the system. With this setup, the Bandage project is now more dynamic and easier to maintain.