DAY 3

API INTEGRATION REPORT-NIKE E-COMMERCE

1. How I Integrated the API

- I used the provided Products API to fetch product data like names, prices, stock levels, and images.
- The API was connected to my Next.js front-end by writing a utility function using fetch().This function fetched product data and passed it to the components to display on the website.
- To store the data, I run the provided data migration. mjScript, which fetched data from the API and migrated it into Sanity CMS.

2. Adjustments I Made to the Schema

The Sanity CMS schema for products (product.ts) needed adjustments to match the API fields.

- I Change the name to product Name to match my schema

3. Steps I Followed for Data Migration

API Review:

- I opened the API documentation and tested the products endpoint using Postman to understand the structure of the data.

Schema Validation:

- I compared the API fields with the product.ts schema in Sanity CMS.
- Updated the schema to ensure all fields matched the API structure.

Running the Data Migration Script:

- I run the provided data migration.mj script
- This script fetched the product data from the API and imported it into Sanity CMS.

VerifyingDatainSanityCMS:

- I logged into the Sanity Content Studio and checked that the products were successfully added, with correct fields like names, prices, stock, and images.

Connecting the Front-End:

- In Next.js, I created a utility function to fetch data from Sanity CMS using the @sanity/clientpackage.
- Fetched the products and displayed them on the Product Listing Page.

I successfully refined the 'Items' page by meticulously addressing the data fetching errors encountered while integrating API and Sanity data. This involved a combination of independent investigation and leveraging the valuable insights provided by ChatGPT.