

Minahil Fatima
426133
Template 3
DAY 3-API INTEGRATION REPORT-NIKE ECOMMERCE

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 ran the provided `datamigration.mjs` script, 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 `productName` 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 ran the provided datamigration.mjs script
- ✓ This script fetched the product data from the API and imported it into Sanity CMS.

Verifying Data in Sanity CMS:

- ✓ 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/client` package.
- ✓ Fetched the products and displayed them on the Product Listing Page.

And then I also update obviously my product detail cart bag page due to fetching api and sanity data it shows some errors which I resolve with the help of gpt and my self too