Day 3 - API Integration Report - [Nike Website]

1. API Integration Process

API Overview

- **Assigned Template:** [Template Name]
- **API Endpoints:**
  - **Products:** `/products`
  - **Categories:** `/categories`
  - **Order History:** `/orders` (if applicable)

Steps for API Integration

1. **Review API Documentation:**

   I began by reviewing the API documentation provided for the assigned template, identifying the key endpoints such as product listings (`/products`), categories (`/categories`), and order history (`/orders`).

2. **Create Utility Functions:**

I created utility functions to fetch data from the API using `fetch()` and `axios` for handling asynchronous calls. The utility functions were modularized for reusability across different components.

3. Render Data in Components:

I integrated the API data into the frontend by using React components. Data fetched from the API was displayed in a structured manner within components like Product Listings, Categories, etc.

4.Error Handling:

Implemented centralized logging for any errors encountered during the data fetch process. I used fallback data and skeleton loaders to enhance the user experience during loading.

2. Adjustments Made to Schemas

Schema Comparison & Adjustments

After reviewing the provided API data structure and my existing Sanity CMS schema, I made the following adjustments:

1. **Field Mapping:**
    - **API Field:** `product_title` → **Schema Field:** `name`
    - **API Field:** `product_description` → **Schema Field:** `description`
    - **API Field:** `product_price` → **Schema Field:** `price`

2. Field Types and Relationships:

I adjusted field types and ensured relationships were correctly mapped. For instance, the product category field was linked to the categories collection in Sanity to maintain relational integrity.

3. Validation:

I validated the schema by checking for data type consistency and ensuring that each field adhered to the required data type and constraints.

3. Migration Steps and Tools Used

Data Migration Method

- **Method Used:**

  I chose to use the provided API to fetch and migrate data into Sanity CMS. The data was transformed using scripts and mapped to my schema.

  Fetching Data:

  1. I wrote a script using `axios` to fetch the data from the API endpoints.

  2. Data Transformation:

  The fetched data was transformed (mapped field names, data types) to align with my Sanity CMS schema.

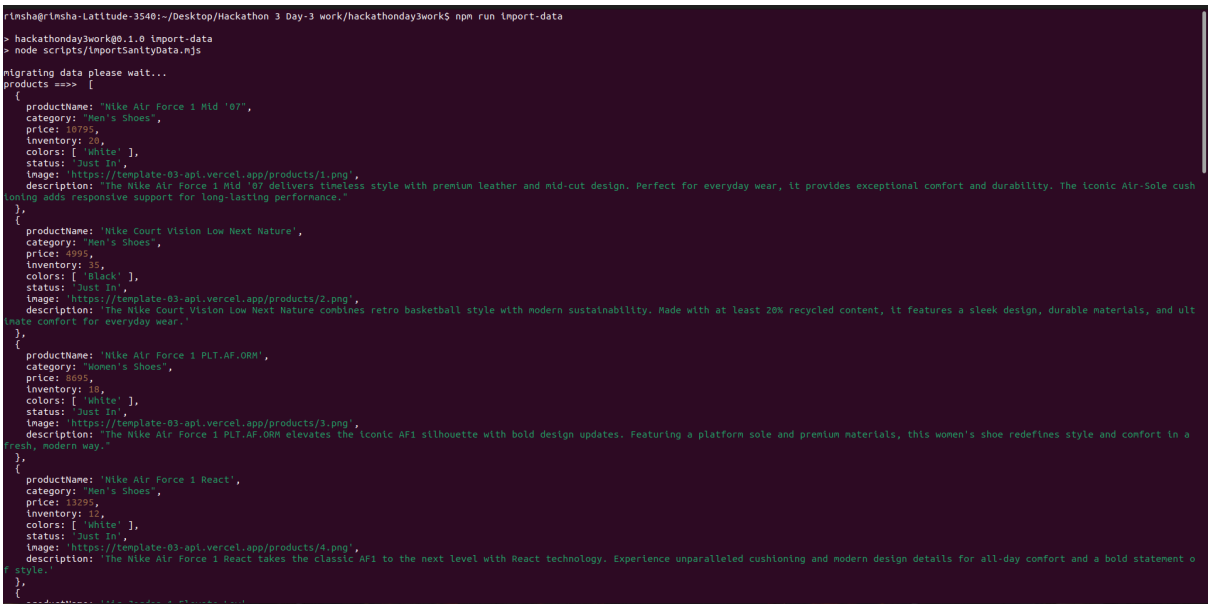
  3. Importing Data into Sanity:

  I used Sanity's API client to upload the transformed data into the Sanity CMS, ensuring it populated correctly.

4. Screenshots

API Calls

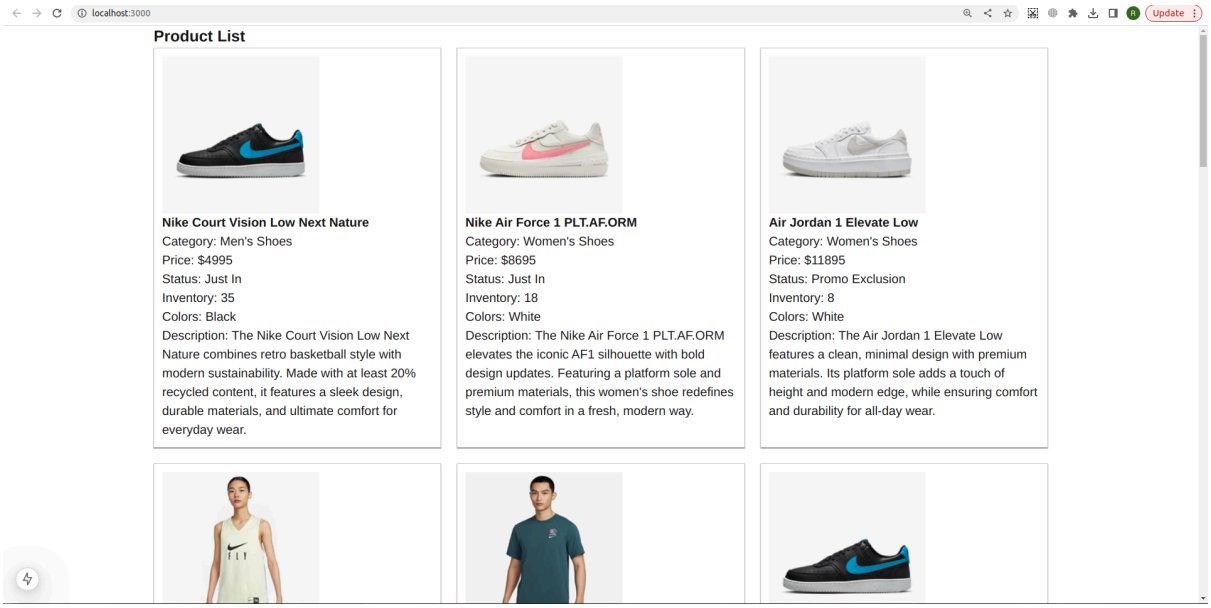
- **Screenshot 1: API response from `/products` endpoint**

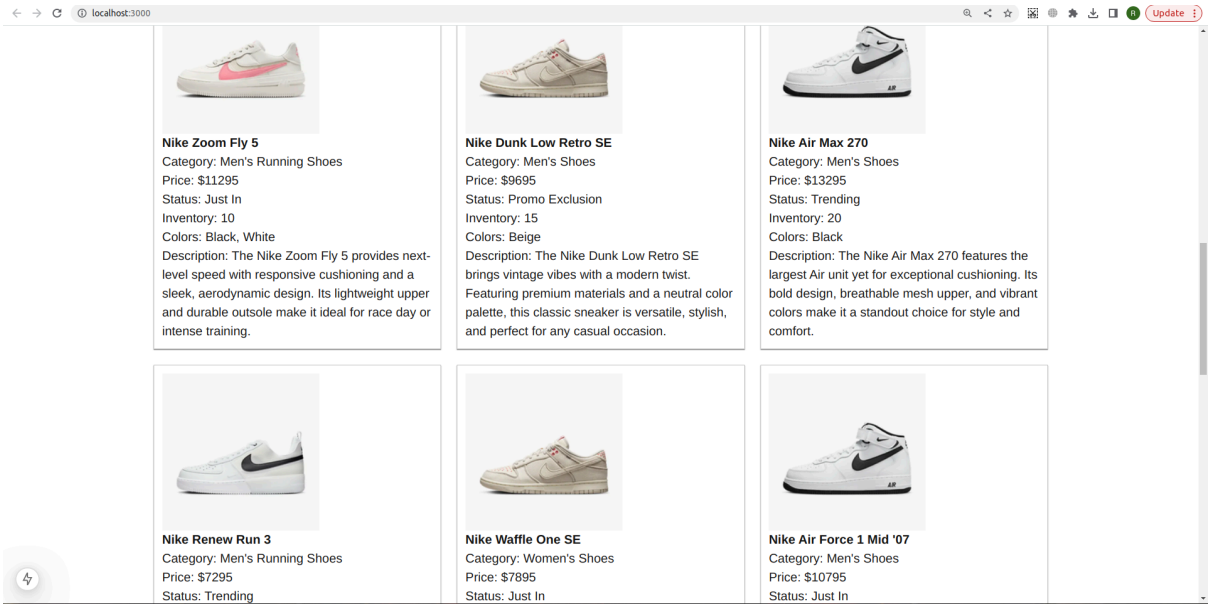


```
rimsha@rimsha-Latitude-3540:~/Desktop/Hackathon 3 Day-3 work/hackathonday3work$ npm run import-data

> hackathonday3work@0.1.0 import-data
> node scripts/importSanityData.mjs

migrating data please wait...
products ===>  [
  {
    productName: 'Nike Air Force 1 Mid '07',
    category: "Men's Shoes",
    price: 10795,
    inventory: 20,
    colors: [ 'White' ],
    status: 'Just In',
    image: 'https://template-03-api.vercel.app/products/1.png',
    description: "The Nike Air Force 1 Mid '07 delivers timeless style with premium leather and mid-cut design. Perfect for everyday wear, it provides exceptional comfort and durability. The iconic Air-Sole cush
ioning adds responsive support for long-lasting performance."
  },
  {
    productName: 'Nike Court Vision Low Next Nature',
    category: "Men's Shoes",
    price: 4995,
    inventory: 35,
    colors: [ 'Black' ],
    status: 'Just In',
    image: 'https://template-03-api.vercel.app/products/2.png',
    description: 'The Nike Court Vision Low Next Nature combines retro basketball style with modern sustainability. Made with at least 20% recycled content, it features a sleek design, durable materials, and ult
imate comfort for everyday wear.'
  },
  {
    productName: 'Nike Air Force 1 PLT.AF.ORM',
    category: "Women's Shoes",
    price: 8695,
    inventory: 18,
    colors: [ 'White' ],
    status: 'Just In',
    image: 'https://template-03-api.vercel.app/products/3.png',
    description: "The Nike Air Force 1 PLT.AF.ORM elevates the iconic AF1 silhouette with bold design updates. Featuring a platform sole and premium materials, this women's shoe redefines style and comfort in a
fresh, modern way."
  },
  {
    productName: 'Nike Air Force 1 React',
    category: "Men's Shoes",
    price: 13295,
    inventory: 12,
    colors: [ 'White' ],
    status: 'Just In',
    image: 'https://template-03-api.vercel.app/products/4.png',
    description: 'The Nike Air Force 1 React takes the classic AF1 to the next level with React technology. Experience unparalleled cushioning and modern design details for all-day comfort and a bold statement o
f style.'
  },
  {
```

Data Displayed in Frontend

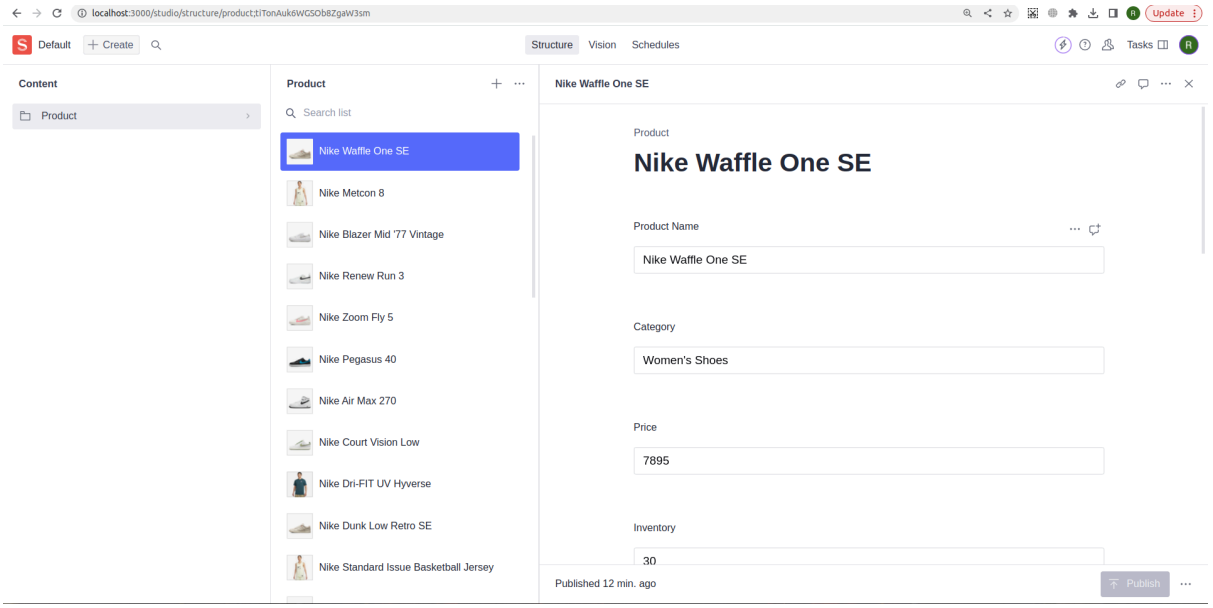
- **Screenshot 2: Products displayed on the frontend after API integration**





Populated Sanity CMS Fields

- **Screenshot 3: Sanity CMS populated with data from the API**

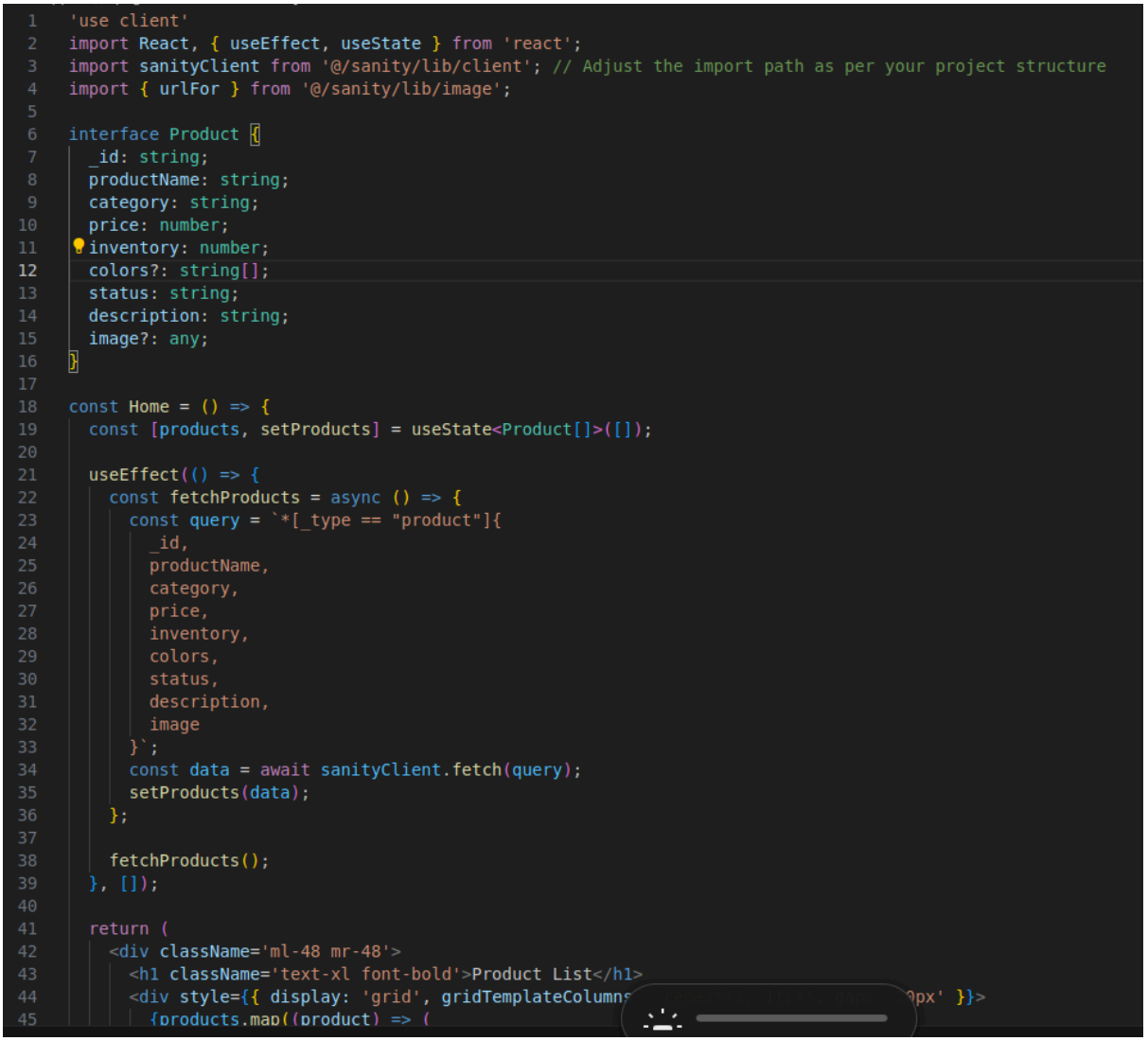


5. Code Snippets

API Integration Script

```
import { createClient } from '@sanity/client';
import axios from 'axios';
import dotenv from 'dotenv';
import { fileURLToPath } from 'url';
import path from 'path';

// Load environment variables from .env.local
const __filename = fileURLToPath(import.meta.url);
const __dirname = path.dirname(__filename);
dotenv.config({ path: path.resolve(__dirname, '../.env.local') });

// Create Sanity client
const client = createClient({
  projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
  dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
  useCdn: false,
  token: process.env.SANITY_API_TOKEN,
  apiVersion: '2021-08-31'
});


async function uploadImageToSanity(imageUrl) {
  try {
    console.log(`Uploading image: ${imageUrl}`);
    const response = await axios.get(imageUrl, { responseType: 'arraybuffer' });
    const buffer = Buffer.from(response.data);
    const asset = await client.assets.upload('image', buffer, {
      filename: imageUrl.split('/').pop()
    });
    console.log(`Image uploaded successfully: ${asset._id}`);
    return asset._id;
  } catch (error) {
    console.error('Failed to upload image:', imageUrl, error);
    return null;
  }
}

async function importData() {
  try {
    console.log('migrating data please wait...');

    // API endpoint containing car data
    const response = await axios.get('https://template-03-api.vercel.app/api/products');
    const products = response.data.data;
    console.log("products ==>> ", products);
```

Sanity Import Script



```tsx
'use client'
import React, { useEffect, useState } from 'react';
import sanityClient from '@/sanity/lib/client'; // Adjust the import path as per your project structure
import { urlFor } from '@/sanity/lib/image';

interface Product {
  _id: string;
  productName: string;
  category: string;
  price: number;
  inventory: number;
  colors?: string[];
  status: string;
  description: string;
  image?: any;
}

const Home = () => {
  const [products, setProducts] = useState<Product[]>([]);

  useEffect(() => {
    const fetchProducts = async () => {
      const query = `*[_type == "product"]{
        _id,
        productName,
        category,
        price,
        inventory,
        colors,
        status,
        description,
        image
      }`;
      const data = await sanityClient.fetch(query);
      setProducts(data);
    };

    fetchProducts();
  }, []);

  return (
    <div className='ml-48 mr-48'>
      <h1 className='text-xl font-bold'>Product List</h1>
      <div style={{ display: 'grid', gridTemplateColumns
                                                        px' }}>
        {products.map((product) => (
```

Updated Schema in Sanity

```
export const productSchema = {
    name: 'product',
    title: 'Product',
    type: 'document',
    fields: [
      {
        name: 'productName',
        title: 'Product Name',
        type: 'string',
      },
      {
        name: 'category',
        title: 'Category',
        type: 'string',
      },
      {
        name: 'price',
        title: 'Price',
        type: 'number',
      },
      {
        name: 'inventory',
        title: 'Inventory',
        type: 'number',
      },
      {
        name: 'colors',
        title: 'Colors',
        type: 'array',
        of: [{ type: 'string' }],
      },
      {
        name: 'status',
        title: 'Status',
        type: 'string',
      },
      {
        name: 'image',
        title: 'Image',
        type: 'image',
        options: {
          hotspot: true,
        },
      },
      {
```

Presented By:

Rimsha Mukhtar

Slot: Saturday 2 to 5