**API Integration Process for Next.js with Sanity CMS**

**Overview**

This document outlines the process of integrating a Next.js project with Sanity CMS, creating schemas, and using API calls to fetch and display product data. Additionally, it describes the method of handling images by storing them locally and linking their paths in Sanity CMS.

**Step 1: Setting Up Sanity CMS**

1. **Initialize the Sanity Project:**
   * Start by creating a new project in Sanity and configuring it according to our requirements.
   * Deploy the Sanity Studio to manage our website content.
2. **Create Schemas:**
   * Design schemas in Sanity CMS to define the structure of our data. For example, a schema for products can include fields like title, description, price, and image path.
3. **Upload Products:**
   * Use the Sanity Studio interface to upload 50 product entries. Each product entry should include details such as name, description, price, and the corresponding image path.
4. **Handle Images:**
   * Save all product images in a public folder within the Next.js project.
   * Record the file paths of these images in the relevant field within the Sanity CMS.

**Step 2: Setting Up the Next.js Project**

1. **Install Sanity Client:**
   * Add the Sanity client to our Next.js project for connecting to the Sanity CMS.
2. **Configure the Client:**
   * Set up a configuration file to establish a connection between Next.js and Sanity CMS by providing project credentials.

**Step 3: Fetching Data from Sanity CMS**

1. **Query Data:**
   * Use Sanity’s GROQ (Graph-Relational Object Queries) to fetch product data. Queries should retrieve all necessary fields such as title, description, price, and image paths.
2. **Implement Data Fetching:**
   * Configure data fetching in Next.js using methods like getStaticProps or getServerSideProps to fetch product information from Sanity CMS.

**Step 4: Displaying Data in Next.js**

1. **Render Product Information:**
   * Use the fetched data to display product details on the frontend. Include product titles, descriptions, prices, and images.
2. **Image Integration:**
   * Use the image paths stored in Sanity CMS to render images saved locally in the project folder.

**Step 5: Testing and Deployment**

1. **Test the Integration:**
   * Verify that all products are displayed correctly, ensuring that data and images load seamlessly.
2. **Deploy the Project:**
   * Deploy the integrated project on a hosting platform like Vercel to make it accessible to users.

This process ensures a smooth and efficient integration between Next.js and Sanity CMS, providing a scalable and dynamic solution for managing and displaying product data.