Steps to Integrate External API Data into a Sanity CMS and Display it in a Next.js Application

1. **Define a Schema in Sanity for the API Data:**
    a. Open your Sanity Studio project.
    b. Navigate to the `schemas` directory and create a new schema file, e.g., product.ts
    c. Define the schema fields that correspond to the structure of the API data.

```
export default {
  name: 'product',
  type: 'document',
  title: 'Product',
  fields: [
   {
    name: 'name',
    type: 'string',
    title: 'Name',
   },
   {
    name: 'image',
    type: 'image',
    title: 'Image',
    options: {
     hotspot: true, // Enable image cropping
    },
   },
   {
    name: 'price',
    type: 'number',
    title: 'Price',
   },
   {
    name: 'description',
    type: 'text',
    title: 'Description',
   },
   {
```

```
    name: 'discountPercentage',
    type: 'number',
    title: 'Discount Percentage',
   },
   {
    name: 'isFeaturedProduct',
    type: 'boolean',
    title: 'Featured Product',
   },
   {
    name: 'stockLevel',
    type: 'number',
    title: 'Stock Level',
   },
   {
    name: 'category',
    type: 'string',
    title: 'Category',
   },
  ],
 };
```

d. Register the schema in the `schema.js` file.

```
import product from './product';
export default createSchema({
  name: 'default',
  types: schemaTypes.concat([product]),
});
```

2. **Fetch Data from an External API:**
    a. Use a JavaScript file or a script to fetch data from the external API.
    b. Use `fetch` or a library like `axios` to retrieve the data.
3. **Insert the Data into Sanity:**
    a. Use Sanity's client library to insert the fetched data.

```
import { createClient } from '@sanity/client';
```

```
import fetch from 'node-fetch';

// Initialize Sanity client
const client = createClient({
  projectId: "id",
  dataset: "production",
  useCdn: false, // Set to true if you want faster reads
  apiVersion: '2025-01-13',
  token: "token", // Replace with your Sanity token
});

// Function to upload an image to Sanity
async function uploadImageToSanity(imageUrl) {
  try {
    console.log(` Uploading image: ${imageUrl}` );

    const response = await fetch(imageUrl);
    if (!response.ok) {
      throw new Error(` Failed to fetch image: ${imageUrl}` );
    }

    const buffer = await response.arrayBuffer();
    const bufferImage = Buffer.from(buffer);

    const asset = await client.assets.upload('image', bufferImage, {
      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;
  }
}

// Function to upload a single product to Sanity
async function uploadProduct(product) {
```

```
 try {
  const imageId = await uploadImageToSanity(product.imagePath);

  if (imageId) {
   const document = {
    _type: 'product',
    id: product.id,
    name: product.name,
    image: {
     _type: 'image',
     asset: {
      _ref: imageId,
     },
    },
    price: parseFloat(product.price), // Ensure the price is a number
    description: product.description,
    discountPercentage: product.discountPercentage,
    isFeaturedProduct: product.isFeaturedProduct,
    stockLevel: product.stockLevel,
    category: product.category,
   };

   const createdProduct = await client.create(document);
   console.log(` Product "${product.name}" uploaded successfully: `,
createdProduct);
  } else {
   console.log(` Product "${product.name}" skipped due to image upload
failure. `);
  }
 } catch (error) {
  console.error('Error uploading product:', error);
 }
}

// Function to fetch products from the provided API and upload them to
Sanity
async function migrateProducts() {
 try {
```

```
    const response = await fetch('https://template-0-
beta.vercel.app/api/product');

    if (!response.ok) {
      throw new Error(` HTTP error! Status: ${response.status}` );
    }

    const products = await response.json();

    for (const product of products) {
      await uploadProduct(product);
    }
  } catch (error) {
    console.error('Error fetching products:', error);
  }
}

// Start the migration
migrateProducts();
```

```
import { createClient } from '@sanity/client';


const client = createClient({
  projectId: 'koprfd8a', // Replace with your Sanity project ID
  dataset: 'production', // Replace with your dataset name
  useCdn: false,
  token:
'ske6miMVTCpdgBjy8vTl9IBmPBZJ0ykOIWlfby0ycDMOXVq5tRSQJ8aywym7P4P3UyshI
MyMLUY1lu2yww9Twlmg13K8wPI7irasbVecsEmDe1j3uLgnjUJ5EwmdLdX6Jwaw8ahP0ku
U8atkqPYbV9l2PJ5ijFuq6wOHopiH9Yf1CMq3DOWZ',
});


export default client;
```

4. **Retrieve and Display the Data in a Next.js Application:**
    a. Use `getStaticProps` or `getServerSideProps` to fetch data from Sanity.

```
import Hero from '@/app/Components/Hero';
import ShopProduct from '@/app/Components/ShopProduct';
import NewArrivals from '@/app/Components/NewArrivals';
import client from "@/sanity/lib/client";
import Image from 'next/image';
import Link from 'next/link';

interface Product {
  _id: string;
  name: string;
  description: string;
  price: string;
  image_url: string;
  rating: number;
}

const getProducts = async (): Promise<Product[]> => {
  const products = await client.fetch(
    `*[_type=="product"][0..21]{
     _id,
     name,
     description,
     price,
     "image_url": image.asset->url,
     rating
    }`
  );
  return products;
};

const Home = async () => {
  const products = await getProducts();
  console.log(products);
  return (
    <div>
      <Hero />
      <ShopProduct />
```

```
    {/* Top Picks Section */}
    <div className="w-full min-h-[800px]">
      <div className="flex flex-col items-center text-center">
        <p className="font-[500] text-[36px] leading-[54px] mt-20">
          Top Picks For You
        </p>
        <p className="font-[500] text-[16px] leading-[24px] text-[#9F9F9F] mt-5 max-w-2xl">
          Find a bright ideal to suit your taste with our great selection of suspension, floor, and table lights.
        </p>
      </div>

      <div className="flex justify-center">
        <div className="grid grid-cols-1 xs:grid-cols-2 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-6 mt-10 px-4 sm:px-6 w-full">
          {products.map((product) => (
            <div
              key={product._id}
              className="bg-white shadow-lg rounded-lg overflow-hidden cursor-pointer transition-transform hover:scale-105"
            >
              <Image
                src={product.image_url}
                alt={product.name}
                width={500}
                height={500}
                className="w-full h-48 object-cover"
              />
              <div className="p-4">
                <h3 className="text-lg font-semibold text-gray-900">
                  {product.name}
                </h3>
                <p className="text-gray-600 mt-2">{product.price}</p>
              </div>
            </div>
          ))}
        </div>
      </div>
```

```
      {/* View More Button */}
      <div className="flex justify-center mt-28">
       <Link href='../Shop'>
        <p className="underline underline-offset-8 mt-2 cursor-pointer font-[500] text-
[16px] transition-transform hover:scale-105 hover:text-gray-700">
          View More
        </p>
       </Link>
      </div>
    </div>

    <NewArrivals />

    {/* Our Blog Section */}
    <div className="w-full min-h-[844px] bg-[#FFFFFF] px-11">
     <div className="flex flex-col items-center text-center">
      <p className="font-[500] text-[36px] leading-[54px]">Our Blogs</p>
      <p className="text-[#9F9F9F] font-[500] text-[16px] leading-[24px] mt-4">
       Find a bright idea to suit your taste with our great selection
      </p>
     </div>

     <div className="flex justify-center items-center">
      <div className="mt-10 grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 gap-6 px-5 w-
full">
       {products.map((product) => (
        <div
         key={product._id}
         className="bg-white shadow-lg rounded-lg overflow-hidden cursor-pointer
transition-transform hover:scale-105"
        >
         <Image
          src={product.image_url}
          alt={product.name}
          width={500}
          height={500}
          className="w-full h-48 object-cover"
```

```
            />
            <div className="p-4">
              <h3 className="text-lg font-semibold text-gray-900">
                {product.name}
              </h3>
            </div>
            <div className="text-gray-600 mt-2 flex justify-center">
              <Link href="/Blogpage">
                <p className="underline hover:text-gray-700">
                  {product.price}
                </p>
              </Link>
            </div>
          </div>
        ))}
      </div>
    </div>

    <div className="flex justify-center mt-28">
      <Link href='/Blogpage'>
        <p className="underline underline-offset-8 mt-2 cursor-pointer font-[500] text-
[20px] transition-transform hover:scale-105 hover:text-gray-700">
          View All Post
        </p>
      </Link>
    </div>
  </div>

  {/* Contact to Instagram Section */}
  <div className="relative w-full h-auto">
    <div className="w-full h-[450px]">
      <Image
        src={"/pic12.png"}
        alt="Instagram Banner"
        width={1440}
        height={450}
        className="w-full h-full object-cover"
      />
```

```
      </div>

      <div className="absolute inset-0 flex flex-col items-center justify-center space-y-4">
       <div className="text-center">
        <p className="font-bold text-[40px] md:text-[60px] leading-[50px] md:leading-
[90px]">
          Our Instagram
        </p>
        <p className="font-[400] text-[16px] md:text-[20px] leading-[24px] md:leading-
[30px]">
          Follow our store on Instagram
        </p>
       </div>
       <div>
        <Link href='https://www.instagram.com/'>
         <button className="w-[200px] h-[50px] md:w-[255px] md:h-[64px] rounded-full
bg-white transition-transform hover:scale-105 text-black font-[500] text-[16px] md:text-
[20px] drop-shadow-lg">
           Follow Us
         </button>
        </Link>
       </div>
      </div>
    </div>
   </div>
  );
};

export default Home;
```

By following these steps, you can integrate external API data into Sanity CMS and display it in a Next.js application.