Steps to migrate API Data into Sanity and Sanity to Frontend

(API → Sanity → Frontend)

The API, API documentation and the Sanity Schema and was provided by our faculty teachers (Sir Anas and Sir Ubaid Ur Rehman) according to template

API Link 🔗: https://next-ecommerce-template-4.vercel.app/api/product
API doc 📄:
https://github.com/anasseth/next-ecommerce-template4/blob/master/README.md
Sanity Schema:
https://github.com/anasseth/next-ecommerce-template4/blob/master/src/sanity/schemaTypes/product.ts

Problem:
There were 21 products in the provided API. If we add products data manually in sanity. It will be time consuming.

Solution:
We can import data from API and import it to the sanity by the help of import data script. (axios) And then fetch data from sanity to the frontend. By this implementation, this will be a more complex task. By doing the following process, we can easily achieve the same thing discussed before by the help of import data into sanity and then fetch data from sanity after that display it on the frontend.

Step 1:

- Setup Sanity(after creating Nextjs setup):

```
npm create sanity@latest -- --template clean --create-project
"learning-sanity-project" --dataset production
```

Step 2:

Navigate to sanity studio by writing `https://localhost:3000/studio`

Setup environment variables by making a .env file on the root for security purposes. (people can see your env variables from GitHub and can use it in their project)
To find the sanity api token. Navigate to the API tab and then go to the token. Generate it via ‘developer’ or ‘editor’

Like this:
```
NEXT_PUBLIC_SANITY_API_VERSION=2025-01-15
NEXT_PUBLIC_SANITY_PROJECT_ID=8brhlsnp
NEXT_PUBLIC_SANITY_DATASET=production
SANITY_API_TOKEN=sk8S7JYhy8WiFnxW4xMmTSbW663b0mAIR4MkXNIQFzDnUQox5N34Sirew1jvf
aMONTdyOQRpATgSA5Q0lK9w2aWlyr8cWRrrAT4rAgowhTAMsToquNJ0shJR6ZkNgQLvQXLM9UKjqit
zAmxMCmsRUeWJTmFsjkERhMI6arGqo2WGdBqKQ8Yo
```

Step 3:
Now define the schema sanity dir(src/sanity/product.ts). I have include one more field for slug that will be beneficial for dynamic routing
```
export default {
    name: 'products',
    type: 'document',
    title: 'New Products',
    fields: [
      {
        name: 'name',
        type: 'string',
        title: 'Name',
        validation: (Rule: any) => Rule.required().error('Name is required'),
      },
      {
        name: 'slug',
        type: 'slug',
        title: 'Slug of your product',
        options: {
          source: 'name',
        },
      },
      {
        name: 'image',
        type: 'image',
        title: 'Image',
        options: {
          hotspot: true,
        },
        description: 'Upload an image of the product.',
      },
      {
        name: 'price',
        type: 'string',
        title: 'Price',
        validation: (Rule: any) => Rule.required().error('Price is required'),
      },
      {
        name: 'description',
        type: 'text',
        title: 'Description',
        validation: (Rule: any) =>
          Rule.max(150).warning('Keep the description under 150 characters.'),
      },
      {
        name: 'discountPercentage',
        type: 'number',
        title: 'Discount Percentage',
        validation: (Rule: any) =>
          Rule.min(0).max(100).warning('Discount must be between 0 and 100.'),
      },
      {
        name: 'isFeaturedProduct',
        type: 'boolean',
        title: 'Is Featured Product',
      },
      {
        name: 'stockLevel',
        type: 'number',
        title: 'Stock Level',
        validation: (Rule: any) => Rule.min(0).error('Stock level must be a
positive number.'),
      },
      {
        name: 'category',
        type: 'string',
        title: 'Category',
        options: {
          list: [
            { title: 'Chair', value: 'Chair' },
            { title: 'Sofa', value: 'Sofa' },
          ],
        },
        validation: (Rule: any) => Rule.required().error('Category is
required'),
      },
    ],
  };
```

Now import it into /sanity/schemaTypes/index.ts

```
import { type SchemaTypeDefinition } from 'sanity';
import { blogSchema } from '../blog';
import products from '../products';
import faq from '../faq';


export const schema: { types: SchemaTypeDefinition[] } = {
  types: [
    products,
  ],
}
```

Step 4:

Make a scripts directory in the src folder and then create a file named import-data.mjs (*src/scripts/import-data.mjs*). Following imports are necessary to run the scripts and import products in Sanity

- createClient from sanity/client
- axios
- dotnev
- fileUrlToPath from url
- Path

You can clearly see the script:

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

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

const client = createClient({
  projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
  dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
  token: process.env.SANITY_API_TOKEN,
  apiVersion: '2025-01-15',
  useCdn: false,
});

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('Fetching Product Data From API ...');

    const response = await
axios.get("https://next-ecommerce-template-4.vercel.app/api/product")
    const products = response.data.products;

    for (const item of products) {
      console.log(`Processing Item: ${item.name}`);

      let imageRef = null;
      if (item.imagePath) {
        imageRef = await uploadImageToSanity(item.imagePath);
      }

      const sanityItem = {
        _type: 'products',
        name: item.name,
        category: item.category || null,
        price: item.price,
        description: item.description || '',
        discountPercentage: item.discountPercentage || 0,
        stockLevel: item.stockLevel || 0,
        isFeaturedProduct: item.isFeaturedProduct,
        image: imageRef
          ? {
              _type: 'image',
              asset: {
                _type: 'reference',
                _ref: imageRef,
              },
            }
          : undefined,
      };

      console.log(`Uploading ${sanityItem.category} - ${sanityItem.name} to
Sanity !`);
      const result = await client.create(sanityItem);
      console.log(`Uploaded Successfully: ${result._id}`);

console.log("----------------------------------------------------------")
      console.log("\n\n")
    }

    console.log('Data Import Completed Successfully !');
  } catch (error) {
    console.error('Error Importing Data : ', error);
  }
}
```

```
importData();
```

Change the `sanityItem` variable according to your schema.

Now just navigate to `package.json`.

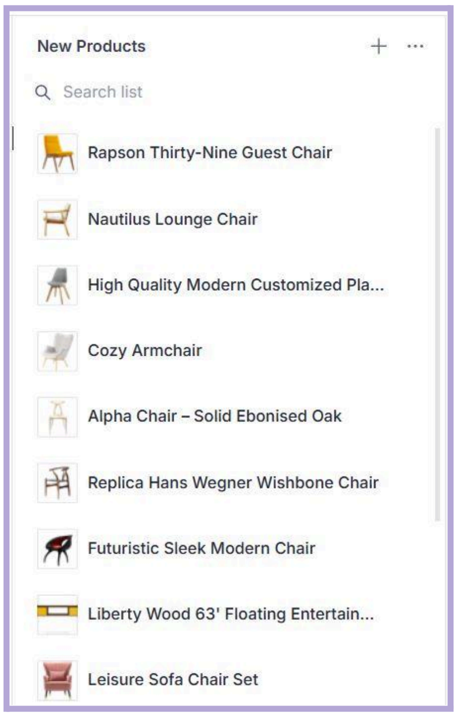
```
  "scripts": {
    "dev": "next dev",
    "build": "next build",
    "start": "next start",
    "lint": "next lint",
    "import-data": "node src/scripts/import-data.mjs"

  },
```

Add this line into your package file. Now in the terminal write

```
npm run import-data
```

The products will be added to the sanity

For Example:



New Products

Search list

Rapson Thirty-Nine Guest Chair

Nautilus Lounge Chair

High Quality Modern Customized Pla...

Cozy Armchair

Alpha Chair – Solid Ebonised Oak

Replica Hans Wegner Wishbone Chair

Futuristic Sleek Modern Chair

Liberty Wood 63' Floating Entertain...

Leisure Sofa Chair Set

For the testing purpose, navigate to Vision tab and write the query for Featured Products

```
*[_type == 'products' && isFeaturedProduct == true]{
    _id,
    name,
    "image": image.asset->url,
    description,
    price,
    category,
    "slug": slug.current
}
```

The estimated output would be like this:

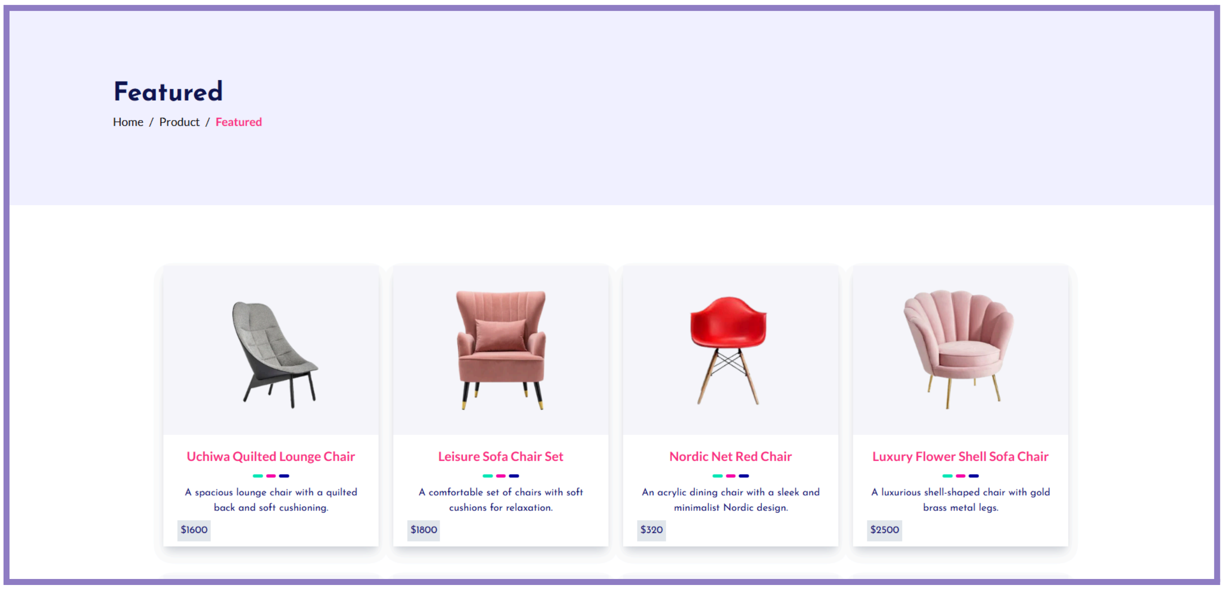
```
  {
    description: 'A modern chair with a carbon fiber frame and bold
red accents.',
    price: '2000',
    category: 'Chair',
    slug: 'futuristic-sleek-modern-chair',
    _id: 'VwUNdavwrygYi6aJzKjpdJ',
    name: 'Futuristic Sleek Modern Chair',
    image:
'https://cdn.sanity.io/images/8brhlsnp/production/cc09d3ce34a323e7d6
0432a366145277d5bb3c4f-606x528.png'
  }
]
```

Fetch all the data (products) according to the query by the help of client in sanity

```
const query = `
   *[_type == 'products' && isFeaturedProduct == true]{
     _id,
     name,
     "image": image.asset->url,
     description,
     price,
     category,
     "slug": slug.current
   }
```

```
 `;

  let data = await client.fetch(query);
  console.log(data)
```

Lastly, map all the products in frontend. All the output would be like this:



Featured

Home / Product / Featured

Uchiwa Quilted Lounge Chair

A spacious lounge chair with a quilted back and soft cushioning.

$1600

Leisure Sofa Chair Set

A comfortable set of chairs with soft cushions for relaxation.

$1800

Nordic Net Red Chair

An acrylic dining chair with a sleek and minimalist Nordic design.

$320

Luxury Flower Shell Sofa Chair

A luxurious shell-shaped chair with gold brass metal legs.

$2500

So Alhamdulillah, our migration to displaying products in frontend has been done. If you want to add something else, clone this repo. Contributions are welcome 🌟

https://github.com/marjan-ahmed/Marketplace_Technical_Foundation-Hekto/tree/main