

# DanyFur Dream

## Online Furniture Market Business

### Introduction

This document outlines the steps and process used to set up an e-commerce furniture website using **Next.js** and **Sanity**. The process includes schema creation, API integration, data fetching into Sanity, and displaying the products on the shop page of the website.

### Step 1: Project Setup

1. Install and configure **Sanity** in the project.
2. Set up the basic structure of the e-commerce website using **Next.js**.
3. Prepare the project environment to work with **Sanity** as the backend for product data.

```
? Create a new project or select an existing one Create new project
? Your project name: marketplace
Your content will be stored in a dataset that can be public or private, depending on
whether you want to query your content with or without authentication.
The default dataset configuration has a public dataset named "production".
? Use the default dataset configuration? Yes
? Creating dataset
? Would you like to add configuration files for a Sanity project in this Next.js folder? Yes
? Do you want to use TypeScript? Yes
? Would you like an embedded Sanity Studio? Yes
? What route do you want to use for the Studio? /studio
? Select project template to use Clean project with no predefined schema types
? Would you like to add the project ID and dataset to your .env.local file? Yes
Added http://localhost:3000 to CORS origins
Running 'npm install --legacy-peer-deps --save @sanity/vision@3 sanity@3 @sanity/image-url@1 styled-components@6'
npm warn deprecated @sanity/block-tools@3.70.0: Renamed - use '@portabletext/block-tools' instead. '@sanity/block-tools'
will no longer receive updates.

added 843 packages, changed 4 packages, and audited 1363 packages in 4m

265 packages are looking for funding
  run `npm fund` for details

1 moderate severity vulnerability

To address all issues, run:
  npm audit fix --force
```

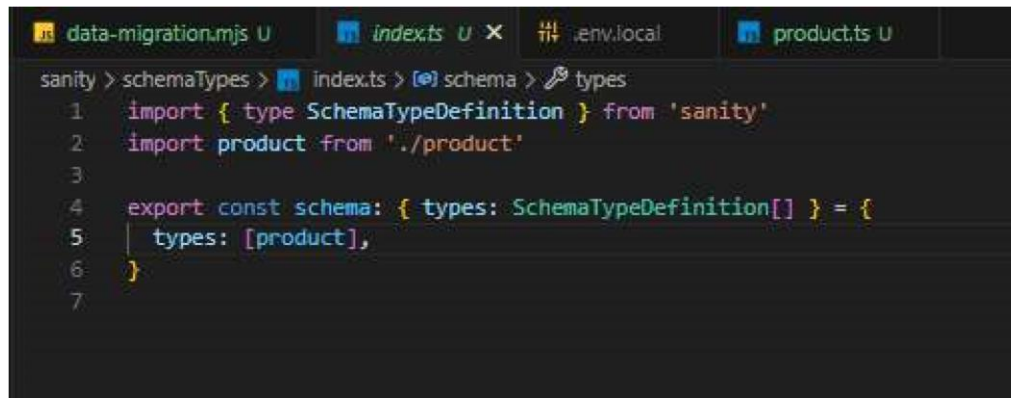
## Step 2: Schema Definition

1. Create a product schema for the e-commerce website in **Sanity**.
2. The schema should define the structure of the product data such as title, description, price, image, etc.
3. The schema is defined in a file named **product.ts**.
4. This schema allows for easy management and updating of product information.

```
sanity > schemaTypes > product.ts > productSchema > fields
1  const productSchema = {
2    name: 'product',
3    title: 'Product',
4    type: 'document',
5    fields: [
6      {
7        name: 'id',
8        title: 'ID',
9        type: 'string',
10      },
11    ],
12    {
13      name: 'name',
14      title: 'Name',
15      type: 'string',
16    },
17    {
18      name: 'image',
19      title: 'Image',
20      type: 'image',
21    },
22    {
23      name: 'imagePath',
24      title: 'Image Path',
25      type: 'url',
26    },
27    {
28      name: 'price',
29      title: 'Price',
30      type: 'number',
31    },
32    {
33      name: 'description',
34      title: 'Description',
35      type: 'text',
36    },
37    {
38      name: 'discountPercentage',
39      title: 'Discount Percentage',
40      type: 'number',
41    },
42    {
43      name: 'isFeaturedProduct',
44      title: 'Is Featured Product',
45      type: 'boolean',
46    },
47    {
48      name: 'stockLevel',
49      title: 'Stock Level',
50      type: 'number',
51    },
52    {
53      name: 'category',
54      title: 'Category',
55      type: 'string',
56    },
57  ],
58 };
59 export default productSchema;
```

## Step 3: Importing the schema

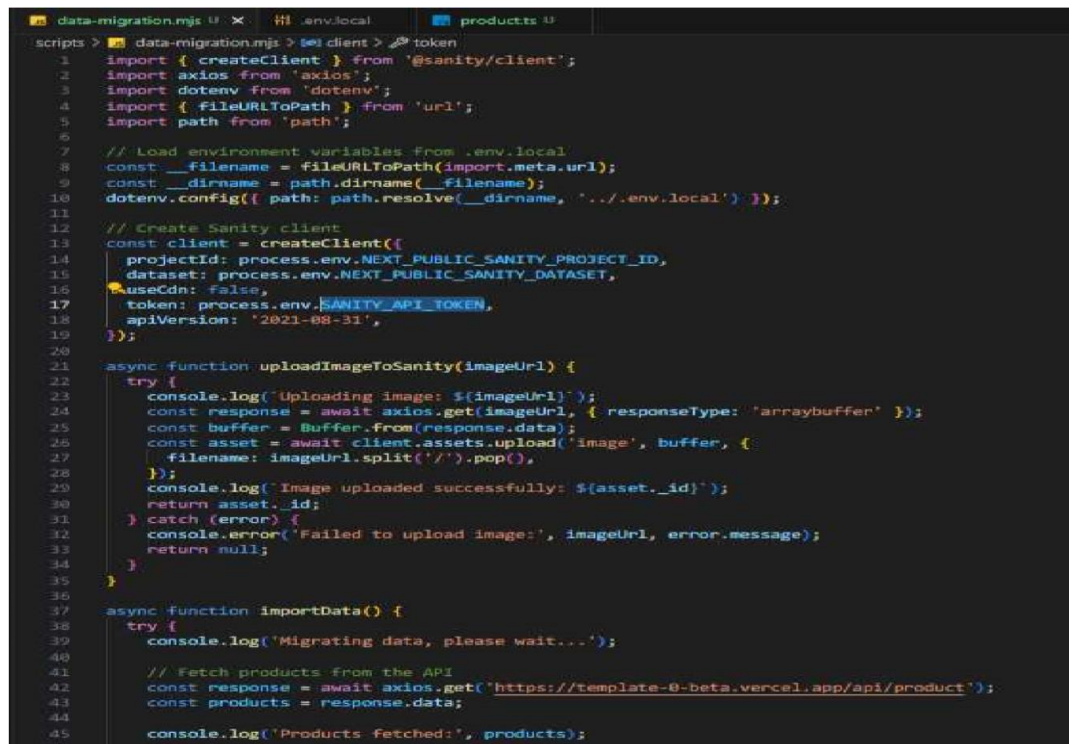
1. Import the product schema into the main application file **index.ts**.
2. Ensure that the project can interact with the **Sanity** backend by correctly importing the schema.



```
sanity > schemaTypes > index.ts > schema > types
1  import { type SchemaTypeDefinition } from 'sanity'
2  import product from './product'
3
4  export const schema: { types: SchemaTypeDefinition[] } = {
5    types: [product],
6  }
7
```

## Step 4: Data fetching script

1. Create a **data-migration.mjs** script in the **scripts/** folder.
2. The script will fetch product data from an external API and migrate it to **Sanity**.
3. This script helps to automate the process of getting product data into the **Sanity** backend.



```
scripts > data-migration.mjs > client > token
1  import { createClient } from '@sanity/client';
2  import axios from 'axios';
3  import dotenv from 'dotenv';
4  import { fileURLToPath } from 'url';
5  import path from 'path';
6
7  // Load environment variables from .env.local
8  const __filename = fileURLToPath(import.meta.url);
9  const __dirname = path.dirname(__filename);
10  dotenv.config({ path: path.resolve(__dirname, '../.env.local') });
11
12  // Create Sanity client
13  const client = createClient({
14    projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
15    dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
16    useCdn: false,
17    token: process.env.SANITY_API_TOKEN,
18    apiVersion: '2021-08-31',
19  });
20
21  async function uploadImageToSanity(imageUrl) {
22    try {
23      console.log('Uploading image: ', imageUrl);
24      const response = await axios.get(imageUrl, { responseType: 'arraybuffer' });
25      const buffer = Buffer.from(response.data);
26      const asset = await client.assets.upload('image', buffer, {
27        filename: imageUrl.split('/').pop(),
28      });
29      console.log('Image uploaded successfully: ', asset._id);
30      return asset._id;
31    } catch (error) {
32      console.error('Failed to upload image: ', imageUrl, error.message);
33      return null;
34    }
35  }
36
37  async function importData() {
38    try {
39      console.log('Migrating data, please wait...');
40
41      // Fetch products from the API
42      const response = await axios.get('https://template-0-beta.vercel.app/api/product');
43      const products = response.data;
44
45      console.log('Products fetched: ', products);
46    }
47  }
```

```

47:   for (const product of products) {
48:     let imageRef = null;
49:
50:     if (product.imagePath) {
51:       imageRef = await uploadImageToSanity(product.imagePath);
52:     }
53:
54:     const sanityProduct = {
55:       _type: 'product',
56:       _id: product.id,
57:       name: product.name,
58:       category: product.category,
59:       description: product.description,
60:       discountPercentage: product.discountPercentage,
61:       isFeaturedProduct: product.isFeaturedProduct,
62:       stockLevel: product.stockLevel,
63:       price: parseFloat(product.price),
64:       image: imageRef
65:       ? {
66:         _type: 'image',
67:         asset: {
68:           _type: 'reference',
69:           _ref: imageRef,
70:         },
71:       }
72:       : undefined,
73:       imagePath: product.imagePath, // Store original image URL
74:     };
75:
76:     await client.create(sanityProduct);
77:     console.log('Product created in Sanity: ${sanityProduct.id}');
78:   }
79:
80:   console.log('Data migrated successfully!');
81: } catch (error) {
82:   console.error('Error in migrating data:', error.message);
83: }
84: }
85:
86: importData();

```

## Step 5: Package Configuration

1. Modify the **package.json** file to add a custom script for running the data migration script.
2. The script is added under the **"scripts"** section:

```

private: true
  > Debug
  "scripts": {
    "dev": "next dev",
    "build": "next build",
    "start": "next start",
    "lint": "next lint",
    "Data": "node scripts/data-migration.mjs"
  },

```

3. This allows you to run the data migration with the command `npm run Data`.

## Step 6: Running the data migration command

1. Run the following command in the terminal to trigger the data migration:

**npm run Data**

2. The data is fetched from the API and migrated into **Sanity**, making it available in the backend.

## Step 7: Displaying products on the shop page

1. The products that were added to **Sanity** are now displayed dynamically on the **shop page** of the website.
2. The product data is fetched from **Sanity** and shown on the front end using **Next.js**.
3. This integration ensures that the shop page is automatically updated whenever the product data is modified in **Sanity**.





Chair Wibe

\$1200



Alpha Table

\$900



Replica Table

\$750



Sleek Modern Table

\$2000



Liberty Center

\$1100



Leisure Sofa Chair Set

\$1800



Diondre Chair

\$720



Matilda Velvet Bed

\$600