

E-commerce Web Application with Next.js and Sanity CMS

This project is a modern e-commerce web application built with Next.js and Sanity CMS, offering a seamless shopping experience with dynamic product management.

Repository Structure

The repository is organized as follows:

- `src/` : Contains the main source code for the application
 - `app/` : Next.js app directory with route handlers and page components
 - `components/` : Reusable React components
 - `lib/` : Utility functions and shared code
 - `sanity/` : Sanity CMS configuration and schema definitions

Usage Instructions

Installation

Prerequisites:

- Node.js (v14 or later)
- npm (v6 or later)

To install the project dependencies, run:

```
npm install
```

Getting Started

1. Create a `.env.local` file in the root directory with the following variables:

```
NEXT_PUBLIC_SANITY_PROJECT_ID=your_sanity_project_id
NEXT_PUBLIC_SANITY_DATASET=your_sanity_dataset
NEXT_PUBLIC_SANITY_API_VERSION=your_sanity_api_version
```

Replace the values with your Sanity project details.

2. To start the development server, run:

```
npm run dev
```

This will start the Next.js development server and the Sanity Studio. You can access the application at `http://localhost:3000` and the Sanity Studio at `http://localhost:3000/studio`.

API Routes and Data Fetching

The application includes several API routes for fetching product data:

- `/api/new-arrival` : Fetches the latest 4 products
- `/api/product-detail` : Retrieves details for a specific product
- `/api/products` : Fetches all products
- `/api/top-selling-products` : Fetches the top 4 selling products

To fetch data from these API routes and display it on the UI, you can use the `axios` library or the built-in `fetch` function. Here's an example of how to fetch and display products:

```

import axios from 'axios';
import { useEffect, useState } from 'react';

const ProductList = () => {
  const [products, setProducts] = useState([]);

  useEffect(() => {
    const fetchProducts = async () => {
      try {
        const response = await axios.get('/api/products');
        setProducts(response.data.data);
      } catch (error) {
        console.error('Error fetching products:', error);
      }
    };

    fetchProducts();
  }, []);

  return (
    <div>
      <h2>Product List</h2>
      <ul>
        {products.map((product) => (
          <li key={product._id}>{product.name} - ${product.price}</li>
        ))}
      </ul>
    </div>
  );
};

export default ProductList;

```

This example demonstrates how to fetch products from the `/api/products` route and display them in a list. You can use a similar approach for other API routes, adjusting the endpoint and data structure as needed.

Troubleshooting

1. Issue: Images not loading
 - Verify that the Sanity project ID and dataset are correct in the `.env.local` file
 - Ensure that the images are properly uploaded to Sanity
2. Issue: Sanity Studio not loading
 - Make sure you're running the development server with `npm run dev`
 - Check if the Sanity configuration in `sanity.config.ts` is correct
 - Verify that you have the necessary permissions for your Sanity project

For debugging, you can enable verbose logging in Next.js by setting the `DEBUG` environment variable:

```
DEBUG=* npm run dev
```