

Day 7 Documentation Page: E-commerce Website Development Journey

Website Name : Furnish & Find

Website Link : <https://furnish-find-v2-by-arsalan-ahmed.vercel.app/>

Student Information:

- **Name:** Arsalan Ahmed
 - **Roll Number:** 222623
 - **Email:** arsalan23c@gmail.com
 - **Slot:** Monday, 2 PM to 5 PM
-

Introduction

This documentation covers my journey of developing a live e-commerce website using **Sanity**, **Next.js**, and **TypeScript**. This project encapsulates the integration of a content management system (Sanity), the power of server-side rendering (Next.js), and the advantages of strong typing (TypeScript).

Through this project, I have applied concepts from Quarter 1 (focused on TypeScript) and Quarter 2 (focused on Next.js) in a practical, real-world context. This journey also highlights my learning curve and provides insights into how these technologies interconnect and create a seamless experience for both developers and users.

Project Overview

The primary objective of the project was to build an e-commerce website capable of dynamic content fetching, product display, search functionalities, and a shopping cart system. The website uses **Sanity** as a headless CMS, which makes the content management seamless. **Next.js** is used to ensure server-side rendering for better performance and SEO. **TypeScript** is leveraged to provide static typing, minimizing errors and ensuring code quality.

Technologies Used

1. **Sanity CMS**

Sanity is a flexible and real-time content management system that allows dynamic content management. I integrated Sanity to manage product data and ensure easy updates to the website without any backend coding.

2. **Next.js**

Next.js was the framework of choice due to its capabilities for building dynamic web pages with server-side rendering (SSR). It allows us to create SEO-friendly pages and ensures fast loading times.

3. **TypeScript**

TypeScript played a significant role in ensuring that the code is statically typed, making it easier to detect errors and improve code maintainability. This technology was particularly valuable during the development phases where types were essential for consistent API interactions and overall structure.

Quarter 1: TypeScript

In Quarter 1, we focused on mastering **TypeScript**, which I applied extensively in this project to define strict types for various objects, functions, and API calls. This helped to eliminate runtime errors, provided better code completion in IDEs, and made debugging simpler and more efficient. My focus on **TypeScript** was instrumental in ensuring that the codebase remained clean and scalable.

Quarter 2: Next.js

Quarter 2 provided an in-depth understanding of **Next.js** and its powerful features such as **server-side rendering** and **static site generation**. In this project, I utilized Next.js to build dynamic product pages, optimize SEO, and enhance the overall user experience. The routing system in Next.js made it easier to navigate through different product categories and pages, while its automatic code splitting and optimization features ensured fast load times.

Development Process

1. **Sanity Integration**

The integration of **Sanity** with Next.js involved using Sanity's API to fetch product data, ensuring that the content could be updated in real-time without requiring a redeployment of the website.

2. **Next.js Dynamic Pages**

I leveraged Next.js's **getServerSideProps** and **getStaticProps** methods for fetching product details and ensuring that the pages load fast and are optimized for search engines.

3. **TypeScript for Static Typing**

I utilized **TypeScript** to ensure that API responses, product data, and UI components adhered to strict type definitions, thus preventing errors in data handling and improving overall code quality.

Testing and Bug Fixing

During the development, I ran several test cases to ensure the functionality of core features like product display, cart functionality, and search bar. Some issues were faced, such as incorrect API endpoint configurations and problems with the search bar, but with effective debugging and problem-solving, these issues were resolved.

Acknowledgments

I would like to extend my sincere thanks to the following individuals for their unwavering support, guidance, and encouragement throughout this journey:

- **Sir Zia Khan**
- **Sir Ameen Alam**
- **Sir Asharib Ali**
- **Sir Daniyal Nagori**
- **All the Teaching Faculty**

Their teaching and feedback helped me navigate through the complexities of these technologies and inspired me to always push the boundaries of my learning.

Conclusion

Building this e-commerce website has been an enriching experience where I applied all the skills learned during the course. The integration of **Sanity**, **Next.js**, and **TypeScript** has enhanced my understanding of full-stack web development, especially in creating scalable, maintainable, and high-performance applications.

Thank you, and I look forward to applying these learnings in future projects.

Regards,
Arsalan Ahmed