Q-Commerce Website Development - Day 7 Documentation

Table of Contents

1. Introduction

The Q-Commerce website is designed to revolutionize quick commerce by providing users with an efficient platform to browse, order, and receive products swiftly. The project focuses on user experience, seamless API integration, and modern web development practices.

2. Objectives

- **Efficiency: Enable fast product browsing and checkout processes.**
- **User Experience: Deliver a responsive and intuitive interface.**
- **Scalability: Ensure the system can handle high traffic and a growing user base.**
- **Integration: Utilize APIs and CMS for dynamic content management.**

3. Features

- **Product Listing: Display a wide range of products with categories.**
- **Dynamic Pages: Individual product pages with detailed descriptions.**
- **Shopping Cart: Add, remove, and manage products in the cart.**
- **Checkout Process: Streamlined payment and order placement.**
- **Admin Panel: Manage products, orders, and user data through Sanity CMS.**

4. Technical Requirements

Frontend

- **Framework: Next.js 15**
- **Styling: Tailwind CSS**

- **Animations: AOS Library**

Backend

- **CMS: Sanity.io**
- **Database: MongoDB (optional for storing additional user data)**
- **API: RESTful APIs for product and order management**

Tools

- **Version Control: Git & GitHub**
- **IDE: VS Code**
- **Browser: Microsoft Edge**

5. System Architecture

Overview

- **Client: User-facing website (Next.js)**
- **Server: API layer for handling requests and data updates**
- **CMS: Sanity.io for managing product and content data**
- **Database: Optional database for user-specific information**

6. API Design

Product APIs

- **GET /products: Fetch all products**
- **GET /products/:id: Fetch product details**
- **POST /cart: Add product to cart**
- **DELETE /cart/:id: Remove product from cart**

Order APIs

- **POST /order: Place a new order**
- **GET /order/:id: Fetch order details**

7. Sanity CMS Schema

Food Schema

- **Fields:**
  - **Name (string)**
  - **Category (string)**
  - **Price (number)**
  - **Original Price (number)**
  - **Tags (array of strings)**
  - **Image (image)**

- Description (text)
- Availability Status (boolean)

8. Implementation Plan

Day 1

- Define project scope and objectives
- Research and gather requirements

Day 2

- Create system architecture and workflows
- Plan API and CMS schema design

Day 3

- Set up Next.js project and integrate Tailwind CSS
- Develop Sanity CMS schemas

Day 4

- Implement product listing and dynamic pages
- Add shopping cart functionality

9. Challenges and Solutions

Challenge: Ensuring Fast API Responses

- Solution: Implement caching mechanisms and optimize database queries.

Challenge: Managing Dynamic Content

- Solution: Use Sanity CMS for flexible and efficient content updates.

10. Future Enhancements

- User Authentication: Add login and registration functionality.
- Order Tracking: Enable users to track their orders in real time.
- AI Recommendations: Implement AI to suggest products based on user preferences.
- Multi-Language Support: Expand the platform to support multiple languages.

11. Conclusion

The Q-Commerce website aims to set a benchmark in the quick commerce industry by offering a robust, scalable, and user-friendly platform. With its dynamic features and future-focused enhancements, the project promises to deliver an exceptional user experience.

Umama Rajput