

Technical Planning Documentation

Overview

This document outlines the technical plan for developing an Q-Commerce Marketplace to empower small businesses and individuals by providing a platform to sell their products online. The technical planning follows the brainstorming from Hackathon Day 1 and incorporates the recommendations from the Day 2 guidelines.

Key Technologies

Frontend: Next.js

Content Management System (CMS): Sanity

Frontend Frameworks & Library : Tailwind CSS & Shadcn UI

Add To Cart: Redux Toolkit

For Forms : React hook forms

For Authentication : Clerk

Payment Gateway: Stripe

Order Tracking and Shipment: ShipEngine

Testing : Cypress

Hosting and Deployment: Vercel

Code Repository: Github

Technical Planning Documentation

Technical Architecture

System Overview

❑ Frontend Development

➤ Framework: Next.js

- Chosen for its **server-side rendering (SSR)** and **static site generation (SSG)** capabilities, which improve page load speed and SEO.
- Next.js simplifies routing and enhances performance, making it suitable for e-commerce platforms requiring fast responses.

❑ Content Management System (CMS)

➤ Sanity

- Sanity is a highly flexible CMS that enables developers to create and manage structured content.
- Provides real-time collaborative editing and content versioning, ensuring seamless management of product details, blogs, and other dynamic content.

❑ Frontend Frameworks & Libraries

➤ Tailwind CSS

- A utility-first CSS framework that ensures rapid styling of UI components.
- Promotes a clean, responsive, and customizable design system.

➤ Shadcn UI

- A collection of pre-designed components integrated with Tailwind CSS.
- Helps build modern, aesthetically pleasing UIs with minimal effort.

Technical Planning Documentation

Technical Architecture

System Overview

❑ Add to Cart Functionality

➤ Redux Toolkit

- Centralized state management for managing the cart efficiently.
- Facilitates scalability by providing tools for handling global states like product selection, quantities, and user preferences.

❑ Form Management

➤ React Hook Forms

- Simplifies form handling by managing input fields, validations, and submission processes efficiently.
- Lightweight and optimized for better performance in handling complex forms, like user registration, login, and checkout.

❑ Authentication

➤ Clerk

- Provides pre-built authentication solutions, such as sign-up, login, and user profile management.
- Offers secure, scalable, and customizable authentication flows.

➤ Payment Gateway

• Stripe

- Reliable and secure payment integration that supports multiple payment methods, currencies, and subscriptions.
- Simplifies implementation with SDKs and APIs, ensuring seamless payment processing.

Technical Planning Documentation

Technical Architecture

System Overview

❑ Order Tracking and Shipment

➤ ShipEngine

- Handles order tracking and shipment processes.
- Integrates multiple carriers, enabling users to track orders in real-time.
- Supports rate comparisons, shipping labels, and delivery updates.

❑ Testing

➤ Cypress

- Ensures the application works as expected by providing end-to-end testing.
- Verifies critical workflows like product search, adding to the cart, checkout, and payments.
- Reduces bugs and ensures a smooth user experience.

❑ Hosting & Deployment

➤ Vercel

- Optimized for deploying Next.js applications.
- Features auto-scaling, serverless functions, and a global CDN, ensuring fast delivery of assets and minimal downtime.
- Simplifies continuous integration (CI) and deployment workflows.

❑ Code Repository

➤ GitHub

- Used for source code management and version control.
- Provides a collaborative platform for developers to track changes, review code, and manage branches.

Technical Workflow

❖ Frontend and CMS Integration:

- Next.js communicates with Sanity via APIs to fetch and display dynamic content like product listings, categories, and blogs.
- Tailwind CSS and Shadcn UI ensure the frontend is modern and responsive.

❖ E-commerce Features:

- Redux Toolkit stores cart data centrally, enabling users to add/remove products seamlessly.
- React Hook Forms manages forms like registration, login, and checkout efficiently.

❖ User Authentication and Security:

- Clerk ensures secure user authentication with customizable sign-up/login flows.

❖ Payment and Order Fulfillment:

- Stripe handles secure payment processing.
- ShipEngine integrates with carriers to track orders and provide shipment updates.

❖ Testing and Deployment:

- Cypress tests workflows to ensure bug-free functionality.
- Vercel hosts the platform for high availability and speed.

❖ Collaboration and Code Management:

- GitHub maintains the codebase, tracks changes, and enables collaboration among developers.