

Hackathon Day 2**Technical Plan for Q-Commerce Marketplace****Introduction:**

Our Q-commerce platform focuses on rapid delivery of fresh and high-quality products within 30 minutes. This document outlines the technical architecture, workflows, and API requirements to build a scalable, user-friendly, and efficient Q-commerce marketplace.

1. System Architecture Overview**Tech Stack:**

- **Next.js:** for server-side rendering
- **Tailwind:** for styling and responsiveness.
- **ShadCN UI:** UI components library
- **React Icon:** for vector & icons

Frontend (UI)

- **Responsive Design:** For mobile, tablet, and desktop devices.
- **Pages:**
 1. Home Page
 2. Menu
 3. About
 4. FAQ Page
 5. Product Listing
 6. Product Details
 7. Add Cart
 8. Blog Page
 9. Login Page
 10. Signup Page
 11. Order Confirmation

Backend

- Sanity for managing content like products, categories, and banners.
- API Layer: Sanity GROQ queries and API endpoints in Next.js to fetch and manage data

Hackathon Day 2

Technical Plan for Q-Commerce Marketplace**API Requirements**

API End Points	Method	Purpose
/products	Get	Fetch Product list
/order	Post	Create new order
/delivery Zone	Fetch	Delivery zone details.

2. Workflows**1. Product Management**

- Admin adds/edit products via the Sanity dashboard.
- The product information (name, price, availability, etc.) syncs to the frontend via GROQ queries

2. Key Components

- **Front-End(Next.js):**
A responsive and interactive interface where User browse and place orders.
- **Back-End(Sanity CMS):**
Handle products data, order management and website content updates.
- **Third Party APIs:**
Used for shipment tracking, payments and quick delivery integration.

3. Product Browsing

- User navigates through product categories.
- Product data fetched from Sanity CMS

4. Real-Time Delivery Tracking

- Order shipment data is fetched through the third party and display on the frontend.

Hackathon Day 2

Technical Plan for Q-Commerce Marketplace