Submission Title:
"Marketplace Technical Foundation - [Rapid Cart]"

This document outlines the user journey for a Q-Commerce marketplace, focusing on key steps and technical solutions to create a reliable, user-friendly platform that's intuitive and enjoyable for all users.

User Flow

User Opens the Application:

User opens the app and sign in or continue as a guest.

Homepage:

User views homepage with featured products and categories..
Option to search for specific items, browse categories, or filter by preferences (price, ratings, etc.).

Search or Browse:

User uses the search bar to find a product or browses by category.
Search results are displayed with relevant filters (e.g., price range, ratings, or brand).

Product Details Page:

User selects a product to view detailed information: images, description, reviews, price, and availability.
Option to choose quantity, color, or size (if applicable).
Option to add product to cart or save for later.

Add to Cart:

User adds product to the shopping cart.
Option to continue shopping or proceed to checkout.

View Cart:

User reviews items in the cart, adjusts quantities, or removes items.
Option to view total price including taxes and delivery.

Checkout:

User proceeds to checkout where they provide shipping details.
Option to sign in for faster checkout or continue as a guest.
Users enter payment details (credit/debit, digital wallets, etc.).

Payment Confirmation:

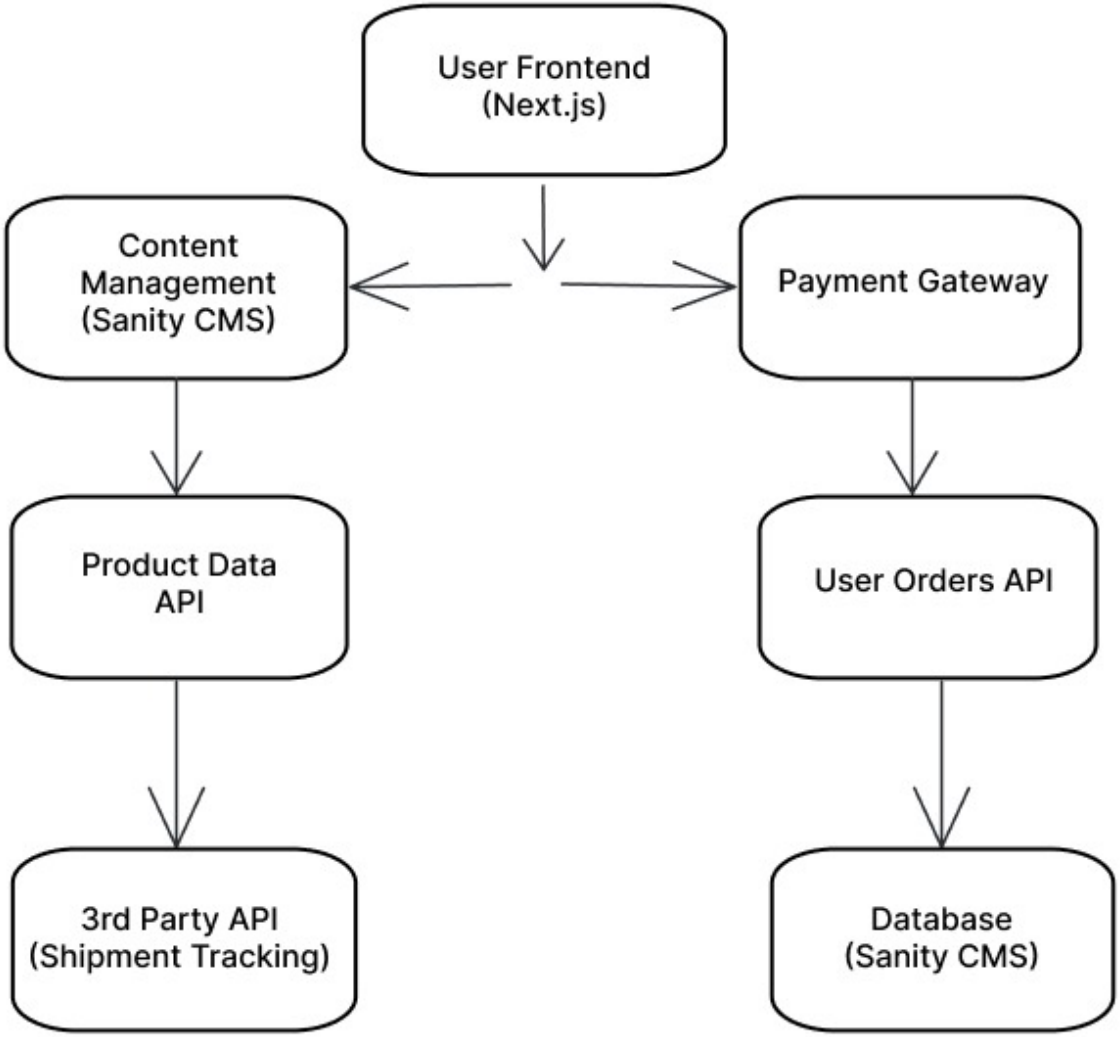
User reviews the order summary (shipping address, payment method, and order details).
Option to confirm the order.

Order Confirmation & Tracking:

User receives an order confirmation screen with an estimated delivery date.
Option to track the order in real time.

Delivery and Feedback:

User receives a notification when the order is out for delivery.
Once delivered, the user is prompted to leave a review or rate the product.


User Frontend
(Next.js)
Content
Management
(Sanity CMS)
Payment Gateway
Product Data
API
User Orders API
3rd Party API
(Shipment Tracking)
Database
(Sanity CMS)

Component Overview:

- **Frontend (Next.js):** Provides the user interface, handling interactions like product browsing, cart management, and order placement.
- **Sanity CMS:** Manages all the content, product details, user information, and order data.
- **Product Data API:** Facilitates communication between the frontend and Sanity CMS for dynamic product listing retrieval.
- **Payment Gateway:** Handles secure payment processing.
- **Shipment Tracking API:** Tracks the real-time status of orders using third-party shipment tracking services.
- **Database (Sanity CMS):** Stores all content and transactional data like products, orders, users, and payment status.

Key Workflows

User-Registration Workflow:

Action: User signs up for an account.

Data Flow: User submits registration details.
Data is stored in Sanity CMS.
Confirmation email is sent to the user.

Product Browsing Workflow:

Action: User views product categories.

Data Flow: Frontend requests product data from the Product Data API.
The API fetches data from Sanity CMS (product name, price, description, image).
Product details are displayed dynamically to the user..

Order Placement Workflow:

Action: User adds items to the cart and proceeds to checkout.

Data Flow: User's order details are sent to the User Orders API and stored in Sanity CMS.
The Payment Gateway processes the payment.
Upon success, order confirmation is sent to the user.

Shipment Tracking Workflow::

Action: User tracks the status of their order.

Data Flow: Frontend makes a request to the Shipment Tracking API.
The API returns the real-time status (shipped, in transit, delivered).
Shipment details are displayed to the user in the frontend.

API Requirements

Endpoint Definitions:

Frame 1

Frame 2 Endpoint	Method	Purpose	Response Example
/products	GET	Fetch all product details from Sanity CMS.	{ "id": 1, "name": "Product A", "price": 100, "stock": 50, "image": "image_url" }
/orders	POST	Create a new order in Sanity CMS.	{ "customer": { "name": "John Doe", "address": "123 St" }, "orderItems": [{"productId": 1, "quantity": 2}], "totalAmount": 200, "status": "pending" }
/shipment	GET	Track order status using a 3rd-party shipment API.	{ "orderId": 123, "status": "In Transit", "expectedDelivery": "2025-01-20" }
/payment	POST	Process the payment for an order.	{ "orderId": 123, "paymentStatus": "Success", "transactionId": "abc123" }

Frame 6

Frame 7

Frame 8

Sanity Schema for Products

```
export default {
  name: 'product',
  type: 'document',
  fields: [
    { name: 'name', type: 'string', title: 'Product Name' },
    { name: 'price', type: 'number', title: 'Price' },
    { name: 'stock', type: 'number', title: 'Stock Level' },
    { name: 'image', type: 'image', title: 'Product Image' },
  ],
};
```

Sanity Schema for Orders

```
export default {
  name: 'order',
  type: 'document',
  fields: [
    { name: 'customer', type: 'reference', to: [{ type: 'customer' }] },
    { name: 'products', type: 'array', of: [{ type: 'reference', to: [{ type: 'product' }] }] },
    { name: 'paymentStatus', type: 'string', title: 'Payment Status' },
    { name: 'orderDate', type: 'datetime', title: 'Order Date' },
    { name: 'shipmentStatus', type: 'string', title: 'Shipment Status' },
  ],
};
```