Final Plan for Q-commerce FoodTuck Restaurant

---

1. Technical Foundation Overview

The Q-commerce FoodTuck Restaurant aims to provide a fast and efficient food ordering and delivery experience. The technical foundation will support this mission by ensuring quick data retrieval, seamless integration with third-party services, and a user-friendly interface.

---

2. Frontend Requirements

- **User-Friendly Interface:** Designed to provide an intuitive experience for browsing and ordering food.
- **Responsive Design:** Optimized for both mobile and desktop devices.
- **Essential Pages:** Home, Product Listing, Product Details, Cart, Checkout, Contact and Order Confirmation.

---

3. Backend (Sanity CMS)

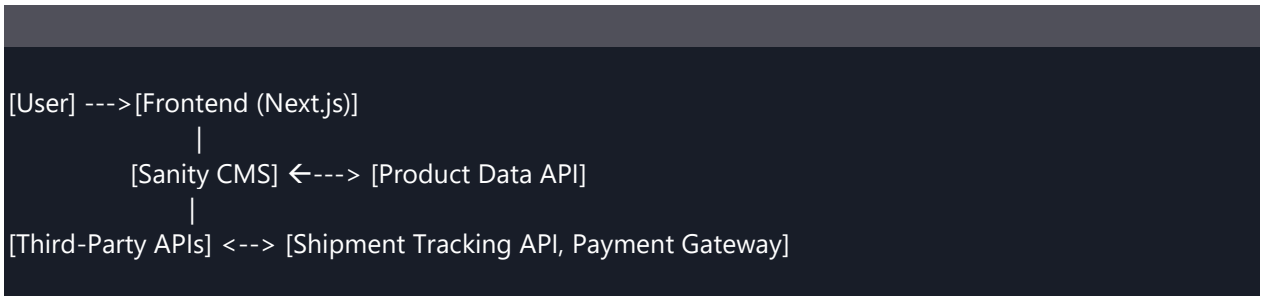
- **Content Management:** Utilize Sanity CMS to manage product data, customer details, and order records.
- **Schema Design:**
    - **Product:** Fields include Id, name, price, description, image, availability, and sell-by time.
    - **Order:** Fields include customer info, list of items, payment details, and delivery status.

---

4. Third-Party Integrations

- **Shipment Tracking:** Integrate with a quick delivery service API (e.g., Uber Eats, Delivery Boy, Food Panda).
- **Payment Gateway:** Use COD, Stripe or local options like Easypaisa/JazzCash.

5. System Architecture



---

6. Key Workflows

1. **User Registration:**
    - Sign up process with data storage in Sanity and confirmation email.
2. **Product Browsing:**
    - Fetch product data from Sanity and display on the frontend.
3. **Order Placement:**
    - Add items to cart, proceed to checkout, and save order details in Sanity.
4. **Shipment Tracking:**
    - Fetch real-time tracking data from the third-party API and display to the user.

---

7. API Requirements

- **Menu API:**
    - GET /api/menu: Fetch all menu items.
    - GET /api/menu/:id: Fetch details of a specific menu item.
- **Order API:**
    - POST /api/orders: Create a new order.
    - GET /api/orders/:id: Fetch order details.
- **Payment API:**
    - POST /api/payments: Process payment for an order.
- **Customer API:**
    - GET /api/customers/:id: Fetch customer profile and order history.

    - PUT /api/customers/:id: Update customer details.
- **Order Tracking API:**
    - GET /api/orders/:id/status: Fetch real-time order status.

8. Technical Documentation

- **System Architecture Document:** Describes the interaction between frontend, backend, and third-party services.
- **API Specification Document:** Details endpoints, methods, payloads, and responses.
- **Workflow Diagram:** Visualizes user interactions and data flows.
- **Data Schema Design:** Defines entities and relationships for Sanity CMS.
- **Technical Roadmap:** Outlines project steps, milestones, and deliverables.

Workflow Diagram for Q-commerce FoodTuck Restaurant

Swimlane Diagram

This diagram is organized into swimlanes, each representing a component involved in the workflow. The swimlanes are:

1. **User**
2. **Frontend (Next.js)**
3. **Sanity CMS**
4. **Payment Gateway**
5. **Shipment Tracking API**

Workflow Steps

1. **User Browses Menu**
    - **User** -> **Frontend**: User navigates to the menu page.
    - **Frontend** -> **Sanity CMS**: Requests menu data.
    - **Sanity CMS** -> **Frontend**: Sends menu data.
    - **Frontend** -> **User**: Displays menu items.
2. **User Adds Items to Cart and Proceeds to Checkout**
    - **User** -> **Frontend**: Adds items to cart and proceeds to checkout.
    - **Frontend** -> **Sanity CMS**: Creates new order.

- **Sanity CMS** -> **Payment Gateway**: Processes payment.
- **Payment Gateway** -> **Sanity CMS**: Confirms payment.
- **Sanity CMS** -> **Shipment Tracking API**: Initiates shipment.
- **Shipment Tracking API** -> **Sanity CMS**: Provides tracking ID.
- **Sanity CMS** -> **Frontend**: Sends order confirmation.
- **Frontend** -> **User**: Displays order confirmation.

User Views Order Tracking

- **User** -> **Frontend**: Requests order tracking.
- **Frontend** -> **Sanity CMS**: Requests order status.
- **Sanity CMS** -> **Shipment Tracking API**: Fetches tracking info.
- **Shipment Tracking API** -> **Sanity CMS**: Sends tracking status.
- **Sanity CMS** -> **Frontend**: Sends tracking status.
- **Frontend** -> **User**: Displays tracking information.

Summary

This workflow diagram provides a clear and organized representation of the interactions between the user, frontend, backend (Sanity CMS), and third-party services (Payment Gateway and Shipment Tracking API) in the Q-commerce FoodTuck Restaurant system. Each step is logically sequenced to ensure a smooth and efficient process from menu browsing to order tracking.