MARKETPLACE BUILDER HACKATHON 2025

MARKETPLACE TECHNICAL FOUNDATION PRESENTATION

INTRODUCTION:

This document outlines the technical foundation for your food business marketplace. The focus is on defining the system architecture, workflows, API requirements, and Sanity CMS schema. The goal is to ensure a seamless integration of frontend and backend components, enabling efficient and scalable operations.

1.DEFINE TECHNICAL REQUIREMENTS:

FRONTEND REQUIREMENT

- ***Responsive Design:***
  - *Ensure the marketplace works seamlessly on mobile and desktop devices.*
- ***Essential Pages:***
  - *Home*
  - *Menu*
  - *Blog*
  - *About*
  - *Shop*
  - *Products/id {dynamic page}*
  - *Sing In*
  - *Sing Up*
  - *Contact*
  - *Checkout*
  - *Order Confirmation*

BACKEND REQUIREMENTS (SANITY CMS):

- ***Use Sanity CMS for managing:***
  - *Product data*
  - *Customer details*
  - *Order records*
- ***Define schemas in Sanity to align with:***
  - *Product structure (e.g., name, price, stock)*
  - *Order structure (e.g., customer info, product details, payment status)*

THIRD-PARTY API'S

- ***Shipment Tracking API:***
    - *Fetch and display order shipment status in real-time.*
- ***Payment Gateway API:***
    - *Securely process payments and return confirmations to the system.*

2. DESIGN SYSTEM ARCHITECTURE

High-Level Architecture Diagram

```
[Frontend (Next.js)]
      |
[Sanity CMS] <-------> [Product Data API]
      |
[Third-Party APIs] ---> [Shipment Tracking API]
      |
                    ---> [Payment Gateway]
```

WORKFLOW DESCRIPTION

1. ***User Browsing Products:***
    - *User visits the frontend (Next.js).*
    - *Frontend requests product data via the Sanity CMS API.*
    - *Products are dynamically displayed to the user.*
2. ***Order Placement:***
    - *User adds products to the cart and proceeds to checkout.*
    - *Order details are sent to Sanity CMS for record-keeping.*
    - *Payment is processed through the Payment Gateway.*
3. ***Shipment Tracking:***
    - *Order status is updated using the Shipment Tracking API.*
    - *Real-time tracking information is displayed to the user.*

5. SANITY CMS SCHEMAS

Product Schema

```
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'
 }
  ]
};
```

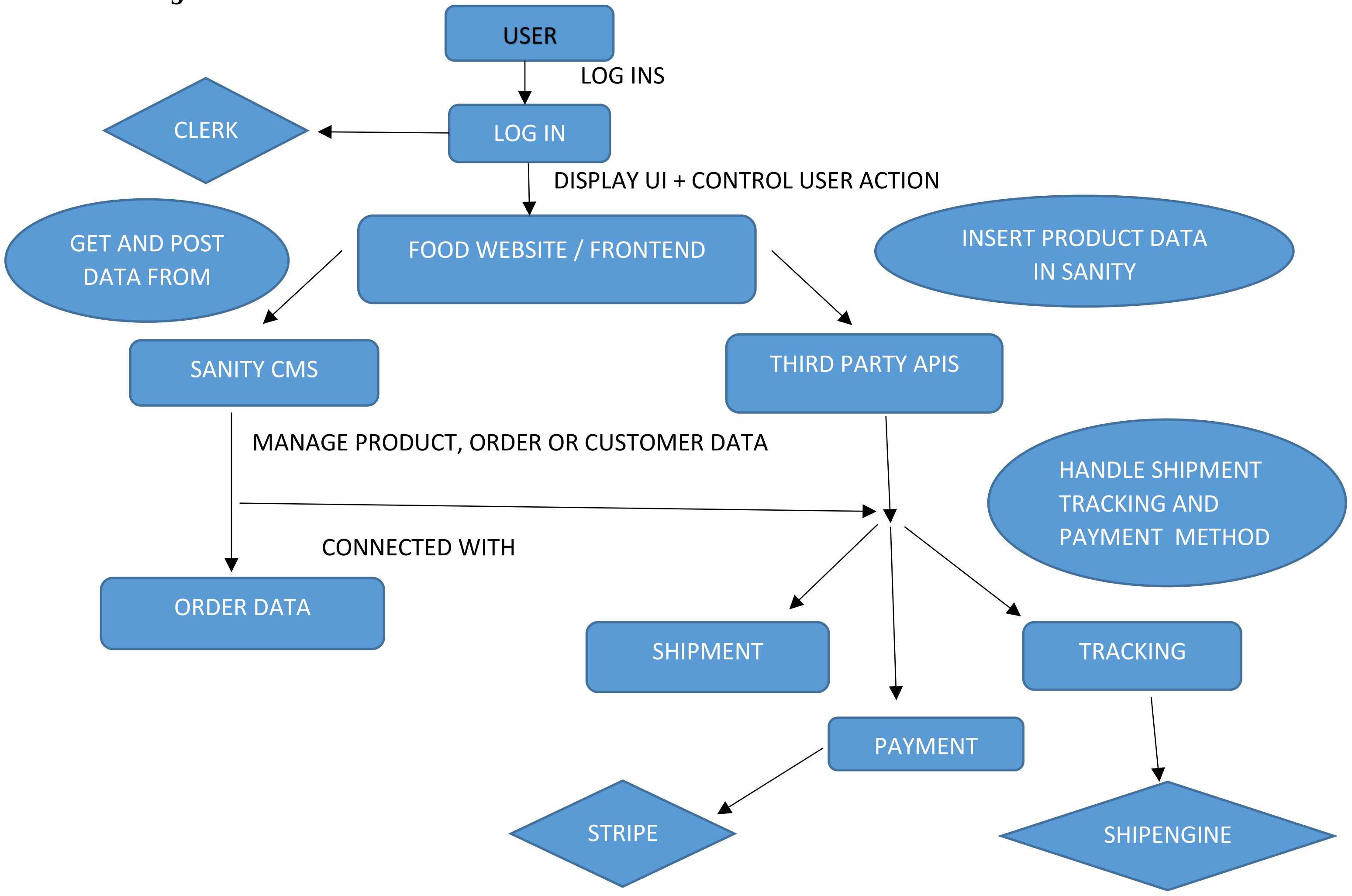
Order Schema

```
export default {
  name: 'order',
  type: 'document',
  fields: [
    { name: 'customer',
      type: 'reference',
       to: [
        { type: 'customer' }],
         title: 'Customer'
},
    { name: 'products',
      type: 'array',
      of: [
     { type: 'reference',
      to: [
          { type: 'product' }
]
 }],
 title: 'Products' },


    { name: 'totalAmount',
      type: 'number',
      title: 'Total Amount' },
    {
       name: 'status',
       type: 'string',
       title: 'Order Status' }
  ]
};
```

6. WORKFLOW DIAGRAM

Diagram


USER
LOG INS
LOG IN
CLERK
DISPLAY UI + CONTROL USER ACTION
GET AND POST DATA FROM
FOOD WEBSITE / FRONTEND
INSERT PRODUCT DATA IN SANITY
SANITY CMS
THIRD PARTY APIS
MANAGE PRODUCT, ORDER OR CUSTOMER DATA
HANDLE SHIPMENT TRACKING AND PAYMENT METHOD
CONNECTED WITH
ORDER DATA
SHIPMENT
TRACKING
PAYMENT
STRIPE
SHIPENGINE