HACKHATHON DAY 2

Transforming My Business Plan to Technical Planning.

The goal of Day 2 is to shift from business focused Planning to technical implementation. At this stage it is essential to define the system Architecture, API-requirements and backend structure to ensure Smooth market place development.

Define Technical Requirements :-

Frontend Requirements.

The frontend will delieves a Seamless user freinally expesicnce with the following Pages:

1) Home Page: Highlight food Categories and featured Products.
2) About: Provide info about our experience and about our best foody Products.
3) Contact: Display Contact information clearly for Customer inquiries.
4) Chefs: Introduce our chefs and highlight their experence and expertise in Cooking.

5) Menu: Showcase the various menu plans offered by the website.

6) Shop: Highlight the full range of Products available including food, ice creams and beverages.

7) Cart: A page where user can view they've added to their Cart, modify quantities, remove items and proceeds to checkout.

8) Sign up: A registration page for users to sign up on the website.

9) Login: A login page for returning users to access their accounts.

Back end with Sanity CMS-

Sanity CMS will serve as the back end to manage dynamic data like products, Customers and orders.

Third-Party API Integration

To enhance the functionality and user experience of the website, the following third party API's will be integrated.

1- Payment Gateway API

- Purpose : Secure payment Processing.
- Example : Stripe, PayPal.
- Features : Transaction Status tracking and accept debit/credit Cards.

2) Shipment Tracking API

- Purpose : Real time order tracking and delivery manage
- Example : Ship Engine API
- Features : Generate Shipping Labels
  Estimate delivery time and Cost.

TECHNICAL DOCUMENTATION:

Technical Documentation for E-Commerce.
This documentation provides a comprehensive guide to the ecommerce System's architecture, workflows, API endpoints and Sanity CMS Schema examples.

System Architecture Overview.

Frontend: The user interface where Customers browse the menu, Place orders and track deliveries.

Frame work: React / Next.JS

Styling : Tailwind CSS

Back End: A server that Process the orders, manage the inventory and integrates the third-Party APIs.

CMS:

Sanity: For managing dynamic content Such as product listing, description and blogs.

Deployment :

Hosting : Vercel or Netlify for deployment.

CI / CD : GitHub for automated deployment.

2- Key Workflows

User Registration : users registered by providing their e-mail, Password and profile details.

Login : user enter their Credentials to obtain a JWT Token enabling Secure Session handling.

Cart Management :

user add products to their Cart.

The Cart updates dynamically, storing items in the database or Local storage.

Check out

The user proceeds to checkout, providing Payment details.

API Endpoints :

- Authentication

  /api/users/register – Register a new user

  /api/users/login – Authenticate user

- Categories
  /api/categories - Retrive all Categories
  /api/categories {id} - Retrive products under a Specific Category.

- Cart
  /api/Cart/add - Add a product to the Cart
  /api/cart - Retrive Cart details.

- Orders
  /api/checkout - process payment and Create an Order -

5 Collaborate and Refine
feedback Integration: Continuously Collect feed back from Stackholders and end-users to enhance features.
Code Reviews: Conduct through peer reviews to maintain Code quality and identify potential issues early.
Documentation Updates: Regularly update this documentation to reflect changes in architecture, workflows on API functionality.

Sanity Schema Design

Product Schema
Product ID
Name
Description
Category
Discount
Price

Customer Schema
Customer ID
Name
Contact
Address
Email

Order Schema
Order ID
Customer ID
Product ID
Amount

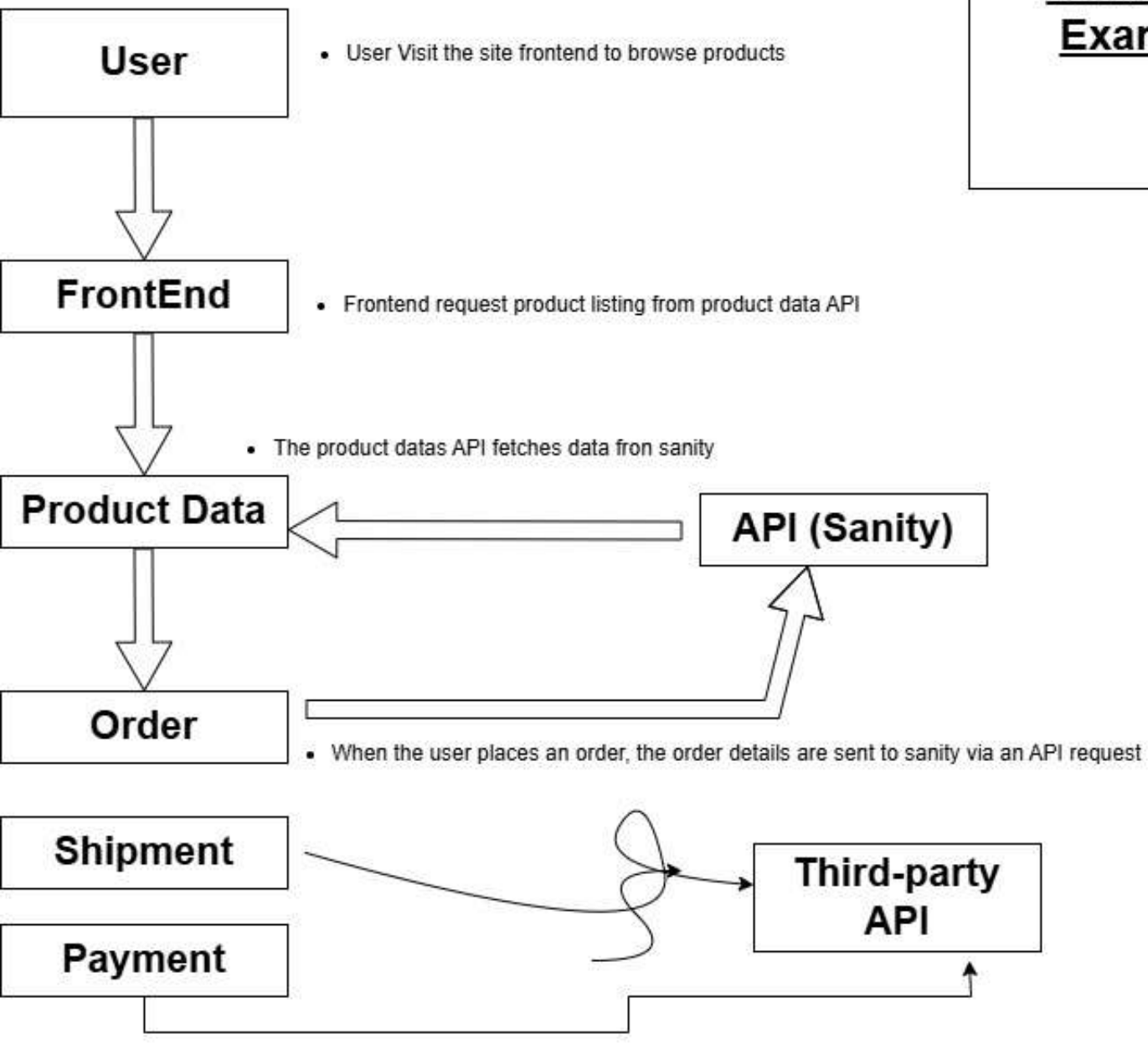
Payment Schema
Payment ID
Order ID
Amount
Payment Status

Shipment Schema
Shipment ID
Order ID
Courier Service
Shipment Status

Design System Architecture

To Visualize how this components interact

Data Flow Example



- User Visit the site frontend to browse products
- Frontend request product listing from product data API
- The product datas API fetches data fron sanity
- When the user places an order, the order details are sent to sanity via an API request
- Shipment tracking information is fetched through a Third-Party API
- Payment details are securely processed through the Payment Gateway

API Requirements

Here is the API endpoints for Q-Commerce food website

Products API : Fetch all products

Endpoint Name: /products

Method: Get

Description: fetch all products

Response Example:

```
[{
  id: 1,
 "name": "Pizza",
 "Price": 600,
 "stock": 50,
 "image": "https// public /image.png",
}]
```

Products API : Fetch single product

Endpoint Name: /products/{id}

Method: Get

Description: Fetches details of a single product by its ID.

Response Example:

```
[{
  id: 1,
 "name": "Pizza",
 "Price": 600,
 "stock": 50,
"image": "https// public /image.png",
}]
```

Orders Api: Create order

Endpoint Name: /orders

Method: post

Description: Creates a new food order with customer information and product details.

Response Example:

```
[{
   "customerId": 1,
   "products": [{"productId": 1, "quantity": 2},
                {productId": 2, "quantity": 1}],
    "totalAmount": 1000,
    "paymentStatus": "pending"
}]
```

Get Order Status

- **Endpoint Name:** `/order/{orderId}`
- **Method: GET**
- **Description: Fetches the status of a specific order by its ID.**
- **Response Example:**

Response Example:

```
{
   "orderId": 1234,
   "status": "in Process",
   "message": "Your order is being prepared"
}
```

Rider Status

- **Endpoint Name: /riders**
- **Method: Post**
- **Description: Assign a rider to a new order for delivery.**
- **Response Example: { "status": "Success", "message": "Rider assigned successfully" }**

Payment Status

- **Endpoint Name: /payment-status**
- **Method: Get**
- **Description: Check the payment status of an order**
- **Response Example: { "orderId": 12345, "paymentStatus": "Paid", "paymentMethod": "Credit Card" }**

Shipment Status

- **Endpoint Name: /shipment-status**
- **Method: Get**
- **Description: Fetch real-time delivery status for an order using third-party APIs.**
- **Response Example: { "orderId": 12345, "status": "In Transit", "ETA": "30 mins", "riderId": 789, "riderName": "John Doe", "riderLocation": "2 km away" }**