HACKATHON 3

DAY 2 PLANNING THE TECHNICAL FOUNDATION

Day 2 Activities : Transitioning to Technical Planning

1.Define Technical Requirements

The following technical requirements are necessary for my marketplace platform:

1.Frontend Requirements

★ **Objective :**

A well-designed gorgeous , user-friendly user interface which catches the user to our marketplace.

★ **Responsiveness** :

Work on the responsiveness of User interface which help User to interact on different devices easily .

★ **Essential Pages** :

Create the following pages:

1.Home Page
2.Signup Page
3.Menu Page
4.Shop Details Page
5.Blog Page
6.Add to cart Page
7.Checkout Page
8.Order Confirmation Page

2.Sanity CMS As Backend

★ objective:

Consider Sanity as a database and Use it to manage all data related to products , customers and order details.

- **Schema Design in sanity**

Design Schemas Like:

Products:

Create fields for product details {name , id , title, color, image , stock, rating etc.…}

Customers:

Create fields for customer details {name , id , email, contact info , etc.…}

Orders:

Create fields for order details {order id , product list , total payment, payment method , etc.…}

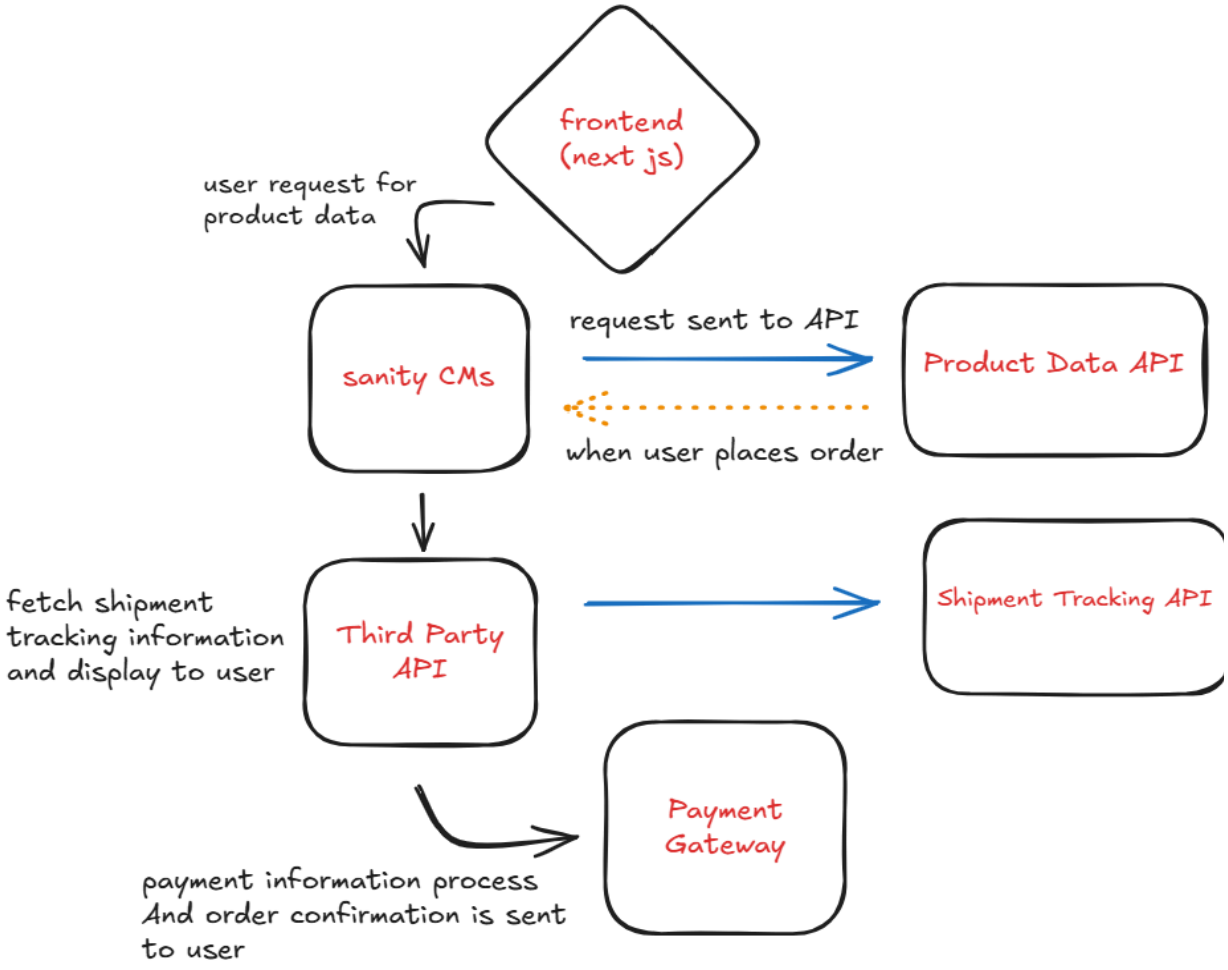
3.Third Party API

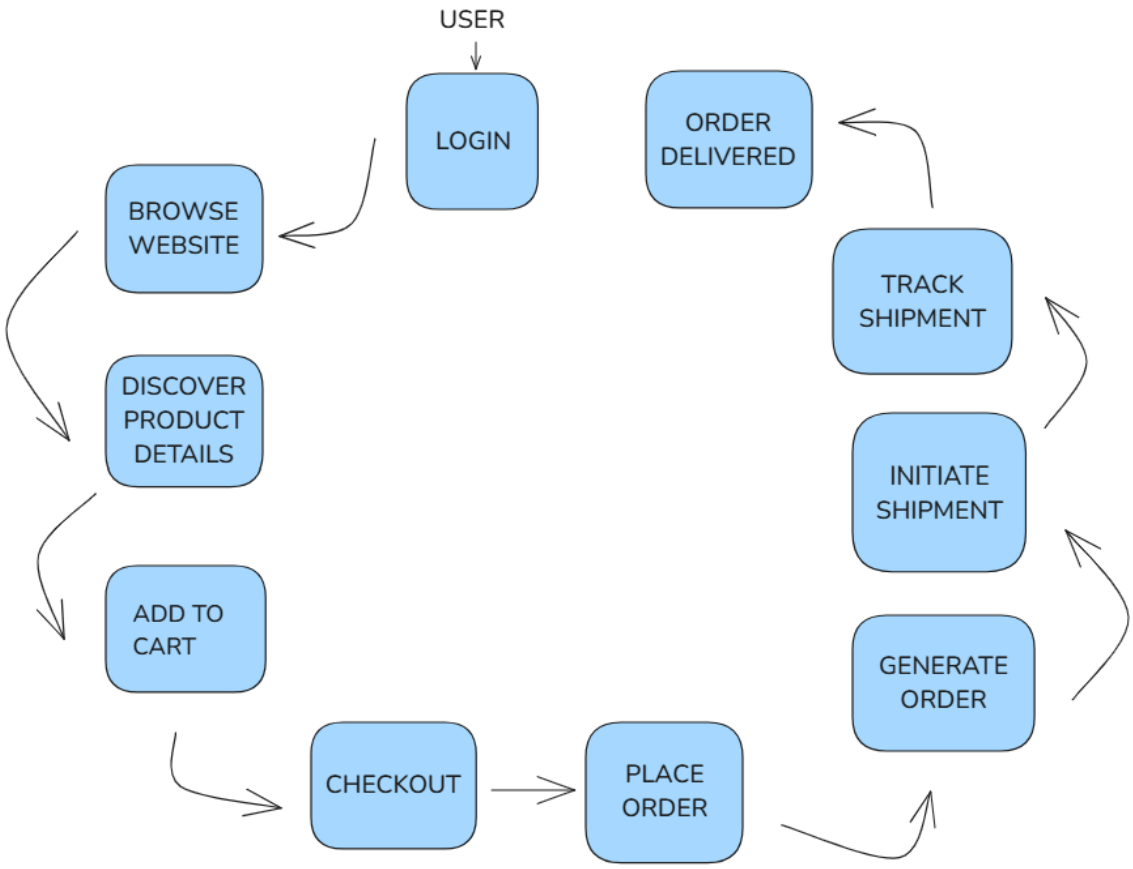
Usage of external third party API for enhancing functionalities of marketplace like:

- Product API
- Shipment Tracking API
- Payment Gateway API
- Logistic and Delivery API
- Restaurant and Menu management API
- Customer management API

2.Design System Architecture

System Architecture Diagram





KEY FLOW DIAGRAM

3.Plain API Requirements

ENDPOINT	METHOD	REQUEST	RESPONSE
/register	POST	request a new user to get register	{status :"succeed" Message:"user registered"}
/login	POST	allow user to login	{message:"welcome to website"}
/product/id	GET	fetch single product detail	{"id" , "name" , "image" ,"'price","stock",'rating"}

/orders	Post	create new order in sanity	{“customer info” ,”product details”, “payment status”}
/shipment	GET	Track order via API	{“shipment id”,”order id” ,”status” ,”expected delivery date”}

4.Sanity Schema Example

- **Product**

```
export default {
        name:”product”,
        title:”Product”,
        type:”document”,
        fields:[
            {name:’name’ , type:’string’} ,
            {name:’price’ , type:’number’} ,
            {name:’description’ , type:’text ’} ,
            {name:’image’ , type:’string’}  ,
            {name:’stock’ , type:’number’} ,
             {name:’rating’ , type:’string’} ,
             ]
             } ;
```

- **Order**

```
export default {
  name:”order”,
  title:”Order”,
  type:”document”,
  fields:[
      {name:’name’ , type:’string’} ,
      {name:’id’ , type:’number’} ,
      {name:’customer id’ , type:’number’} ,
      {name:’total amount’ , type:’number’}  ,
      {name:’status’ , type:’succeed’’} ,
       ] } ;
```

5.Conclusion

In conclusion, this technical planning documentation has outlined the key technical requirements and considerations for building a scalable and efficient Q-commerce platform.By following the technical plan outlined in this document, we are confident that the Q-commerce platform will be a success, providing a scalable, secure, and high-performance platform for users to shop and engage with.

Prepared by: Narmeen Zubair

Slot: Tuesday 2 to 5

Sir : Sir Ali Aftab Sheikh & Sir Fahad