Hackathon # 03

Date

Day 2: Goal - Transition from Business Planning To Technical Implementation

Today, we focus on converting the business plan into a technical framework. The aim is to establish a foundation for the marketplace's technical structure, including system architecture, workflows, and API design.

Key Activities for day 2

Step 1: Define Technical Requirements

* Frontend Requirements

1. Design a users-friendly interface that includes:
   - Homepage: Highlights featured products and categories.
   - Product Listing Page: Displays all products in a grid or list format.
   - Product Details Page: Shows product-specific details such as description, price, reviews, etc.
   - Cart Page: Allows users to view selected items before checkout.
   - Checkout Page: Processes orders with payment options.
   - Order Confirmation Page: Displays purchase details after checkout.

2. Focus on responsive design:
   - Ensure the marketplace is mobile-friendly.
   - Optimize performance for low-end devices.

Signature

No.

Date

★ Backend Requirements (Sanity CMS)

1. Use Sanity CMS to manage:
   - Products: Add, update, and delete products.
   - Orders: Store customer order details.
   - Users: Manage user accounts and profiles.

2. Draft schemas for each major entity in Sanity (e.g., Products, Orders, Users).

- For example:

```
export default {
  name: "product",
  type: "document",
  title: "Product",
  fields: [
    {name: "name", type: "string", title: "Product Name"},
    {name: "price", type: "number", title: "Price"},
    {name: "description", type: "text", title: "Description"},
    {name: "image", type: "image", title: "Product Image"},
  ]
}
```

★ Third-Party APIs

Integrate APIs for:

1. Payment Gateways (e.g., Stripe, PayPal): Handle secure transaction

2. Shipment Tracking (e.g., Shippo, Easy Post): Provide order Tracking updates

3. Authentication (e.g., Firebase, Auth0): Enable secure login

Signature _____ No. _____

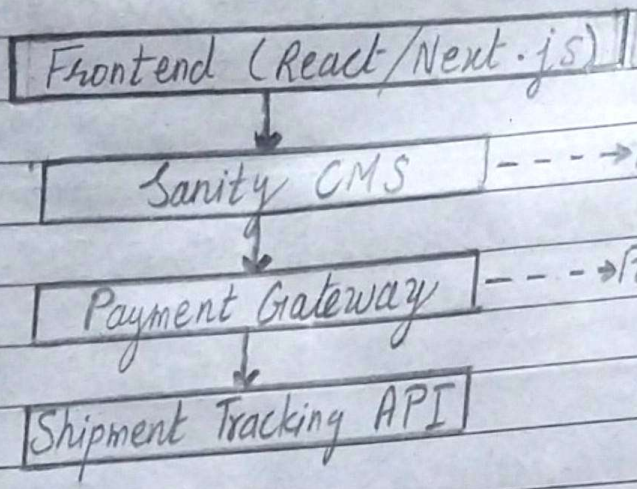
Date

and user management.

Step 2: Design System Architecture

1- Create a Visual diagram showing component interactions. Tools like Lucidchart or pen-and-paper work well.

Example Architecture:



Workflow Example

- Step 1: User browses products via the "Frontend".
- Step 2: Frontend fetches product data from Sanity CMS using the "Product API".
- Step 3: User adds items to the cart and checks out.
- Step 4: Payment Gateway processes the transaction.
- Step 5: Shipment Tracking API provides real-time delivery updates

Date

Step 3: Define API Requirements

* Endpoints Overview

Endpoint	Method	Purpose	Response Example
//products	GET	Fetch all available products.	{ "id" : 1, "name" : "Sofa A", "price" : 200, "stock" : 10, "image" : "url" }
//order	POST	Create a new order	{"order Id" : 123, "status" : "Success", "total" : 300}
//shipment-status	GET	Track shipment status	{"order Id" : 123, "status" : "In Transit", "ETA" : "3 days"}

* Explanation:

- /product : Fetches a list of all products available in the market-place. This endpoint will be used to display product detail like name, price, stock, and images on the frontend.

- /ordess : Allows the creation of a new order. The frontend will interact with this endpoint when a user completes a checkout.

- /shipment-status : Used for tracking orders. Once an order is placed, users can check the shipment status, including estimated delivery time

Signature

No.

Date

Conclusion :-

By the end of this phase, I aim to have :

1 - A clear technical plan that aligns with my business goals.
2 - My system architecture diagram showing how components interact.
3 - API documentation for my key endpoints like `/products`, `/orders`, and `/shipment-status`.
4 - Drafted Sanity CMS schemas for entities such as products and orders.
5 - A detailed workflow explanation outlining how my marketplace function.

This foundation will ensure that my project transitions seamlessly into the development phase, setting the stage for a scalable and efficient marketplace.

X——————X