Date: 17/Jan/2025 Hackathon#3 Day-2

Goal - Transition from Buisness planning To Technical Implement:

Today we focus on converting the buisness plan into a technical framework - The aim is to establish a foundation for the marketplace technical structure., work flows, API Design-

"Key Activities":

1) Technical Requirments:

* Frontend Requirment:

1) Design a user friendly interface that include:
- Home page : Highlight feature, product and categories
- Product listing page : Display all product in a grid or list formet-
- Product Detail page : Display product- Specific detail Such as description, price, review and etc.
- Cart page : Allow users to view selected items before checkout.
- Order Confirmation : Display purchase detail after checkout-

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

Backend Requirment (Sanity CMS)

use Sanity CMS to manage:
product: Add, Update, and Delete product
Orders: Store Customers Order details.
Users: Manage user account and profit

Draft schemas for each major entity in Sanity (eg. products, orders, zone).

For example.

```
export default {
name: "product",
type: "document",
title: "Product",
field: [
{ name:"name", type:"string", title:"Product name"},
name:"price", type:"numbers", title:"price"},
name: "description", type:"text", title: "Description"},
name:"image", type:"image", title:"Product Image"},
[ ]
}
```

* Third - Party APIs

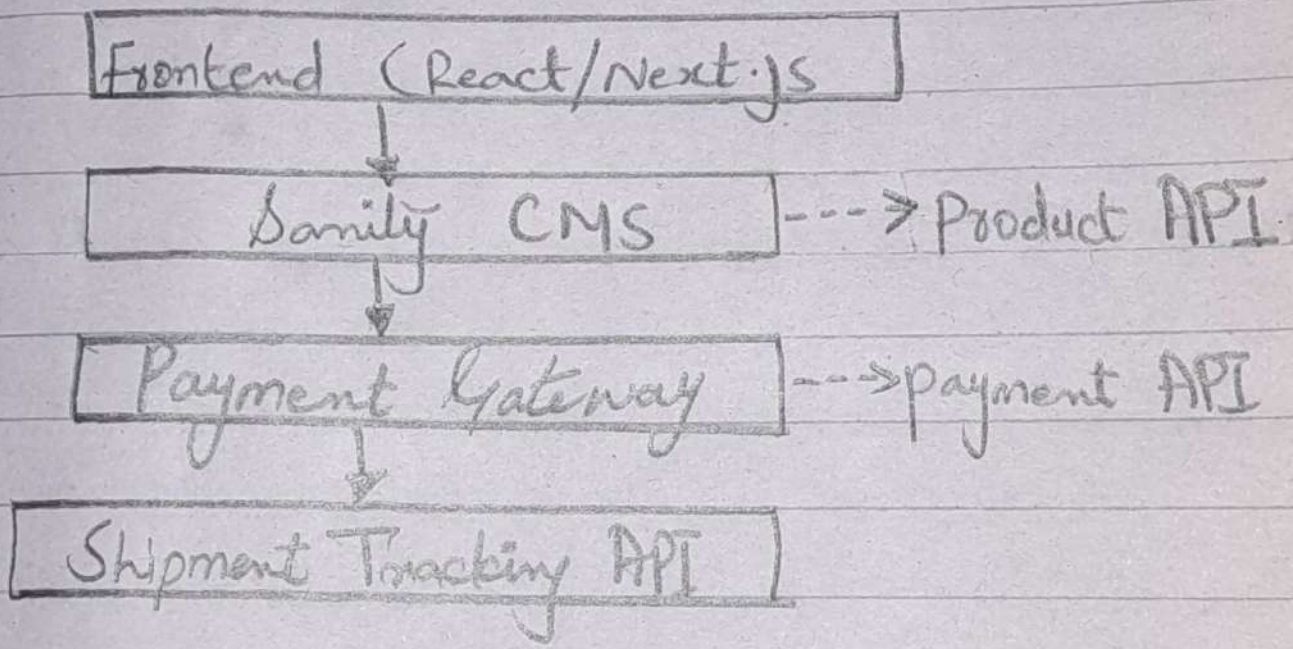
Integrate APIs for:

1) Payment Gateway (e.g, stripe, paypal): Handle Secure transaction
2) Shipment Tracking (e.g: Shippo, easy post): provide Order tracking update.
3) Authentication (e.g firebase, Auth0): enable Secure login and user management-

2) Design System Architecture:

* Create a visual Design - Showing Component interaction - Tools like Lucidchart or pen and paper work well -

Example Architecture:



Workflow Example:

1) User browser products via the frontend.
2) Front end fetch products data from Sanity CMS using the "product API".
3) User adds items to the cart and checkout
4) payment Gateway process the transaction.
5) Shipment tracking API provide real-time Delivery updates.

3) Define APIs Requirements.

* Endpoint Overview:

Endpoint	Method	Purpose	Response Example
//product	GET	fetch all available Products	{"id": 1, "name": "Shoes" "price": 1000, "stock": 12, "image": "url" }
// order	POST	Create a new order	{"Order Id": 123, "Status": "Success", "total": 1400 }

/Shipment	GET	Track Shipment Status	{"OrderId": 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 details like name, price, stock and image on the frontend.

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

/Shipment: used for tracking orders. Once an order is placed, user can check the Shipment process, including estimated delivery time.

Conclusions :-

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

1) A clear technical plan that align with my buisness goals-
2) My system architecture diagram showing how component interact API documentations for my key end points like /products, /orders, /shipment-status.
3) Draft Sanity CMS Schemas for entities such as products.
4) A detailed workflow explanations out ling how my marketplace function.

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