TECHNICAL FOUNDATIONS OF MY MARKETPLACE HACKATHON

Day 2 Activities: Transitioning to Technical Planning

1. Defining Technical Requirements

🆆🆁🅸🆃🅴 Technologies

i. Next.js

ii. Shadcn library for pre made components

iii. React icons for icons

iv. Sanity(CMS) as backend

🆆🆁🅸🆃🅴 Frontend Requirements:

o User-friendly interface for browsing products.

o Responsive design for mobile and desktop users.

o Essential pages: Home, Product Listing, Product Details, Cart, Checkout, and Order Confirmation.

🆆🆁🅸🆃🅴 Sanity CMS as Backend:

o Sanity CMS to manage product data, customer details, and order records. Sanity acts as the database for my marketplace.

o Focusing on designing schemas in Sanity to align with the business goals from Day 1.

🆆🆁🅸🆃🅴 Third-Party APIs:

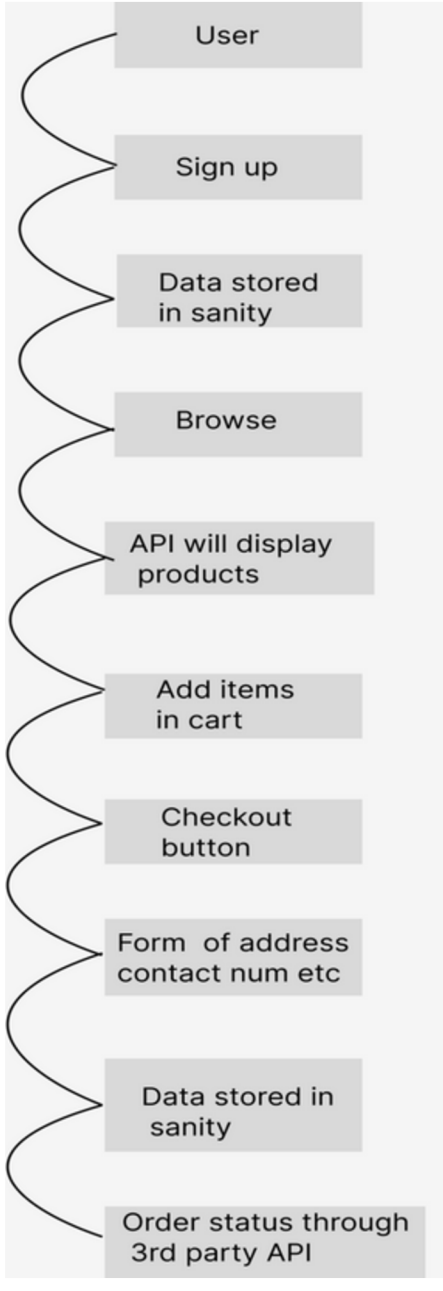
o APIs will be provided by our Teachers

o Ensure APIs provide the necessary data for frontend functionality.

o Implementation of APIs according to product requirement

System Architecture

The whole architecture of the website is defined in this picture.



- User will sign up.
- User data will be stored in sanity
- The API will provide products details as the user visits product page

- As user adds items in cart and clicks on checkout button a form will appear
- After submitting form the data will be stored in sanity
- The user will get the status of the order from Order details page and the status will be from a 3rd party API means shipment API and the payment system will be COD cash on delivery

API REQUIREMENTS

Planning API Requirements

1. Endpoint Name: /products

- Method: GET
- Description: Fetch all available products from Sanity.
- Response: Product details (ID, name, price, stock, image).

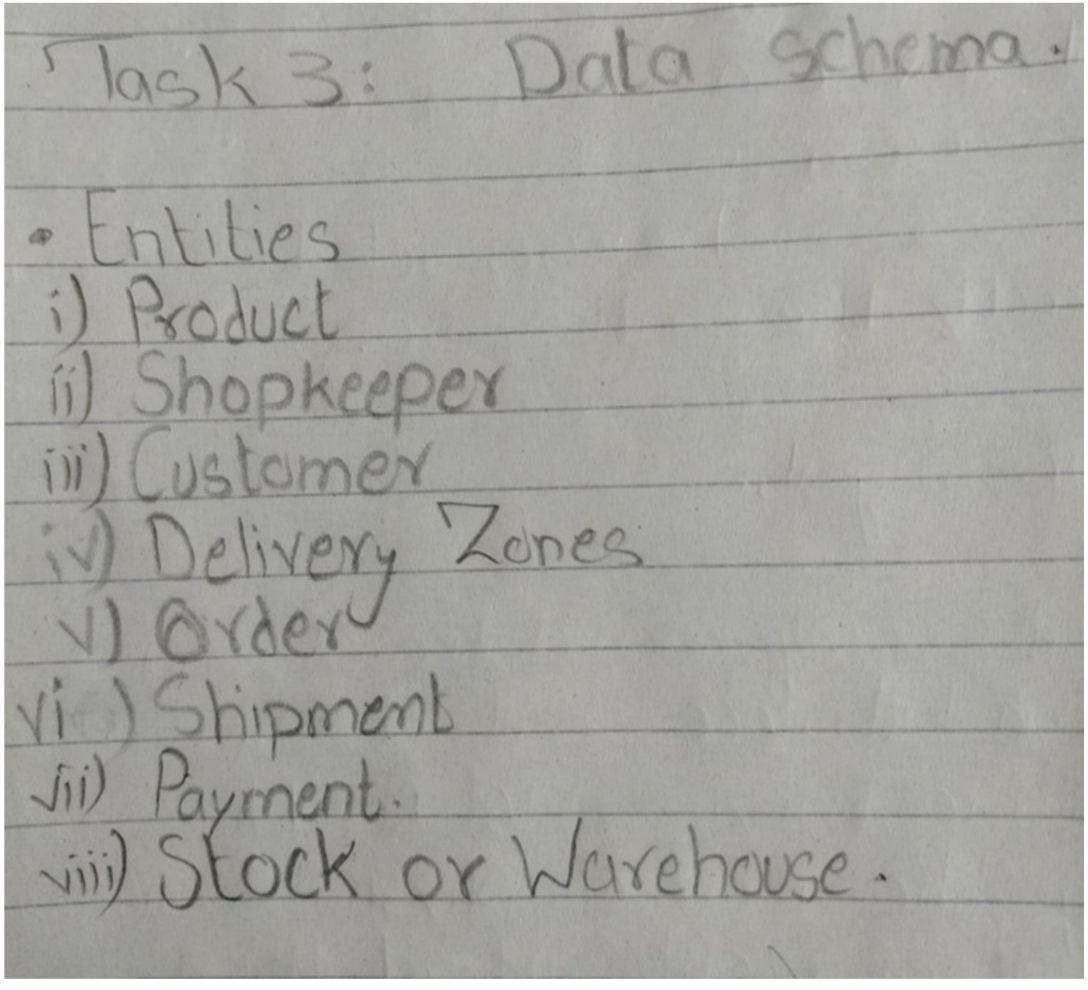
2. Endpoint Name: /orders

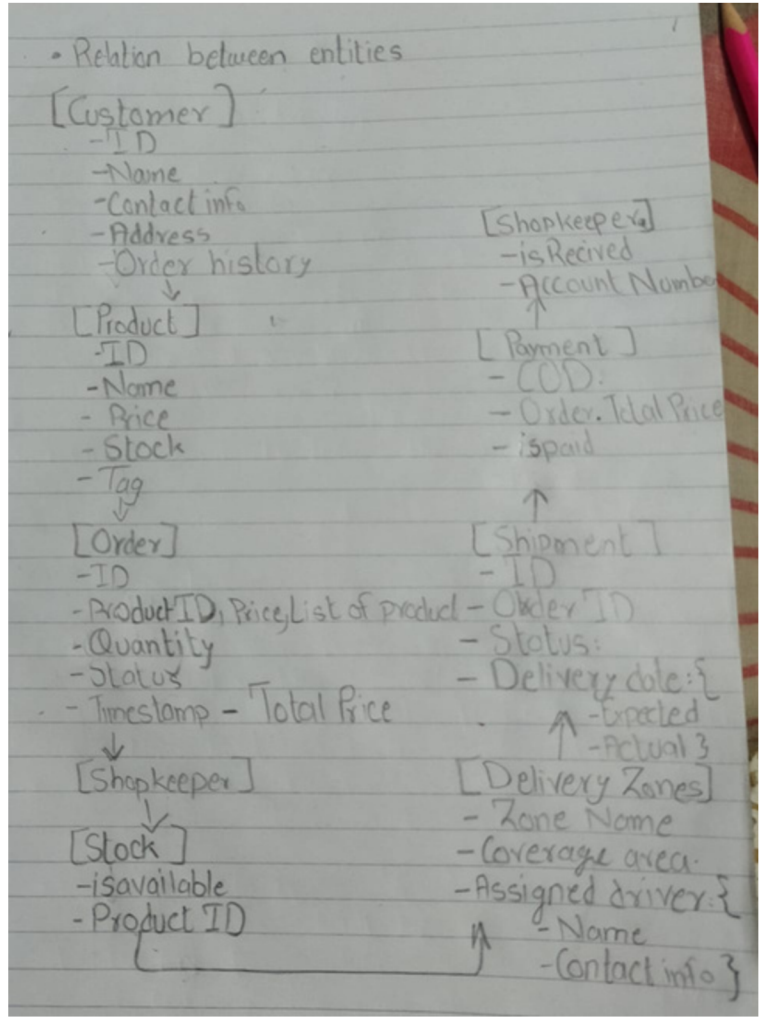
- Method: POST
- Description: Create a new order in Sanity.
- Payload: Customer info, product details, payment status.

3.Endpoint Name: /shipment

- Method: GET
- Description: Track order status via third-party API.
- Response: Shipment ID, order ID, status, expected delivery date.

DATA SCHEMA


Task 3: Data schema.
• Entities
i) Product
ii) Shopkeeper
iii) Customer
iv) Delivery Zones
v) Order
vi) Shipment
vii) Payment.
viii) Stock or Warehouse.


• Relation between entities
[Customer]
-ID
-Name
-Contact info
-Address
-Order history
[Product]
-ID
-Name
-Price
-Stock
-Tag
[Order]
-ID
-ProductID, Price, List of product
-Quantity
-Status
-Timestamp - Total Price
[Shopkeeper]
[Stock]
-isavailable
-Product ID
[Shopkeeper]
-isRecived
-Account Numbe
[Payment]
- COD:
- Order. Total Price
- ispaid
[Shipment]
- ID
- Order ID
- Status:
- Delivery date: {
-Expected
-Actual }
[Delivery Zones]
- Zone Name
-Coverage area
-Assigned driver: {
-Name
-Contact info }