Hackathon Day 2: Planning the Technical Foundation

1-Define Technical Requirements

Frontend Requirements

User-friendly Interface: Provide an easy navigation experience for browsing

Responsive Design: Ensure the design adapts seamlessly for mobile, tablet, and desktop devices

Essential Pages

- Home
- Product Listing
- Product Details
- Cart
- Checkout
- Order Confirmation

Sanity CMS as Backend

Sanity CMS: Use Sanity CMS to manage product data, customer details, and order records

Third-Party APIs

API Integration: Integrate third-party APIs for shipment tracking, payment gateways, and backend services.

Data Requirements: Ensure APIs provide the necessary data for frontend functionalities (order tracking, payment status, etc.)

2-Design System Architecture

[Frontend (Next.js)]

|

[Sanity CMS] ------------> [Product Data API]

|

[Third-Party API] -----> [Shipment Tracking API]

|

[Payment Gateway]

2-Key Workflow

User Registration: User signs up → Data stored in Sanity → Confirmation sent to user

Product Browsing: User browses categories → Product data fetched from Sanity API → Displayed on the frontend.

Order Placement: User adds products to cart → Proceeds to checkout → Order saved in Sanity.

Shipment Tracking: Order updates fetched via third-party API → Displayed in real-time.

3-APIs Requirements

1. User Management

Endpoint: /users
Method: POST
Description: Registers a new user

Endpoint: /users/{userId}
Method: GET
Description: Fetches the details of a user.

Endpoint: /users/{userId}
Method: PUT
Description: Updates an existing user's information.

Endpoint: /users/{userId}
Method: DELETE
Description: Deletes a user from the system.

2. Seller Management

Endpoint: /sellers
Method: POST
Description: Registers a new seller.

Endpoint: /sellers/{sellerId}
Method: GET

Description: Fetches the details of a seller.

Endpoint: /sellers/{sellerId}
Method: PUT
Description: Updates a seller’s information.

Endpoint: /sellers/{sellerId}
Method: DELETE
Description: Deletes a seller from the system

3. Products

Endpoint: /products
Method: POST
Description: Adds a new product to the catalog

Endpoint: /products
Method: GET
Description: Fetches a list of all products.

Endpoint: /products/{productId}
Method: PUT
Description: Updates a product’s details.

Endpoint: /products/{productId}
Method: DELETE
Description: Deletes a product from the catalog.

4. Orders

Endpoint: /orders
Method: POST
Description: Creates a new order.

Endpoint: /orders/{orderId}
Method: GET
Description: Fetches details of a specific order.

Endpoint: /orders/{orderId}
Method: PUT
Description: Updates the status of an order.

Endpoint: /orders/{orderId}
Method: DELETE
Description: Cancels an order

5. OrderDetails

Endpoint: /orderdetails/{orderId}
Method: GET
Description: Fetches detailed information about products in an order.

6. Payments

Endpoint: /payments
Method: POST
Description: Creates a new payment record.

7. Delivery Zones

Endpoint: /deliveryzones
Method: POST
Description: Adds a new delivery zone.

8. Seller Delivery

Endpoint: /sellerdelivery
Method: POST
Description: Creates a delivery record for the seller.

3-Sanity Schema

User

Id

Name

Email

Password

Role

Address

Phone Number

Seller

Id

User id

Store Name

Email

Products

Id

Name

Price

Category

Stock

Seller id

Orders

Id

User id

Zone id

Order status

Order Date

Orders Details

Id

Order id

Product id

Quantity

Price

Payments

Id

Order id

Amount

Payment Method

Payment Status

Delivery Zones

Id

Zones name

City

Delivery Charges

Estimated Delivery Time

Seller Delivery

Zone

Seller id

Zone id

Sumbal Naz