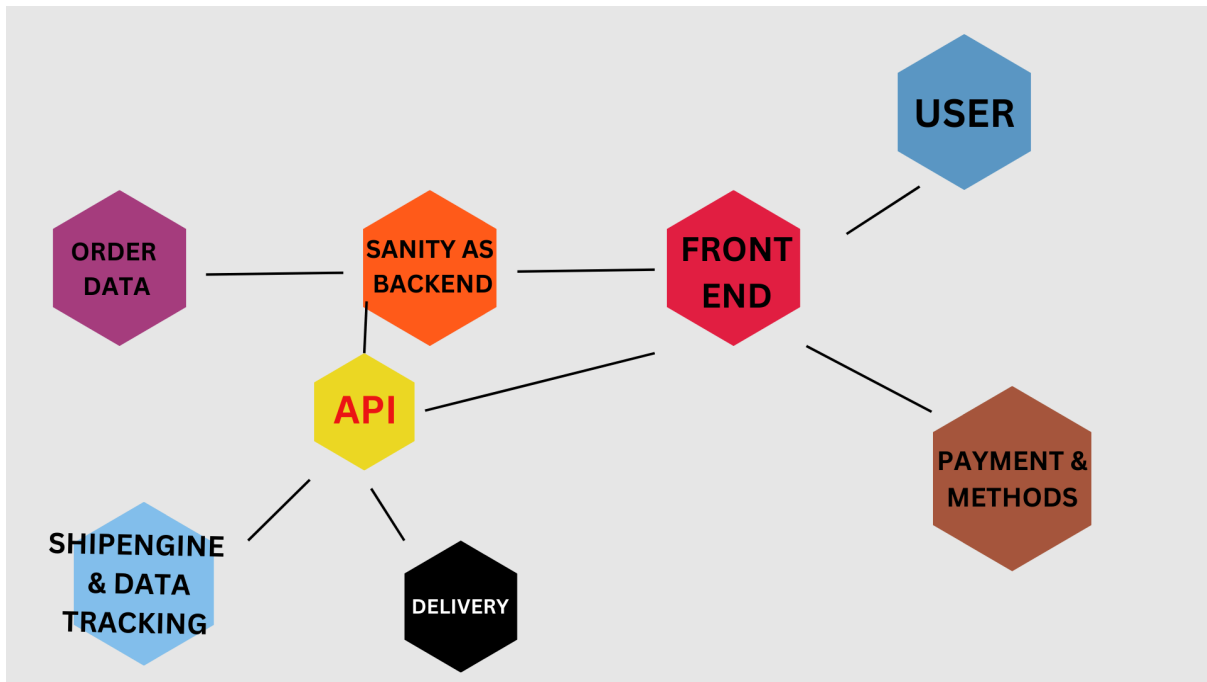


DAY 2 Planning the Technical Foundation



Frontend

Technologies:

- Next.js for server-side rendering and faster page loads.
- Tailwind CSS for styling and responsive design.

Key Features:

- Build a user-friendly interface for browsing products and placing orders.
- Design responsive layouts that work on both mobile and desktop devices.

Implement core pages:

- Home, Collection, About, Contact, Product Details
- Add to cart, Checkout & Order Confirmation.

Sanity CMS as Backend

Purpose:

- Use Sanity CMS to manage product data, customer records, and order information.
- Define structured content schemas for:
 - **Product Details** (ID, Name, Price, Stock, Images).
 - **Orders** (Order ID, Customer ID, Product IDs, Quantity, Price).
 - **Customers** (Customer ID, Customer Name, Address, ZipCode, Contact Info, and Purchase History).
- Real-time updates to reflect stock changes and order processing.

Third-Party APIs

ShipEngine:

- For shipment tracking and logistics.
- Enables assigning delivery riders and tracking shipment zones.

Payment Gateway:

- Support Cash on Delivery and Digital Payments (Stripe, PayPal, or Easypaisa, JazzCash).

Notification APIs:

- Use services like Twilio or Firebase for SMS/Email notifications about order status.

API Requirements

1.) Product Management API

Endpoint Name: /products

< Method: GET >

Description: Fetch all available products from the Sanity CMS.

Response Example:

```
[{ "id": 1, "name": "Product A", "price": 120 , Category: chair }, { "id": 2, "name": "Product B", "price": 250 , Category: chair }]
```

API Requirements

2. Order Management API

Endpoint Name: /orders

< Method: POST >

Description: Create a new order in the system.

Response Example:

```
{"order_id": 456, "status": "Success", "message": "Order created successfully"}
```

API Requirements

3. Shipment Management API

Endpoint Name: /shipment

< Method: GET >

Description: Track the order status via a third-party API.

Response Example:

```
{"shipment_id": 789, "order_id": 456, "status": "In Transit", "expected_delivery_date": "2025-02-7"}
```

API Requirements

4. Express Delivery Status API

Endpoint Name: /express-delivery-status

< Method: GET >

Description: Fetch real-time delivery updates for perishable items.

Response Example:

```
{"order_id": 123, "status": "In Transit", "ETA": "45 minutes"}
```
