Hackathon-3 Day 2: Transitioning to Technical Planning

By Kanwal Rafiqe.

Roll number: 00311635

Wednesday 7:00-10: pm

Technical Requirements

- ## Frontend Requirement:
  1. **User-Friendly Interface**
     - o Clean design for easy product browsing, filtering, and sorting.
  2. **Responsive Design**
     - o Adapts seamlessly to mobile, tablet, and desktop devices.
  3. **Essential Pages**
     - o **Home Page**: Displays featured collections and promotions.
     - o **Product Listing Page**: Shows categorized products with filters.
     - o **Product Details Page**: Provides detailed product info and purchase options.
     - o **Cart Page**: Lists selected items with editing options.
     - o **Checkout Page**: Collects payment and shipping details.
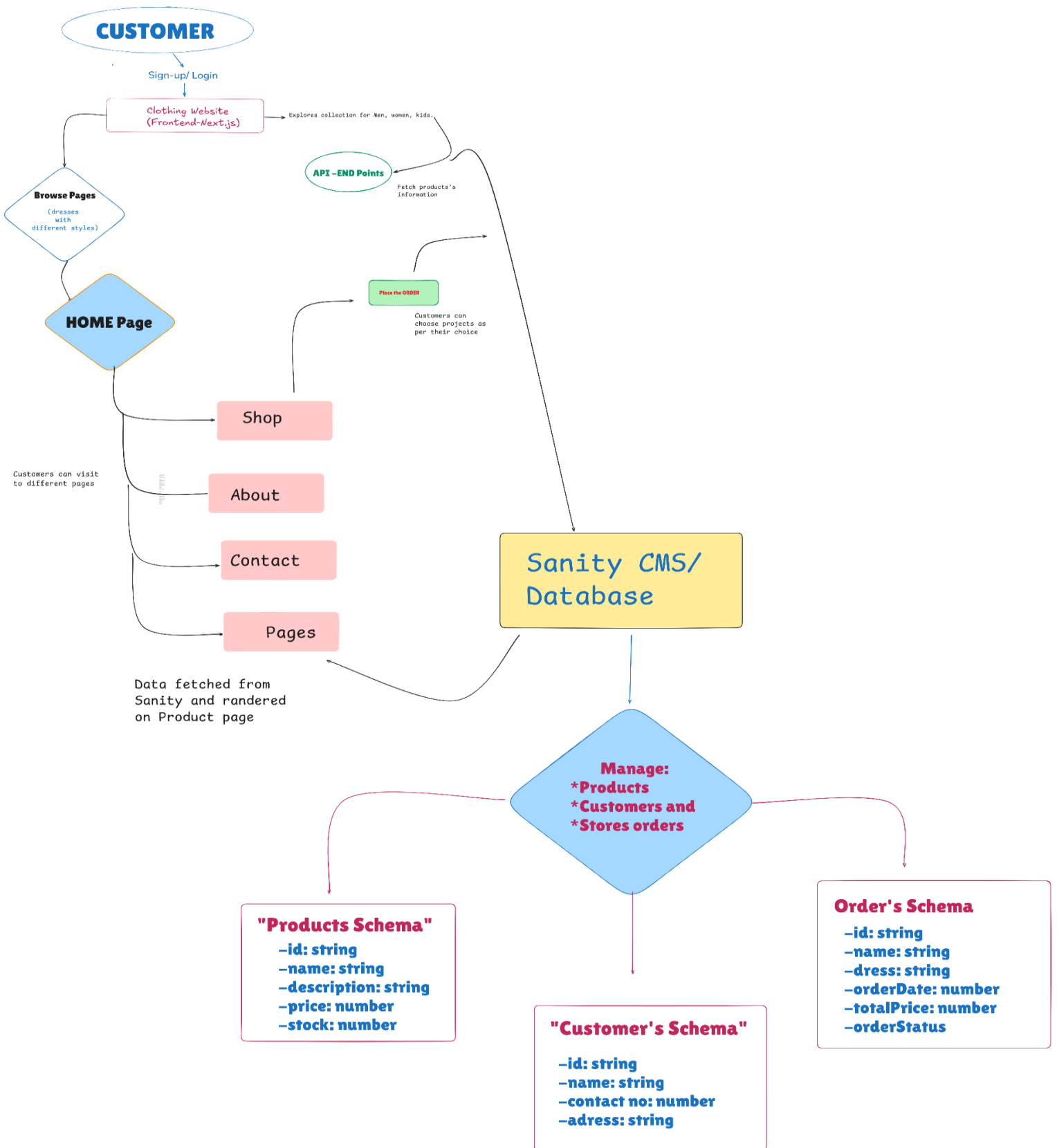   - o **Order Confirmation Page**: Confirms order placement and shows details.
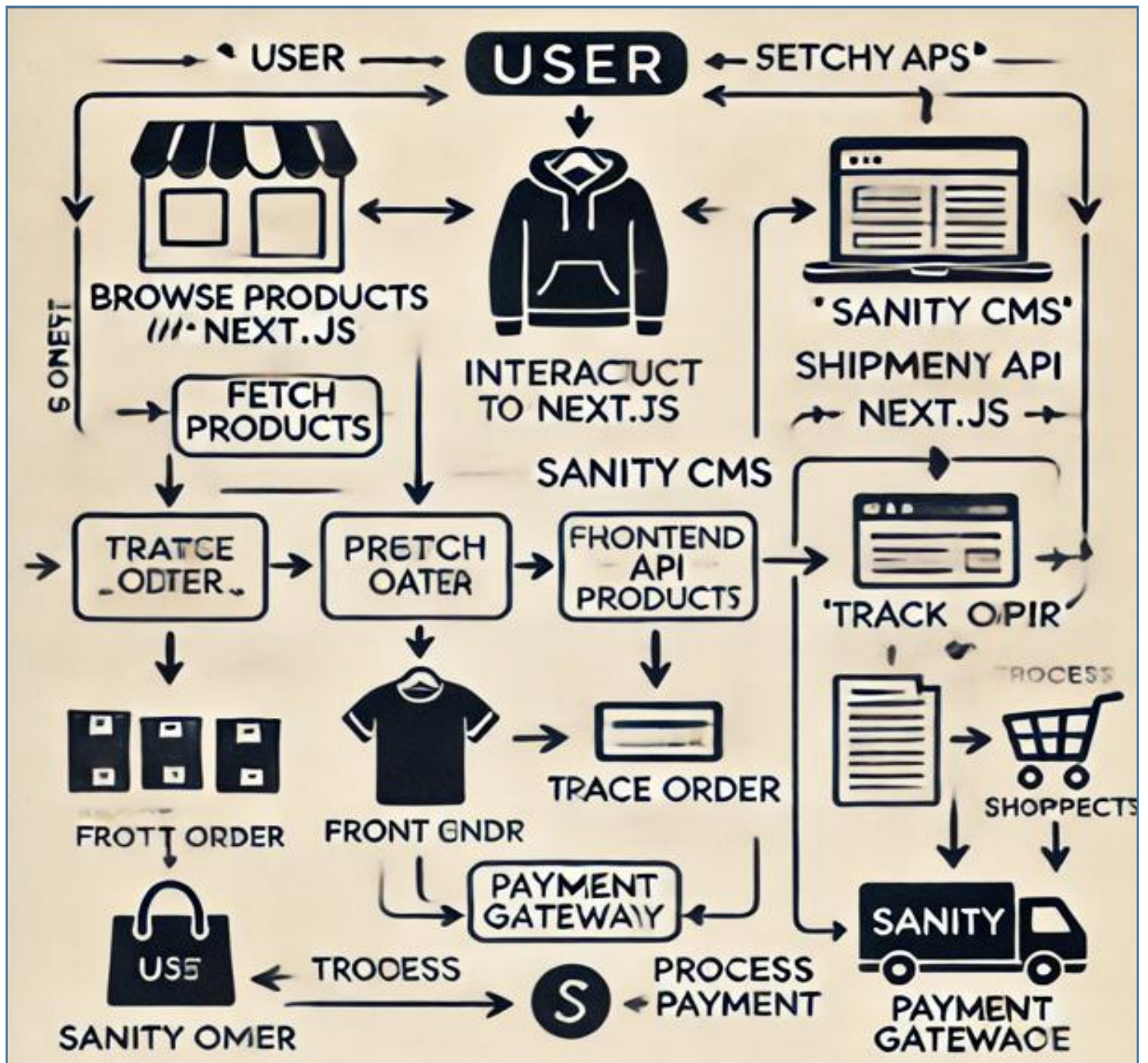
- ## Sanity CMS as Backend
  1. **Product Management**
     - a. Schemas for categories, products, and variants (e.g., size, color).
  2. **Customer Data Management**
     - a. Stores customer profiles and addresses.
  3. **Order Records Management**
     - a. Tracks order details, payment status, and shipping info.

- ## Third-Party APIs

  1. **Shipment Tracking API**
     - o Integrates Shippo, for tracking deliveries.
  2. **Payment Gateway API**
     - o Processes transactions via Stripe.
  3. **Frontend API Integration**
     - o Updates order status and cart dynamically

System Architecture [Frontend (Next.js)

CUSTOMER

Sign-up/ Login

Clothing Website
(Frontend-Next.js) → Explores collection for Men, women, kids.

API –END Points

Fetch products's
information

Browse Pages

(dresses
with
different styles)

HOME Page

Place the ORDER

Customers can
choose projects as
per their choice

Shop

About

Contact

Customers can visit
to different pages

Pages

**Sanity CMS/
Database**

Data fetched from
Sanity and randered
on Product page

Manage:
*Products
*Customers and
*Stores orders

"Products Schema"
  –id: string
  –name: string
  –description: string
  –price: number
  –stock: number

"Customer's Schema"

  –id: string
  –name: string
  –contact no: number
  –adress: string

Order's Schema
  –id: string
  –name: string
  –dress: string
  –orderDate: number
  –totalPrice: number
  –orderStatus

Key Workflows:

APIs End Points:

Endpoints	Method	Description
/Products	Get	Fetches products/dresses
/Products/id	Get	Fetches a specific product/dress
Orders	Post	Creates a new order
Orders/id	Get	Fetches specific order
/shipment/tracking/id	Get	Tracks shipment status

Sanity Schema:

Product's Schema:
```
export const product = {
name: 'product',
type: 'document',
fields: [
{name: 'name', type: 'string', title: 'Product's name'},
{name: 'price, type: 'number, title: 'Price},
{name: 'image, type: 'image, title: 'Product's image},
{name: 'category, type: 'string', title: 'Catogery},
 ]
};
```

Order's Schema:
```
export const product = {
name: 'order,
type: 'document',
fields: [
{name: 'orderNumber', type: 'string', title: 'Order Number'},
{name: 'customer, type: 'reference, to: [{type: 'customer'}]title: 'Customer'},
 ]
};
```

Customer's Schema:
```
export const product = {
name: 'customer,
type: 'document',
fields: [
{name: 'name', type: 'string', title: 'Customer's name'},
{name: 'adress, type: 'string, title: 'Adress'},
{name: 'phone, type: 'number, title: 'Number'},
 ] };
```