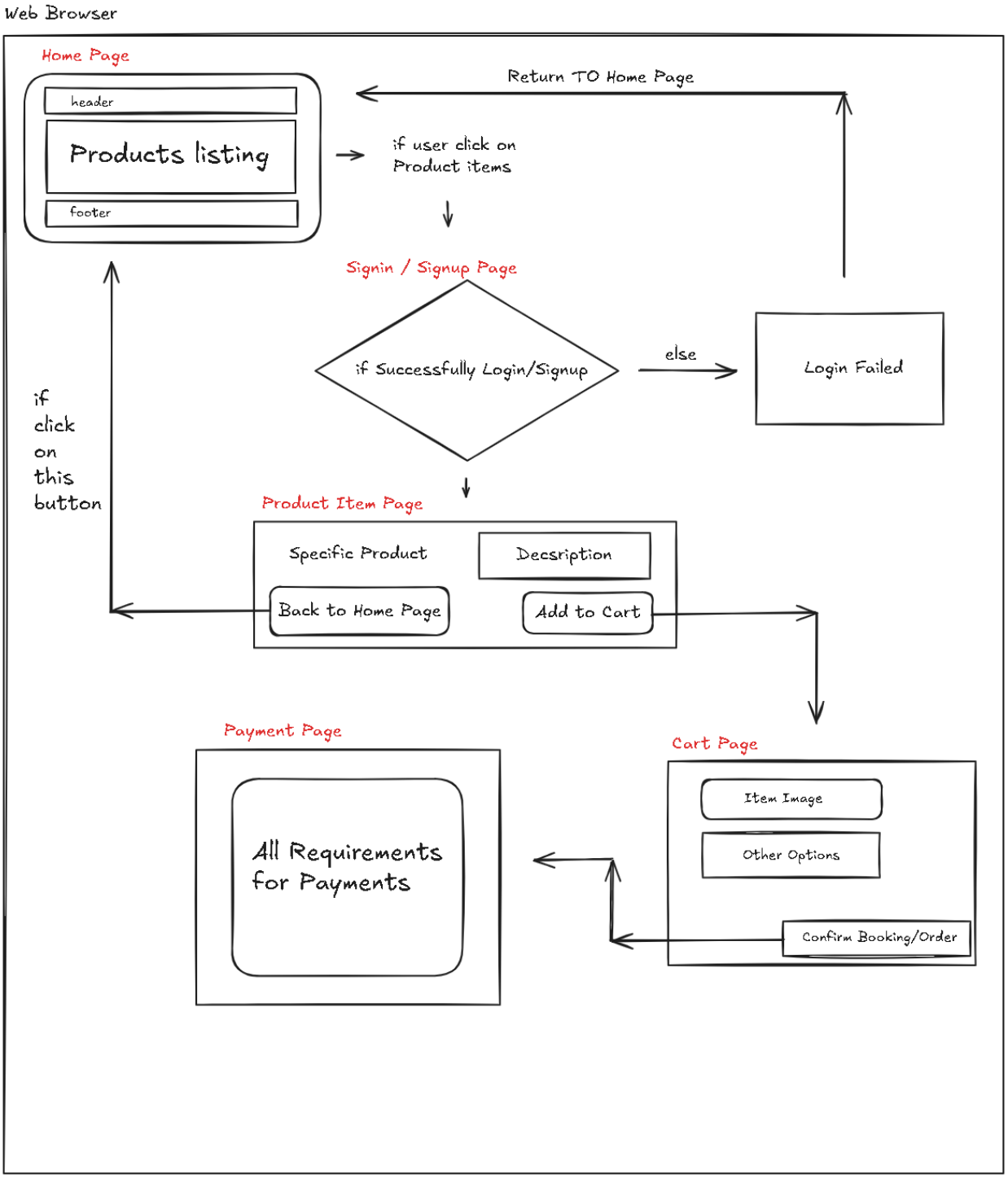
MARKET PLACE HACKATHON 2025
DAY 02: TECHNICAL FOUNDATION PLANNING

1. TECHNICAL REQUIREMENTS

a. FOR FRONTEND

The below diagram illustrate the technical requirements for the frontend which will build using Next.js and Tailwind CSS. More details for the frontend on how it is working are as follows:

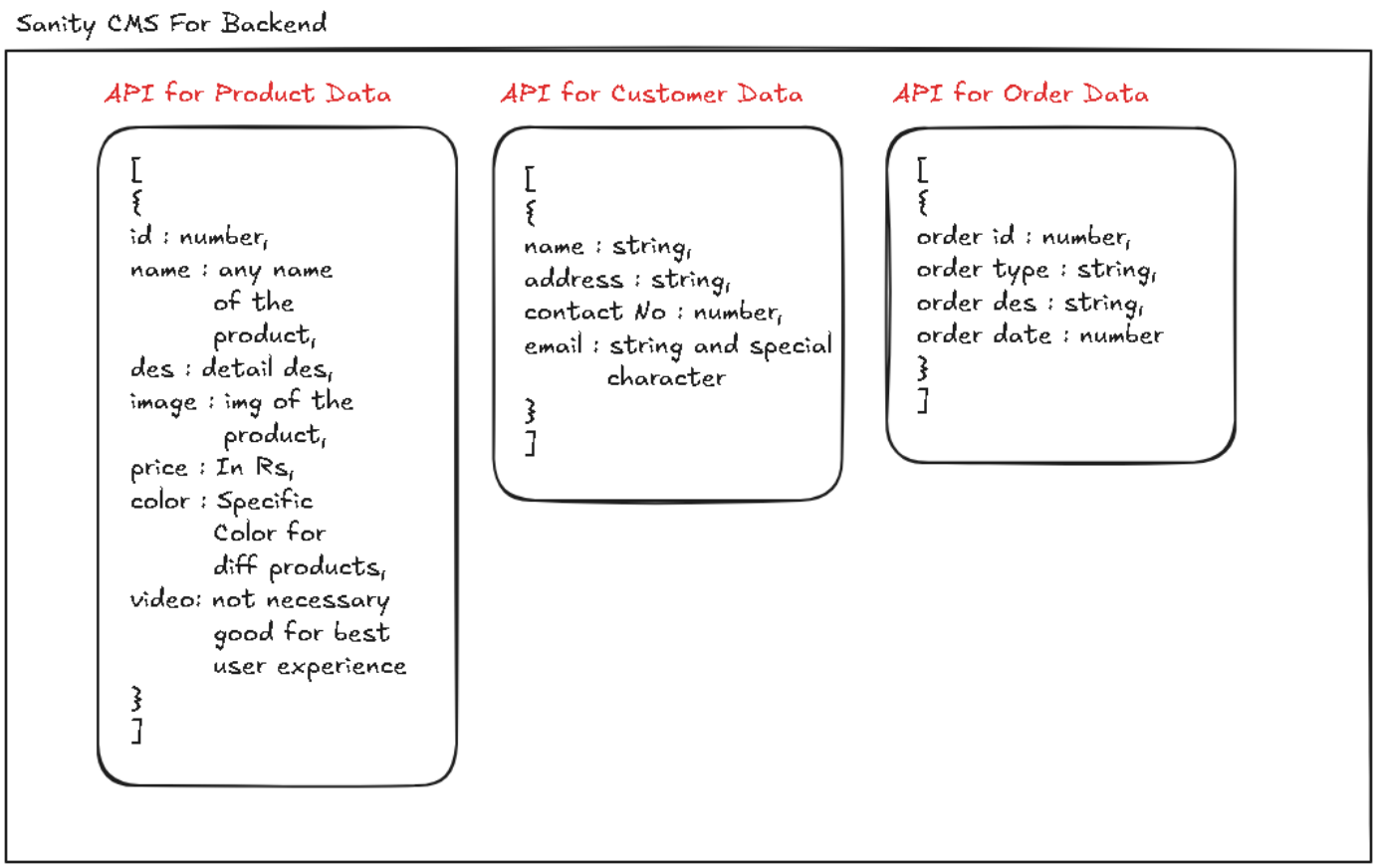


Key Points:

- User visits to the web Home page.
- If the user wants to make a booking for the products on rent, He/ She must sign in / sign up first.
- If sign in / sign up failed the user again navigated to the Home Page.
- Once the successful completion of sign in / sign up the user directly navigates to the product item page.
- There He / She have two options either to go back to Home Page or to add the product to cart.
- Once added to cart, user will cross check the items and confirm the order by clicking the button.
- As user confirm the order, He/ She navigates to the Payment Page where the user fulfilled all the requirements for payment.

b. FOR BACKEND (Sanity CMS)

The below diagram shows the API for products details, customer data, and order data which act as a database. Detail information is as under:



Key Points:

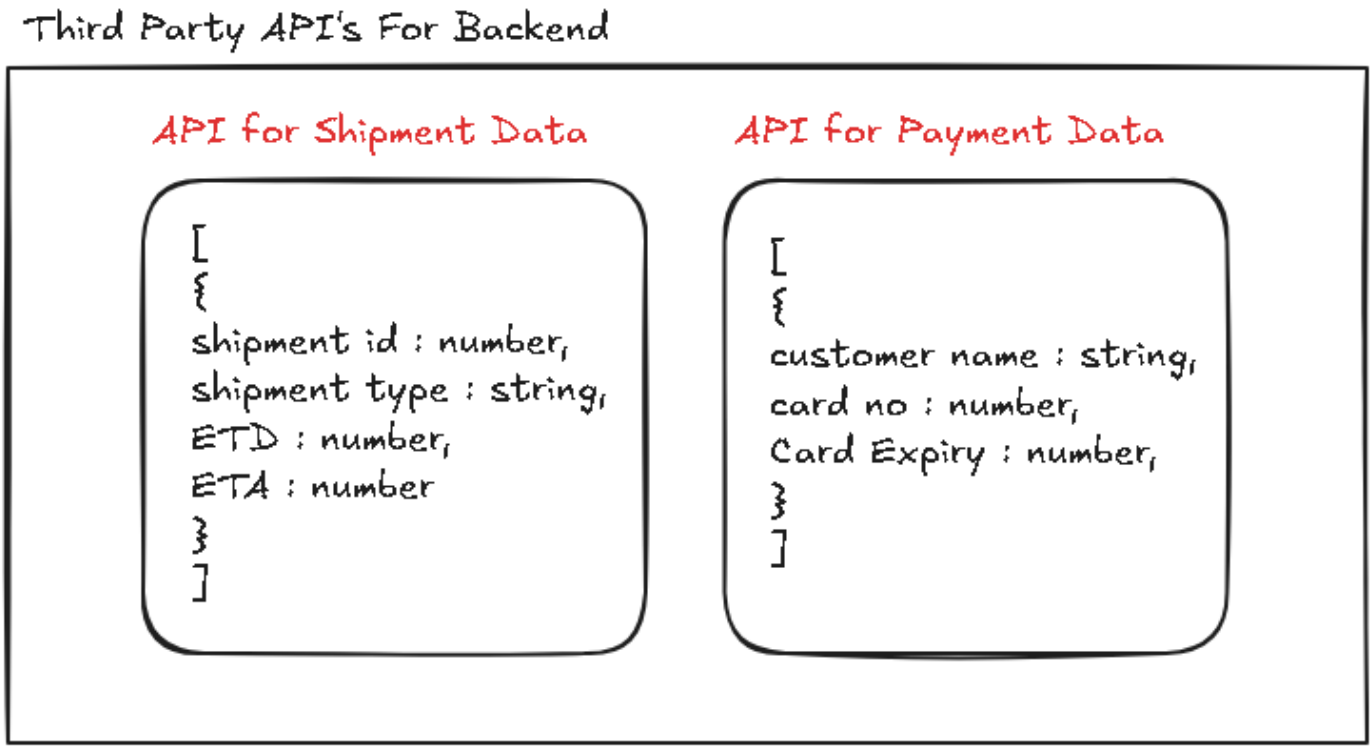
1. Product Data API: The product API includes all the necessary information of the products, which will renders on the web browser when the user wants to see the necessary details or to book the order for the product. The information includes:
    a. Product id
    b. Product name
    c. Product details
    d. Product price
    e. Product color
    f. Product video

2. Customer Data API: The customer API includes all the necessary information to get from the customer, to book the orders. These information includes:
    a. Customer name
    b. Customer address
    c. Customer contact number
    d. Customer email

3. Order Data API: The order API includes the necessary information of the products, which will be book by the use. These information includes:
    a. Order unique id
    b. Order type
    c. Order description
    d. Order date

c. THIRD PARTY API

The below diagram shows the necessary details of the third party API. Detail information is as under:



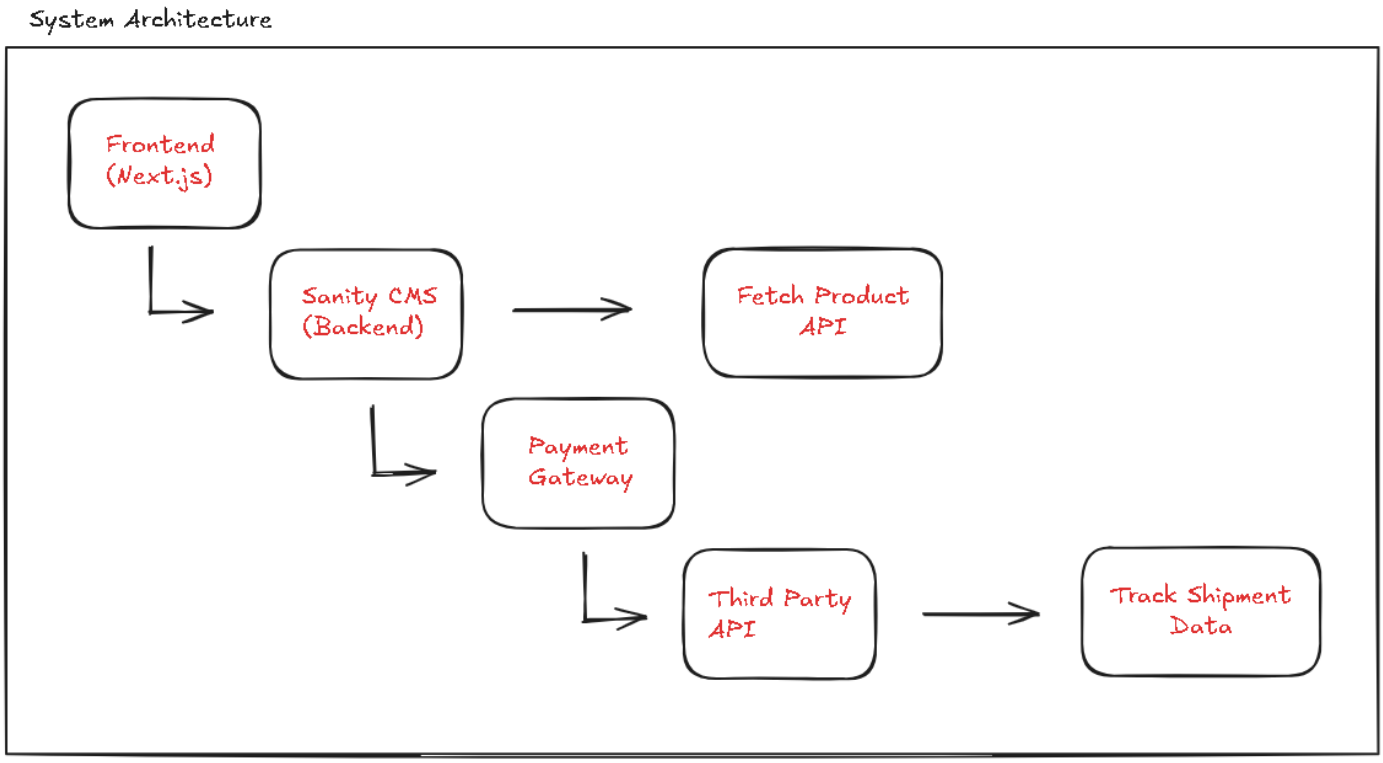
Key Points:

1. Payment Data API: This is the backend for the tracking data when the user booked the order successfully. This data includes the following information:
    a. Shipment id
    b. Shipment type
    c. Estimate departure date
    d. Estimate arrival date

2. Shipment Data API: This is the backend for the payment data. This data includes the following information:
    a. Customer name
    b. Customer card number
    c. Customer card expiry date

2. SYSTEM ARCHITECTURE

The following diagrams shows that how system will technically work. After visiting the website, user click onto the specific product according to their choices. From the backend product API, all details of the products will render on the product detail page. After confirmation of the product user clears all the payments. Once on the successful completion of the payment confirmation of the product will be send to the user and the third party API comes in process where the user can easily track the shipment.



3. API REQUIREMENTS PLAN

The ends point for the ***Rental E-Commerce*** of the necessary APIs used into the fully functioning site is as under:

a. For Products:
- End Point Name: /Categories
- Method: GET
- Info: Details about the products
- Response: As shown above in Product data

b. For Customer:
- End Point Name: /Customer
- Method: POST
- Info: Details about the customer
- Response: As shown above in Customer data

c. For Order:
- End Point Name: /Order
- Method: POST
- Info: Create a new unique order in Sanity CMS (Backend)
- Response: As shown above in Order data

d. For Shipment:
- End Point Name: /Shipment
- Method: GET
- Info: Track the shipment data
- Response: As shown above in Shipment data

4. SANITY SCHEMA FORMAT

The below diagrams illustrates the format of the sanity schema on how the data will be shown in the real word example

Sanity Schema Format

For Customer

```
export default {
    name: 'customer',
    type: 'document',
    title: 'customer data',
    fields: [
        {
            name: 'name',
            type: 'string',
            title: 'Name'
        },
        {
            name: 'address',
            type: 'string',
            title: 'Address'
        },
        {
            name: 'Cell NO',
            type: 'number',
            title: 'Cell Number'
        },
        {
            name: 'email',
            type: 'string',
            title: 'Email'
        }
    ]
}
```