

DAY 1: LAYING THE FOUNDATION FOR YOUR MARKETPLACE JOURNEY

In day 1 task of Hackathon we created a business proposal (Organic Spices) with the help of schema diagram and writing skills.

We showcased our idea on a paper with examples and discussed market value, our aim and issues we are going to target.

Hackathon Day 2: Planning the Technical Foundation

1. Technical Requirements:

Frontend Requirements:

- **User Interface (UI):** Clean, intuitive design for browsing and purchasing organic spices like Chili Powder, Cumin, Turmeric, etc.
- **Responsive Design:** The site will be accessible across desktop, tablet, and mobile.

PAGES:

There will be pages on our website with the titles;

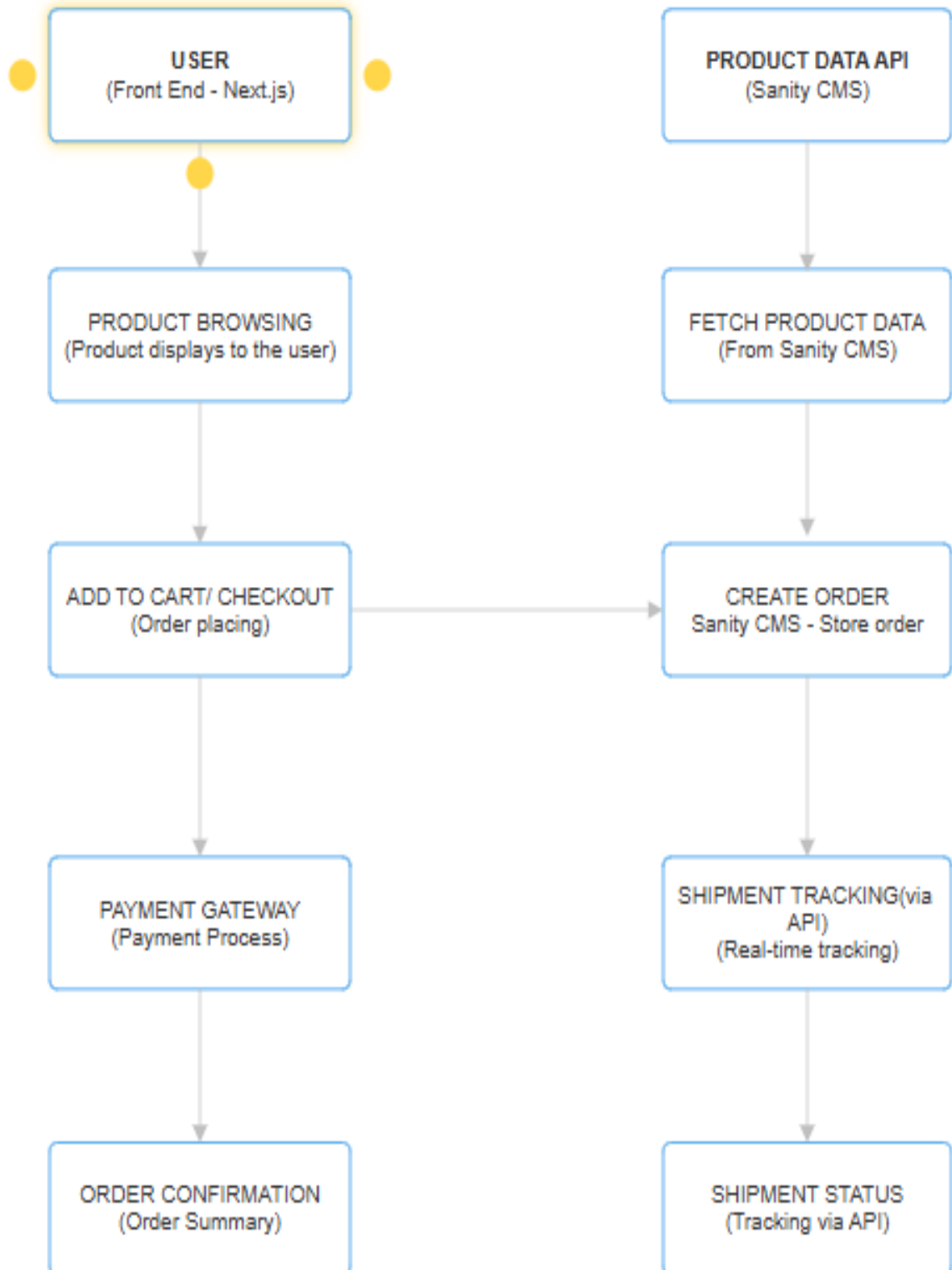
- **Home Page:** Display featured spices.
- **About Page:** Description & Details about us and our brand.
- **Product Listing Page:** List all available spices with filters for categories.
- **Cart:** View added items and their quantities.
- **Checkout:** Provide payment and shipping details.
- **Order Confirmation:** Show order status and shipment tracking.

Backend Requirements:

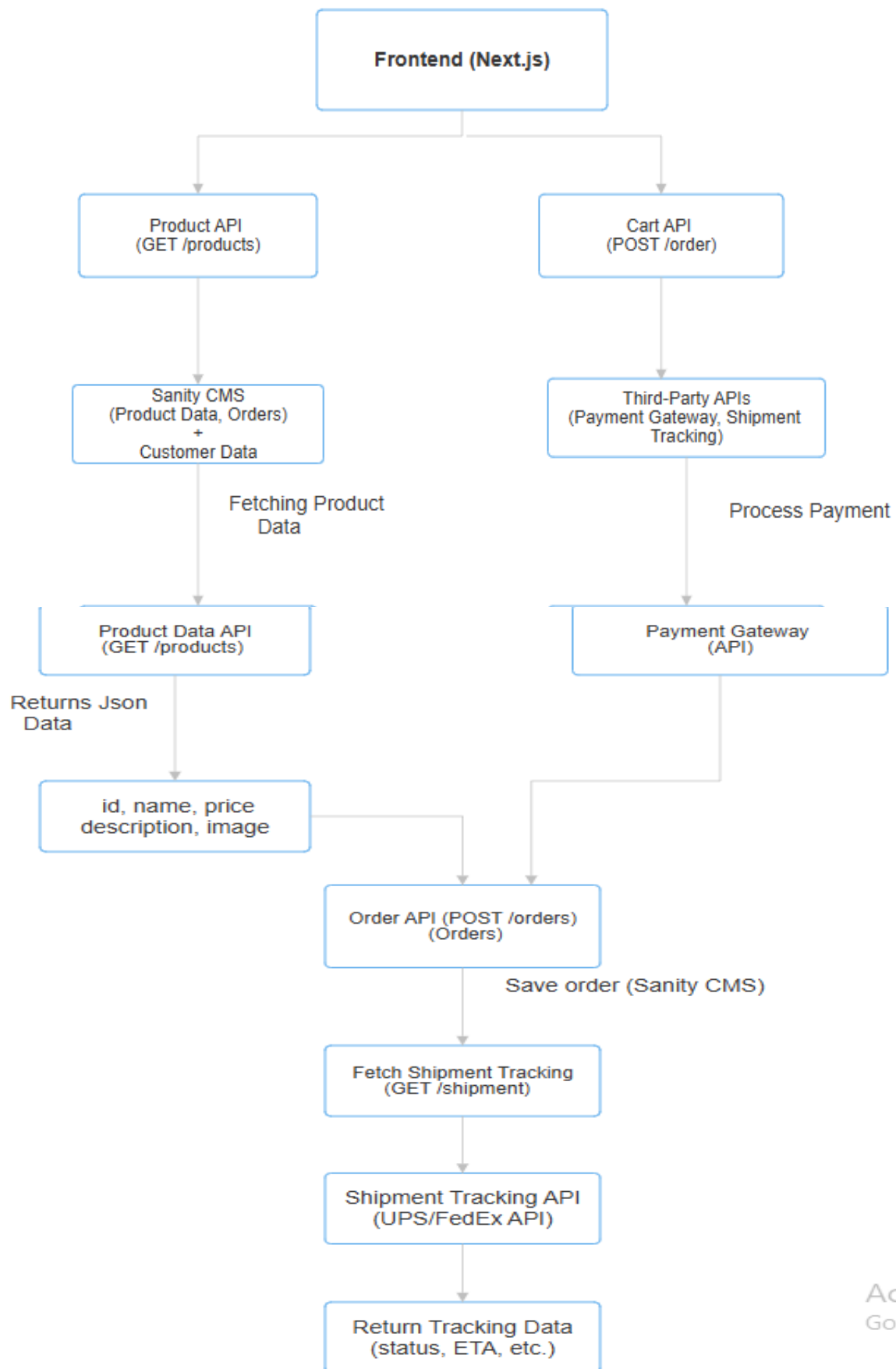
- **Next.js:**
- **Sanity CMS as Backend:** Manage products, customer data, and orders in a flexible, scalable backend.
- **Third-Party APIs:** For for handling transactions & real-time tracking updates.

System Architecture:

The architecture outlines how the different components (Frontend, Backend, Third-Party APIs) interact with each other. Below is a flowchart for your **Organic Spice Marketplace**:



FLOWCHART



Activ
Go to

Step-by-Step Flow Description:

1. User Browsing Products:

- **Action:** The user opens the website and views spices (like Chili Powder, Turmeric, etc.).
- **API:** The frontend sends a **GET request** to the **Product API** (/products) to fetch the list of available spices.
- **Sanity CMS:** Fetches the product data from the **Sanity CMS database**, which contains information such as product name, price, description, and image.
- **Return:** The product list is returned in **JSON** format and displayed on the frontend for the user to browse.

2. User Adds Product to Cart:

- **Action:** The user adds spices to their cart.
- **Cart API:** When the user proceeds to checkout, the frontend makes a **POST request** to the **Order API** (/orders), sending order details like product IDs and quantities.

3. Order Placement:

- **Action:** The user completes the checkout process and confirms the order.
- **Order API:** The frontend sends the order information (such as customer details, product data, and order total) to the **Sanity CMS**, which stores the order.
- **Sanity CMS:** Saves the order in the **orders** collection, ensuring it's linked to the customer and product data.

4. Payment Processing:

- **Action:** The user enters payment details and submits the payment.
- **Payment Gateway (Stripe API):** The frontend sends the **POST request** to the **Payment API** (/payment) with the payment data (credit card info, order ID, and amount).
- **Return:** Stripe processes the payment, and once successful, the payment confirmation is sent back to the frontend.

5. Shipment Tracking:

- **Action:** Once payment is confirmed, the user receives shipment tracking updates.
- **Shipment Tracking API (UPS/FedEx):** The frontend sends a **GET request** to the **Shipment Tracking API** (/shipment) with the order ID.
- **Return:** The tracking status, ETA, and other relevant details are fetched from **UPS/FedEx** and displayed on the frontend.

6. User Views Order Status:

- **Action:** The user can now see the **order status** and **estimated delivery time (ETA)** based on the data fetched from the shipment tracking API.

Sanity Schema Example:

```
export default
{
  name: 'product',
  type: 'document',
  fields: [ {
    name: 'name',
    type: 'string',
    title: 'Product Name'
  },
  {
    name: 'price',
    type: 'number',
    title: 'Price'
  }, {
    name: 'description',
    type: 'text',
    title: 'Description'
  }, {
    name: 'image',
    type: 'image',
    title: 'Product Image'
  } ]
};
```