# Hackathon Day 2: Planning the Technical Foundation

## *Planning the Technical Foundation for My Marketplace*

Day 2 focuses on creating a technical foundation that bridges the business goals outlined in Day 1 with the actual implementation of a scalable marketplace. This document includes system architecture, workflows, API requirements, and data management using Sanity CMS.

## Technical Requirements:

1. Frontend Requirements
2. Backend via Sanity CMS
3. Third-Party APIs Integration

### 1. Frontend Requirements:

- Responsive and user-friendly interface.
- Key pages include:
  - Home: Highlights featured products.
  - Product Listing: Displays categorized items.
  - Product Details: Detailed product view with "Add to Cart."
  - Cart: Summary of selected items.
  - Checkout: Secure payment and order placement.
  - Order Confirmation: Displays order details post-purchase.

### 2. Backend via Sanity CMS:

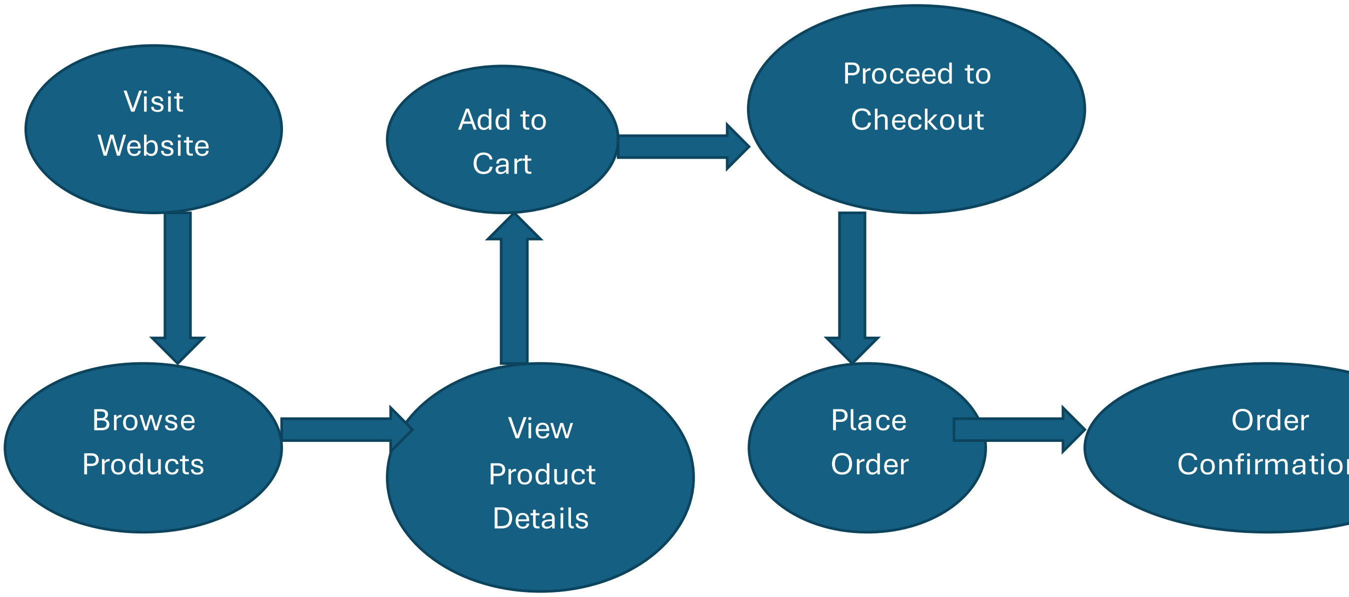
- Schemas to store:
  - Product details (ID, name, price, stock, etc.).
  - Customer data (name, email, phone, address).
  - Orders (product details, total amount, status).

### 3. Third-Party APIs Integration:

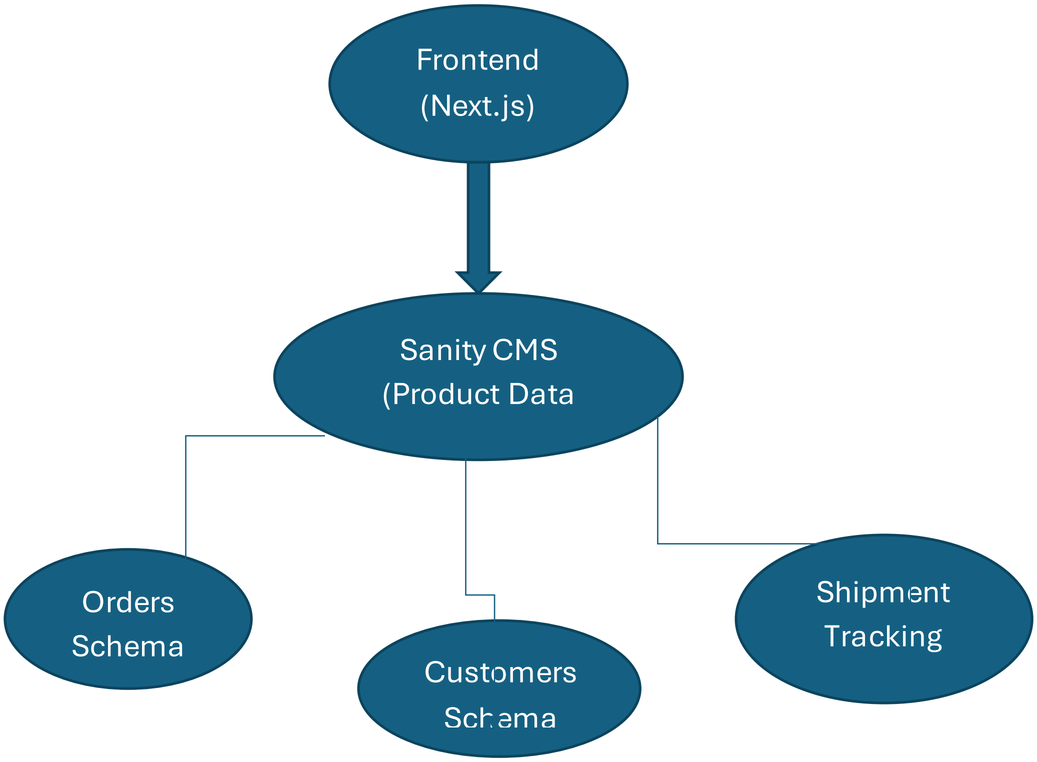
- Shipment Tracking API: For real-time delivery updates.
- Payment Gateway API: For secure transaction processing.

# Flowcharts:

1. **User Journey Flowchart:**
   This flowchart outlines the user's interaction with the marketplace:

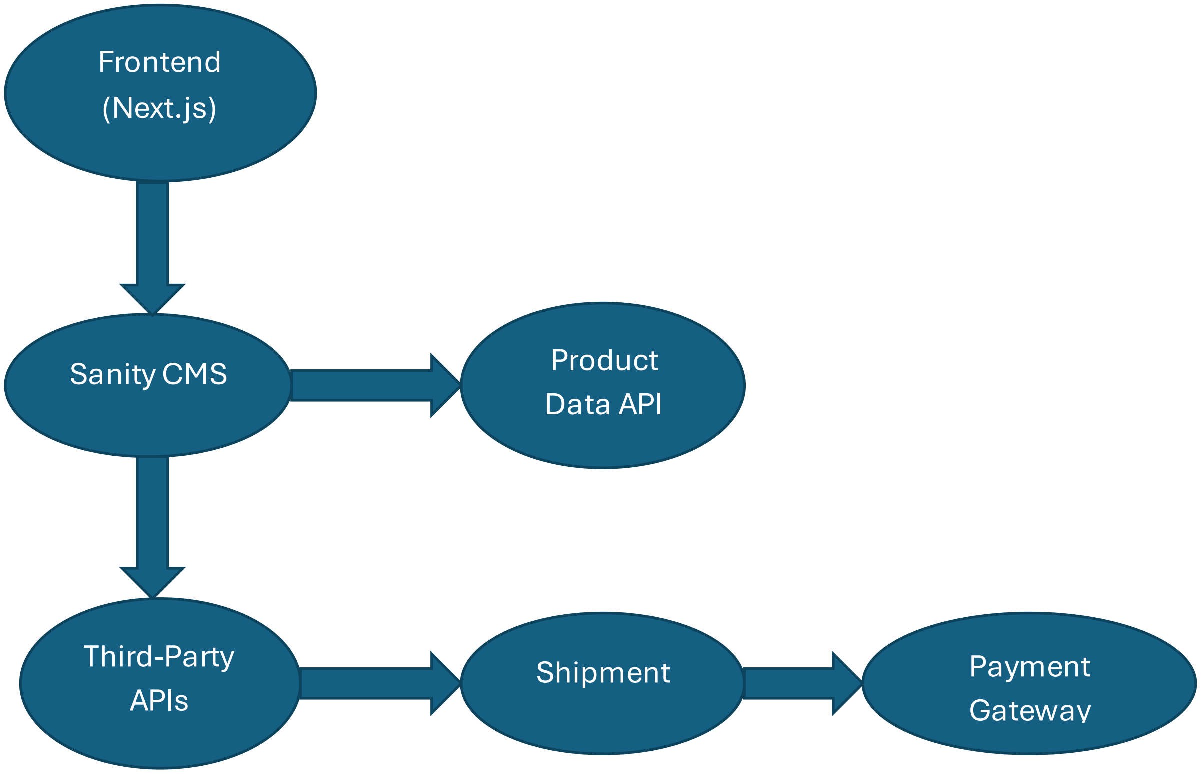


2. **System Data Flow Flowchart:**
   This flowchart represents how data flows between different components of the system:



## System Architecture

The following diagram explains how components interact:



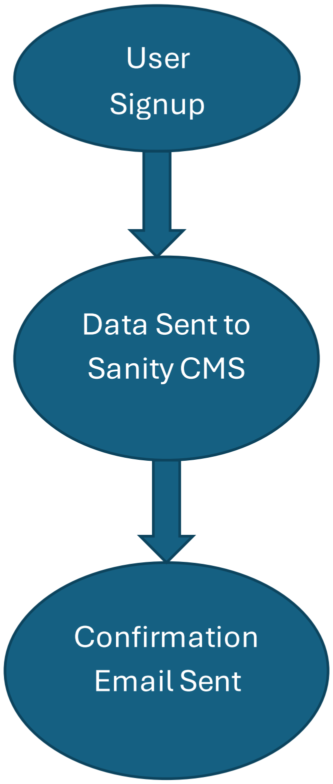
# Workflow and API Design

1. Workflows:

**A. User Registration Workflow:**

1. User enters registration details.
2. Data is sent to Sanity CMS.
3. A confirmation email is sent to the user.

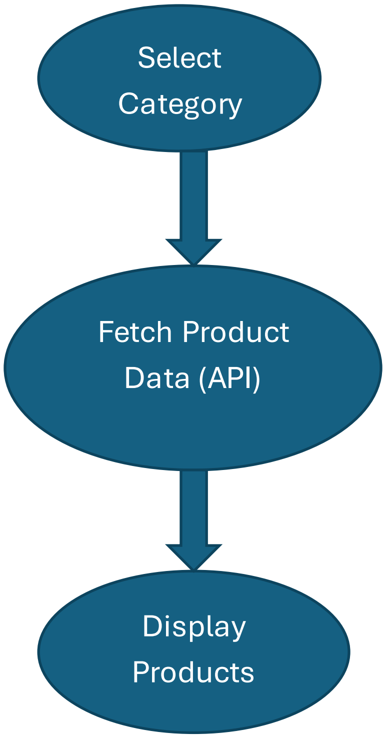
Flowchart:



**B. Product Browsing Workflow:**

1. User selects a product category.
2. Sanity CMS fetches products via API.
3. Products are displayed on the frontend.

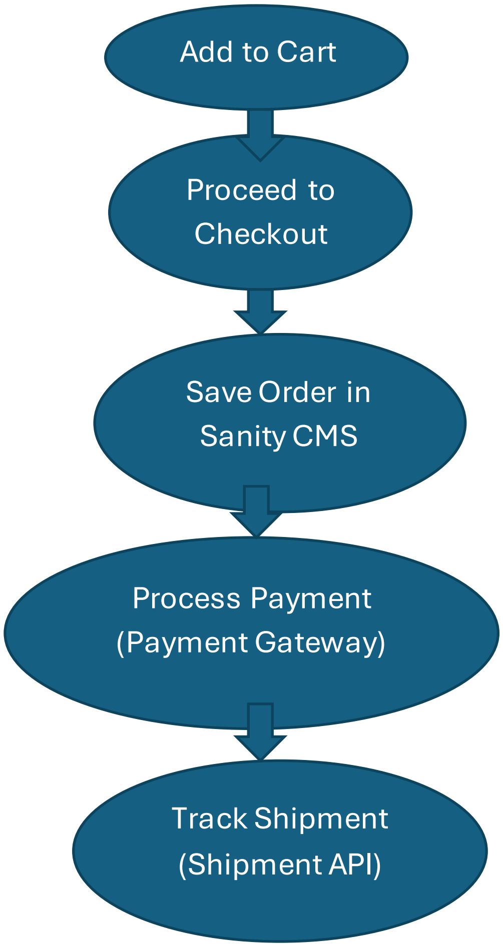
Flowchart:



**C. Order Placement Workflow:**

1. User adds products to the cart.
2. Proceeds to checkout.
3. Order details are saved in Sanity CMS.
4. Payment is processed via the Payment Gateway.
5. Shipment details are retrieved using the Shipment API.

Flowchart:


Add to Cart
Proceed to Checkout
Save Order in Sanity CMS
Process Payment (Payment Gateway)
Track Shipment (Shipment API)

# API Documentation

### 1. Product Management API:

- **Endpoint:** `/products`
- **Method:** GET
- **Description:** Fetches all available products.

### 2. Order Management API:

- **Endpoint:** `/orders`
- **Method:** POST
- **Description:** Saves a new order in Sanity CMS.

### 3. Shipment Tracking API:

- **Endpoint:** `/shipment`
- **Method:** GET
- **Description:** Tracks delivery status.

## Sanity CMS Schemas

### A. Product Schema:

```
export default {
  name: 'product',
  type: 'document',
  fields: [
    { name: 'name', type: 'string', title: 'Product Name' },
    { name: 'price', type: 'number', title: 'Price' },
    { name: 'stock', type: 'number', title: 'Stock Level' },
    { name: 'category', type: 'string', title: 'Category' },
  ],
};
```

### B. Order Schema:

```
export default {
  name: 'order',
  type: 'document',
  fields: [
    { name: 'customerId', type: 'string', title: 'Customer ID' },
    { name: 'products', type: 'array', of: [{ type: 'reference', to: [{ type: 'product' }] }] },
    { name: 'totalPrice', type: 'number', title: 'Total Price' },
    { name: 'status', type: 'string', title: 'Order Status' },
  ],
};
```

## Key Takeaways

- A complete technical foundation aligned with business goals.
- Clear workflows and API documentation for seamless integration.
- Visualized system architecture for better understanding.
- Sanity CMS schemas for efficient data management.