Day 2

Technical Planning for Nike Shoe Marketplace

Recap of Day 1

On Day 1, we established the foundation for our Nike shoe marketplace by defining the following key elements:

- Marketplace Type: E-commerce platform focused on Nike shoes.
- Marketplace Problem: Addressing issues such as product authenticity, availability, and customer trust in online shoe shopping.
- Target Audience: Sneaker enthusiasts, Athletes seeking performance footwear.

Day 2 Activities:

1. Define Technical Requirements

Based on the business goals and marketplace structure established on Day 1, we will outline the technical requirements for the Nike shoe marketplace. This includes frontend requirements, backend management using Sanity CMS, and integration with third-party APIs.

Frontend Requirements:

❖ User -Friendly Interface for Browsing Products

Implement a visually appealing layout that highlights Nike branding and product imagery also Include a search bar with autocomplete suggestions and advanced filtering options (e.g., by size, color, style, and price).

❖ Responsive Design for Mobile and Desktop Users:

Ensure the website is fully responsive, providing a seamless experience on smartphones, tablets, and desktops and Optimize loading times and performance for mobile users.

❖ Essential Pages:

- ✓ **Home Page**: Featured products, promotional banners for new releases, and popular categories.
- ✓ **Product Listing Page:** Display products in a grid format with options to sort by popularity, price, and new arrivals.
- ✓ **Product Details Page:** High-resolution images, detailed descriptions, size guides, customer reviews, and related products.
- ✓ **Cart Page**: Summary of selected items with options to modify quantities, remove items, and view total costs.
- ✓ **Checkout Page:** Secure checkout process with options for guest checkout and account creation.
- ✓ **Order Confirmation Page**: Confirmation message with order details, estimated delivery date, and tracking information.

Sanity CMS as Backend

Sanity CMS will serve as the backbone of our Nike shoe marketplace, allowing us to efficiently manage all essential data related to products, customers, and orders. By using Sanity, we can ensure that our marketplace runs smoothly and meets the needs of our users.

❖ Product Data Management:

This includes details like product names, descriptions, prices, sizes, colors, and high-resolution images. We will create structured data models (schemas) that define how product information is organized, making it easy to update and retrieve data as needed.

❖ Customer Details/Order Records:

Sanity will also keep track of customer information, such as names, email addresses, and order history. All order transactions will be recorded in Sanity, allowing us to track purchases, manage inventory, and handle customer inquiries regarding their orders. This feature ensures that we can provide accurate order confirmations and updates to customers.

❖ Benefits of Using Sanity CMS:

- ✓ Sanity allows us to easily adapt and change our data structures as our business needs evolve.
- ✓ Real-Time changes made in Sanity are reflected instantly on the marketplace, ensuring that customers always see the most up-to-date information.
- ✓ Sanity provides an intuitive interface for our team to manage content without needing extensive technical knowledge.
- ✓ Sanity CMS will be a powerful tool for managing the backend of our Nike shoe marketplace, ensuring that we can efficiently handle product data, customer details, and order records while aligning with our business goals.

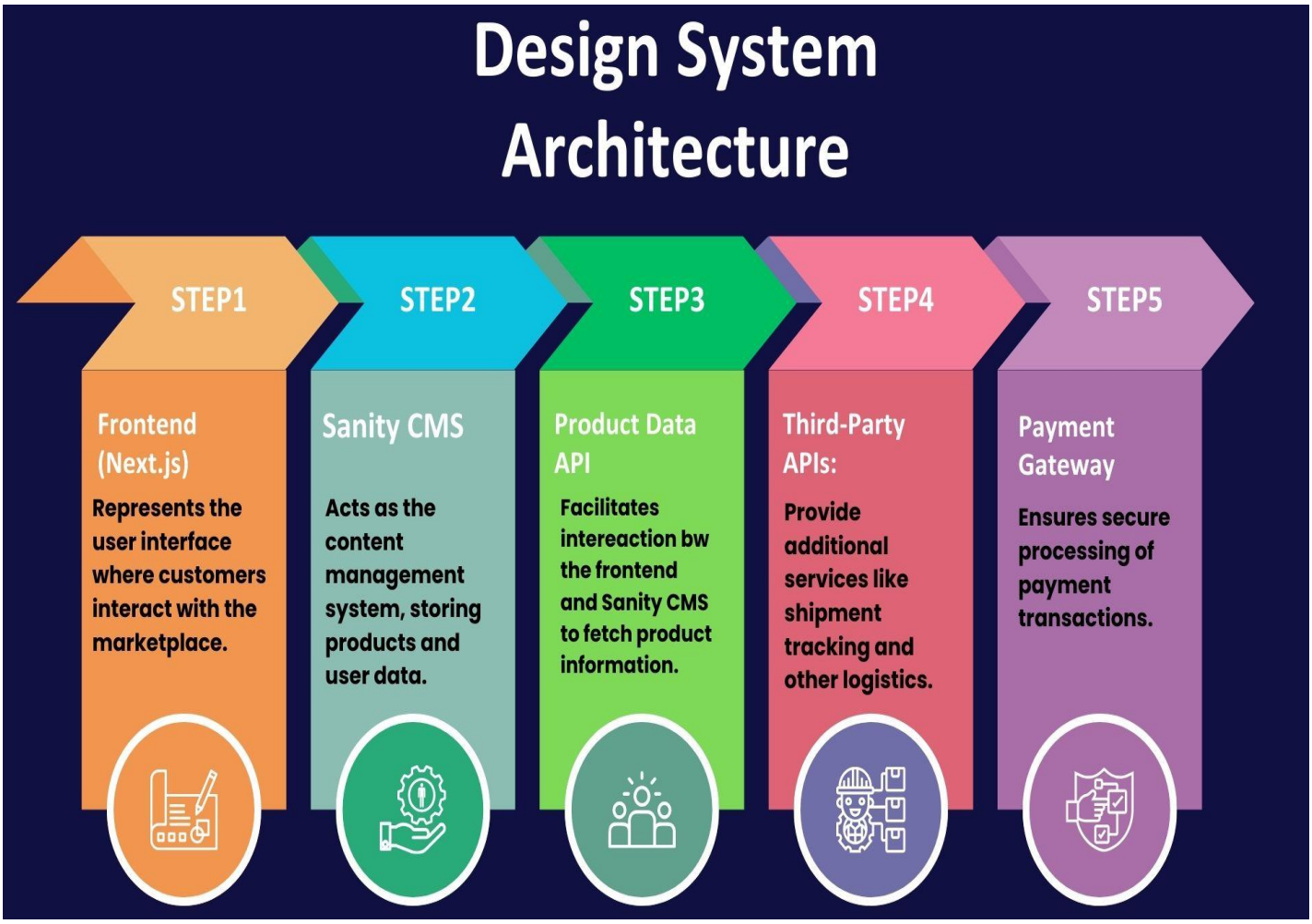
Third-Party APIs

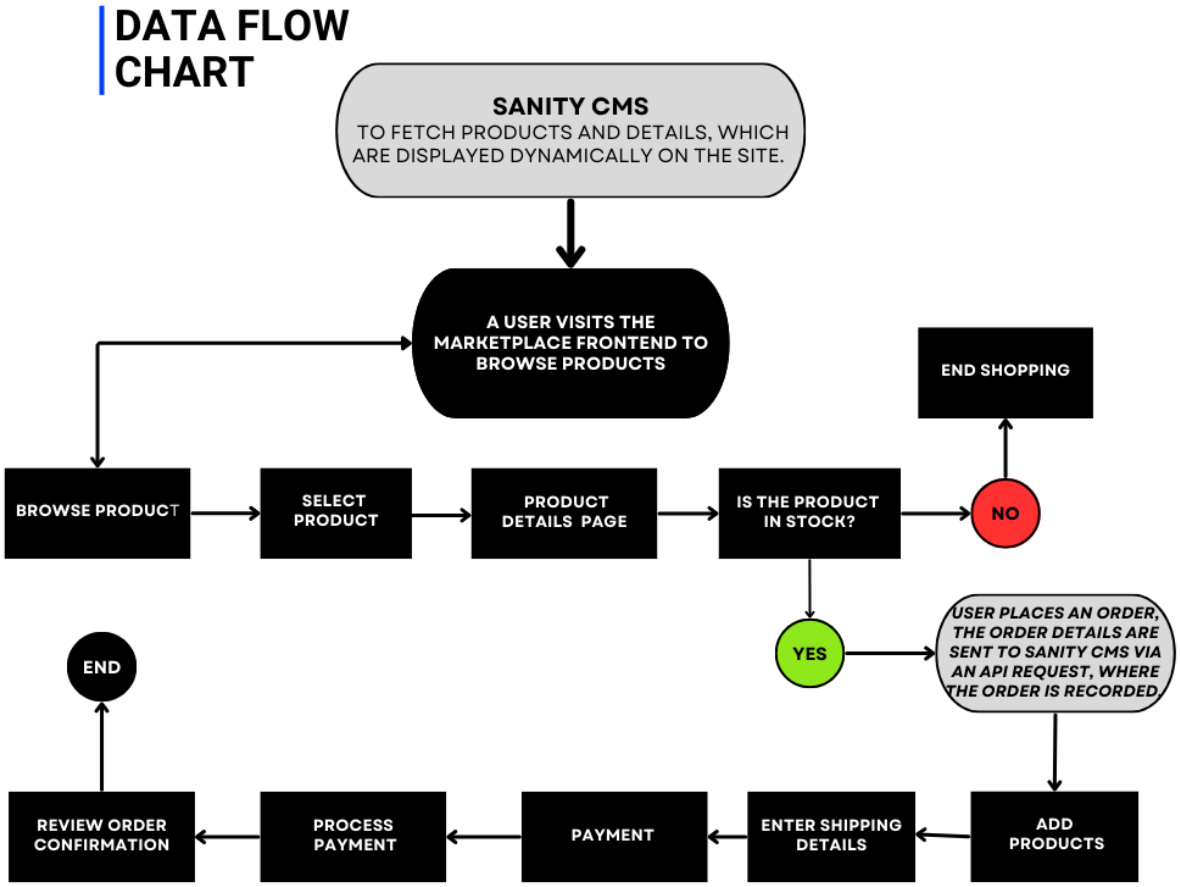
Integrating third-party APIs is crucial for enhancing our Nike shoe marketplace. These APIs will provide essential services like shipment tracking and secure payment processing, improving the overall shopping experience for our customers.

1. **Shipment Tracking APIs**:
    - These APIs will allow customers to track their orders in real-time, showing shipping status, estimated delivery dates, and tracking numbers directly on the Order Confirmation Page and in their accounts.
2. **Payment Gateway APIs:**
    - We will use secure payment gateway APIs to enable various payment options, such as credit/debit cards and digital wallets. This ensures safe transactions and a smooth checkout process for customers.
3. **Additional Services:**
    - We may also integrate APIs for inventory management and customer support, helping us manage stock levels and enhance customer engagement.

2. Design System Architecture

Below is a simplified and easy-to-understand representation of how the components of our Nike shoe marketplace interact. This diagram illustrates the flow of data between the frontend, Sanity CMS, third-party APIs, and the payment gateway.





3. Plan APIS Requirement:

- **Endpoint Name:** /products
- **Method:** GET
- **Description:** GET a list of all available products from the Sanity CMS.

Response Example:


```
[
 {
 "id": 1,
 "name": "Product A",
 "price": 100,
 "stock": 20,
 "image": "https://example.com/product-a.jpg"
 },
 {"id": 2,
```

"name": "Product B",
"price": 150,
"stock": 15,
"image": "https://example.com/product-b.jpg"}]

Create a New Order:

Endpoint Name: /orders

Method: POST

Description: Add a new order to the Sanity CMS, including customer information, product details, and payment status.

Payload Example:

{
"customer": {
"name": "John Doe",
"email": "john.doe@example.com"
},
"products": [
{ "id": 1, "quantity": 2 },
{ "id": 2, "quantity": 1 }
],
"paymentStatus": "Paid"
}

Response Example:

{
"orderId": 12345,
"status": "Order Placed",
"totalAmount": 350
}

Track Order Shipment

Endpoint Name: /shipment

Method: GET

Description: Fetch shipment tracking details for an order via a third-party API.

Response Example:

{
"shipmentId": "SH123456",
"orderId": 12345,
"status": "In Transit",
"expectedDelivery": "2025-01-20"}

4.Write Technical Documentation

Introduction

This document outlines the technical requirements and API endpoints for the e-commerce marketplace, which includes the following components:

1. Frontend: A user-friendly interface built to help users browse, purchase, and track products easily.

2. Backend (Sanity CMS): A content management system used for managing product data, customer details, and order records.

3. Third-Party APIs: APIs for shipment tracking and payment processing.

System Architecture Overview

➢ Frontend Requirements

Simple and intuitive design with a sidebar for product categories, search bar, sections like Flash Sale, Best Selling Products, Limited Time Offers.

• **Responsiveness:** The website is optimized for all devices, providing an excellent experience on mobile and desktop.

• **Essential Pages:**

- ✓ **Home Page:** Displays product categories and sections (Flash Sale, Best Sellers, etc.).
- ✓ **Products Pages:** Separate pages for Men, Women, Accessories, and Sale items.
- ✓ **Product Detail Page**: Displays detailed reviews, or images of individual products.
- ✓ **Cart Page:** Allows us to select quantities, view total price, and make changes to their cart.
- ✓ **Checkout Page**: Displays item amounts, shipping price, total price, and payment method

---

Sanity CMS as Backend

● **Product Data Management**: Manages all product information, including names, descriptions, prices, sizes, colors, and images.

● **Customer Details:** Stores customer information such as names, emails, and order history for personalized shopping.

● **Order Records**: Tracks orders and ensures that stock and order details are accurately managed.

● **Schemas:** Defined for product data, customer data, order data, shipment data, and delivery data.

Third-Party API Integrations

● **Shipment Tracking API:** Provides real-time shipment tracking details such as status, expected delivery date, and tracking number.

● **Payment Gateway API:** Facilitates secure transactions through credit/debit cards and digital wallets.

API Requirements and Endpoints

1. Fetch All Products

● **Endpoint Name:** /products

● **Method**: GET

● **Description:** Fetch a list of all available products from Sanity CMS.

Response Example:

```
[
 {
 "id": 1,
 "name": "Product A",
 "price": 100,
 "stock": 20,
 "image": "https://example.com/product-a.jpg"
```

```
 },
 {
 "id": 2,
 "name": "Product B",
 "price": 150,
 "stock": 15,
 "image": "https://example.com/product-b.jpg"
 }
]
```


2. Create a New Order

● **Endpoint Name:** /orders

● **Method:** POST

● **Description:** Add a new order to Sanity CMS, including customer information, product details, and payment status.

Payload Example:


```
{
 "customer": {
 "name": "John Doe",
 "email": "john.doe@example.com"
 },
 "products": [
 { "id": 1, "quantity": 2 },
 { "id": 2, "quantity": 1 }
 ],
 "paymentStatus": "Paid"
}
```


Response Example:


```
{
 "orderId": 12345,
 "status": "Order Placed",
 "totalAmount": 350
}
```


3. Track Order Shipment

Endpoint Name: /shipment

Method: GET

Description: Fetch shipment tracking details for an order via a third-party API.

Response Example:

```
{
 "shipmentId": "SH123456",
 "orderId": 12345,
 "status": "In Transit",
 "expectedDelivery": "2025-01-20"
}
```

Data Flow and Interactions

1. Fetching Products:

- The frontend makes a GET request to the /products endpoint to fetch product data from Sanity CMS.
- The data is returned in JSON format, which is then displayed on the frontend.

2. Placing Orders:

- A customer adds products to their cart and proceeds to checkout.
- The frontend sends a POST request to the /orders endpoint with customer details, selected products, and payment status.
- The order is saved in Sanity CMS, and a response is returned with the order ID and total amount.

3. Tracking Shipments:

- Once an order is placed, the frontend makes a GET request to the /shipment endpoint to track the order's shipment status via a third-party API.
- The shipment tracking information is returned and displayed to the user in real-time.