Marketplace Technical Foundation Of an E-commerce:

1. System Architecture Overview

The **System Architecture** document describes the design and interaction of components within the marketplace. It provides an overview of how the **Frontend**, **Sanity CMS**, and **Third-Party APIs** work together.

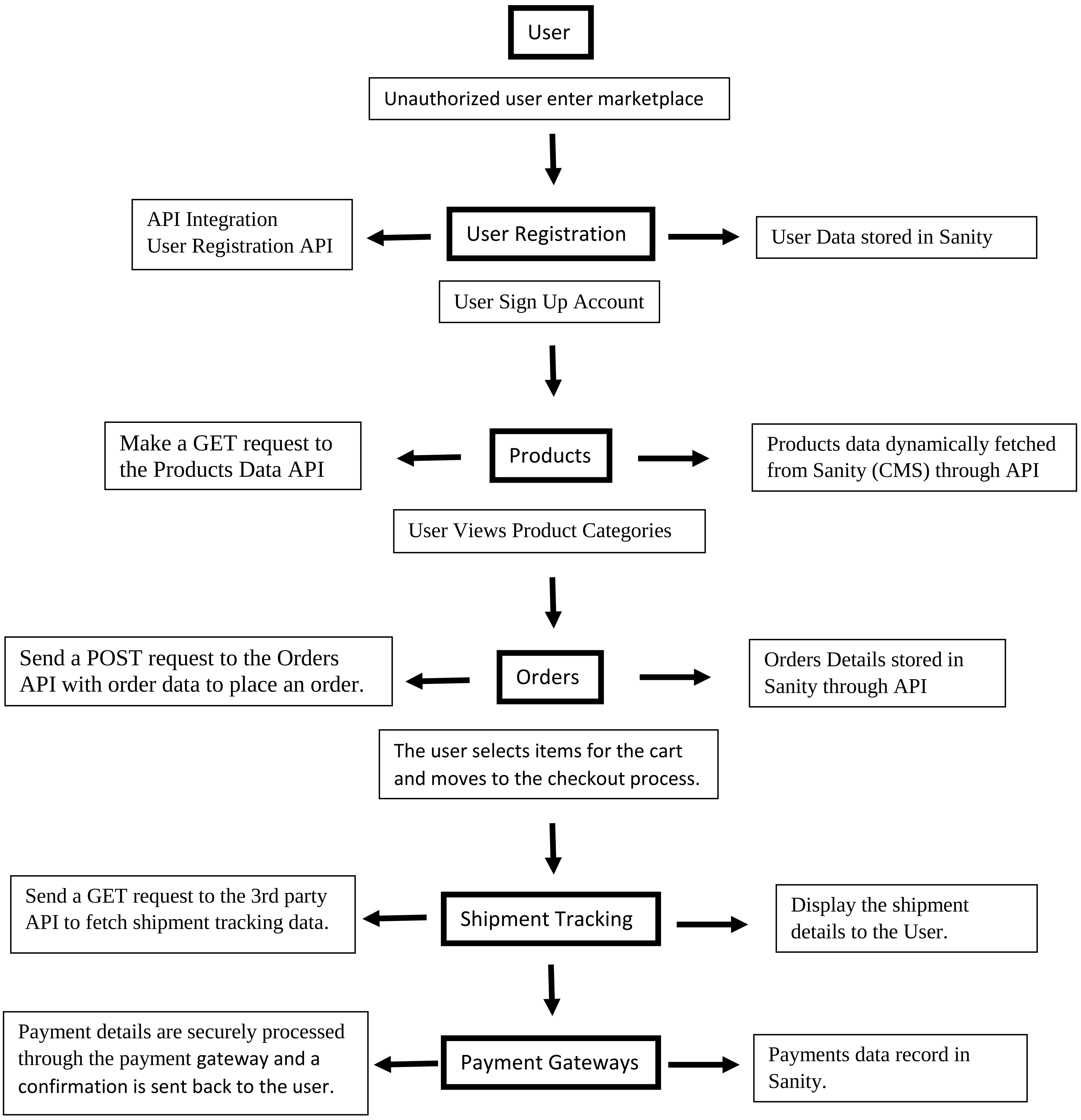
System Architecture Diagram:

Frontend	Backend	Third Party API's
• ***Next.js*** Next.js is used in the frontend to build the visible part of a website, making fast and user-friendly website.	• ***Sanity (CMS)*** Sanity CMS is used to manage and organize website content easily, making it flexible and connected to the frontend seamlessly.	• ***ShipEngine*** ShipEngine is used to handle shipping and tracking, making order delivery smooth and hassle-free.
• ***Tailwind CSS*** Tailwind CSS is used to style a website, making it look attractive and responsive.		• ***Stripe*** Stripe is used to handle online payments securely, making transactions easy and reliable.

Components Breakdown:

1. **Frontend (Next.js & Tailwind CSS)**
   - Manages the user interface, including browsing, cart, and checkout.
2. **Sanity CMS (Backend)**
   - Stores product, order, and user data, and provides access to this data through APIs.
3. **Third-Party APIs**
   - Handles payments (via **Stripe**) and shipment tracking (via **ShipEngine**).

Browse Products And Order Shipment


User
Unauthorized user enter marketplace
API Integration
User Registration API
User Registration
User Data stored in Sanity
User Sign Up Account
Make a GET request to the Products Data API
Products
Products data dynamically fetched from Sanity (CMS) through API
User Views Product Categories
Send a POST request to the Orders API with order data to place an order.
Orders
Orders Details stored in Sanity through API
The user selects items for the cart and moves to the checkout process.
Send a GET request to the 3rd party API to fetch shipment tracking data.
Shipment Tracking
Display the shipment details to the User.
Payment details are securely processed through the payment gateway and a confirmation is sent back to the user.
Payment Gateways
Payments data record in Sanity.

2. Overall Design and Interaction Between Components

---

1. Unauthorized User Enters Marketplace

- **Component Involved**: Frontend (Next.js)
  - **Interaction**: When a user visits the e-commerce platform, they first interact with the **frontend**. At this point, the user is not logged in and can browse products without registering.

---

2. User Registration & Sign-Up

- **Component Involved**: Frontend, Backend (Sanity CMS)
  - **Interaction**: When the user decides to make a purchase, they must register. This involves entering their details such as name, email, and password.
    - The **frontend** sends the registration data to the **backend** (Sanity CMS) via a **POST request** to store the user's information (User Registration API).
    - **Sanity CMS** stores the user data and returns a confirmation to the frontend.

---

3. User Views Products

- **Component Involved**: Frontend, Sanity CMS
  - **Interaction**: After registration, the user can browse products.
    - The **frontend** sends a **GET request** to the **Products API** from **Sanity CMS** to fetch product details like product name, description, price, stock availability, and images.
    - The data fetched dynamically from **Sanity CMS** is displayed on the frontend.

---

4. User Adds Products to Cart

- **Component Involved**: Frontend (Cart Management)
  - **Interaction**: The user selects items and adds them to the shopping cart.
    - The **frontend** maintains the cart state, where product IDs and quantities are stored temporarily in the browser or application state (e.g., React Context or Redux).

---

5. User Places an Order

- **Component Involved**: Frontend, Backend (Sanity CMS), Orders API

    - **Interaction**: Once the user confirms their order, the **frontend** sends a **POST request** to the **Orders API** to place the order.
        - The **Sanity CMS** receives the order details and stores them as a document, including the customer's details, the selected items, and the total amount.
        - The order data is stored in **Sanity CMS**, and the frontend receives a confirmation message to proceed with payment.

---

6. Shipment Tracking

- **Component Involved**: Frontend, Third-Party API (ShipEngine)
    - **Interaction**: After the order is confirmed, the user may want to track their shipment.
        - The **frontend** sends a **GET request** to the **third-party API (ShipEngine)** to fetch real-time shipment tracking information.
        - The **ShipEngine API** returns details such as the tracking number, current location, status, and expected delivery date.
        - The **frontend** displays this information to the user in the **shipment tracking interface**.

---

7. Payment Processing

- **Component Involved**: Frontend, Third-Party API (Payment Gateway such as Stripe)
    - **Interaction**: The user proceeds to payment after reviewing the order and shipment details.
        - The **frontend** sends payment details (e.g., credit card information) to a **third-party API (Stripe)** for payment processing via a **POST request**.
        - The **Stripe API** processes the payment securely and sends a response (success/failure) back to the frontend.
        - On successful payment, a confirmation message is displayed, and the **payment data** is stored in **Sanity CMS** for record-keeping.

---

3. API Specification Document: Marketplace API

This document outlines the API endpoints, methods, payloads, and expected responses used in the marketplace. These endpoints are essential for fetching product data, creating orders, and tracking shipments.

3: Plan API Requirements

General E-commerce:

EndPoint Name	Method	Purpose	Schema	Response

/products	GET	Fetch all products data from Sanity.	{
			productID: string,
			name: string,
			description: string,
			productImage: string,
			price: string,
			stock: string,
			};
				{
				"productID": 1001,
				"name": "CamfyNest 3-Seater Sofa",
				"description": "A stylish and compact 3- Seater sofa with a modern design, plush cushions, and durable fabric, perfect for small living spaces.",
				"productImage": "https://imageURL",
				"price": "$50",
				"stock": "100"
				}

/orders	POST	Create a new order in Sanity.	orderID: string,
			orderDate:string,
			totalAmount:string,
			paymentStatus:string,
			customerInfo: {
			customerID: string,
			customerName:string,
			customerEmail:string,
			customerPhone:string,
			customerAddress:string,
			customerCountry:string,
			customerCity:string,
			},
			productsInfo: Products
				{
				"orderID": "09876",
				"orderDate": "17-1-2025",
				"totalAmount": "$50",
				"paymentStatus": "Active",
				"customerInfo": {
				"customerID": "1234",
				"customerName": "Zija",
				"customerEmail": "zijayaseen15@gmail.com",
				"customerPhone": "923160426977",
				"customerAddress": "House#ABC",
				"customerCountry": "Pakistan",
				"customerCity": "Karachi"
				},
				"productsInfo": [products]
				}

/shipment	GET	Track order via Third Party API.	{
			shipmentID: string,
			orderID: string,
			status: string,
			expectedDeliveryDate: string,
			}
				{
				shipmentID: PKHTRE7890,
				orderID: 09876,
				status: Active,
				expectedDeliveryDate: 20-1-2025,
				}