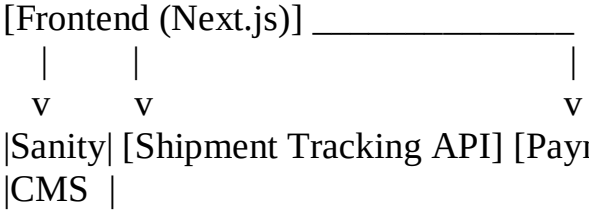
Day 2:

Marketplace Technical Foundation - Women's Fashion E-Commerce

1. System Architecture Overview

Diagram



Component Roles

- **Frontend (Next.js):**
  - Delivers a user-friendly interface for browsing, purchasing, and order tracking.
  - Implements responsive design for seamless access across devices.
- **Sanity CMS:**
  - Backend system managing product data, customer/user details, and order records.
- **Shipment Tracking API:**
  - Supplies real-time order shipment tracking details.
  - Includes status updates, delivery ETA, and tracking numbers.
  - Ex: ship engine
- **Payment Gateway (Stripe):**
  - Processes secure payments and ensures payment status updates are reflected in Sanity CMS.

---

2. Key Workflows

1. User Registration

- **Step 1:** User registers on the frontend using **Next-Auth**.
- **Step 2:** Frontend sends user details to Sanity CMS for storage.

- **Step 3:** Sanity CMS confirms successful registration.

2. Product Browsing

- **Step 1:** User navigates to the homepage or product page.
- **Step 2:** Frontend fetches product data using Sanity's GROQ queries.
- **Step 3:** Sanity CMS responds with product details (name, price, description, image).
- **Step 4:** Frontend dynamically displays the products.

3. Order Placement

- **Step 1:** User adds items to the cart and proceeds to checkout.
- **Step 2:** Frontend sends order details (customer info, products, payment status) to Sanity CMS.
- **Step 3:** Sanity CMS stores the order and sends a confirmation.
- **Step 4:** Payment Gateway processes the transaction and updates the payment status in Sanity CMS.

4. Shipment Tracking

- **Step 1:** User views the order tracking page.
- **Step 2:** Frontend requests shipment status from the Shipment Tracking API.
- **Step 3:** Shipment API provides updates (status, ETA, tracking number).
- **Step 4:** Frontend displays the shipment details.

---

3. API Requirements

General API Endpoints

Endpoint	Method	Purpose	Request Payload/Query	Response Example
`/product`	`GET`	Fetch all available products	None	`[{"id":1, "name":"Dress", "price":100}]`

Endpoint	Method	Purpose	Request Payload/Query	Response Example
		from Sanity.		
/orders	POST	Create a new order in Sanity.	{customerInfo, products, paymentStatus}	{"orderId":123, "status":"Success"}
/shipment	GET	Fetch shipment details via Shipment API.	{orderId}	{"shipmentId":456, "status":"In Transit", "ETA":"2 days"}

---

4. Technical Documentation

4.1 System Architecture Document

- **Overview:**
  Integration of a Next.js frontend, Sanity CMS backend, and third-party APIs for shipment tracking and secure payments.
- **Components:**
  - **Frontend (Next.js):** Manages user interactions and dynamically displays data.
  - **Sanity CMS:** Acts as a database for managing marketplace data.
  - **Shipment Tracking API:** Fetches real-time updates on orders.
  - **Payment Gateway (Stripe):** Secures and processes online transactions.

4.2 Workflow Diagram

- **User Registration:**
  User → Frontend → Sanity CMS → Confirmation to user.

- **Order Placement:**
  User → Frontend → Sanity CMS → Payment Gateway → Update in Sanity CMS.
- **Shipment Tracking:**
  Frontend → Shipment API → Real-time status → Display to user.

4.3 Data Schema Design

Entities:

1. **Product:**
   - Fields: ID, Name, Price, Stock Level, Image.
2. **Order:**
   - Fields: Order ID, Product ID, Quantity, Total Amount.
3. **Customer:**
   - Fields: Customer ID, Name, Contact Info, Address.
4. **Shipment:**
   - Fields: Shipment ID, Status, Delivery Date, Tracking Number.