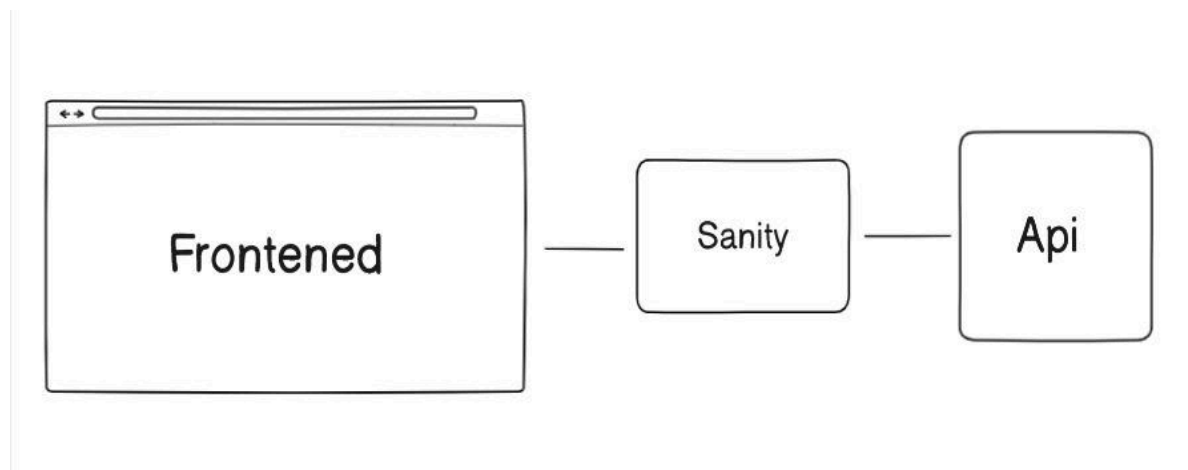


# Marketplace Technical Foundation

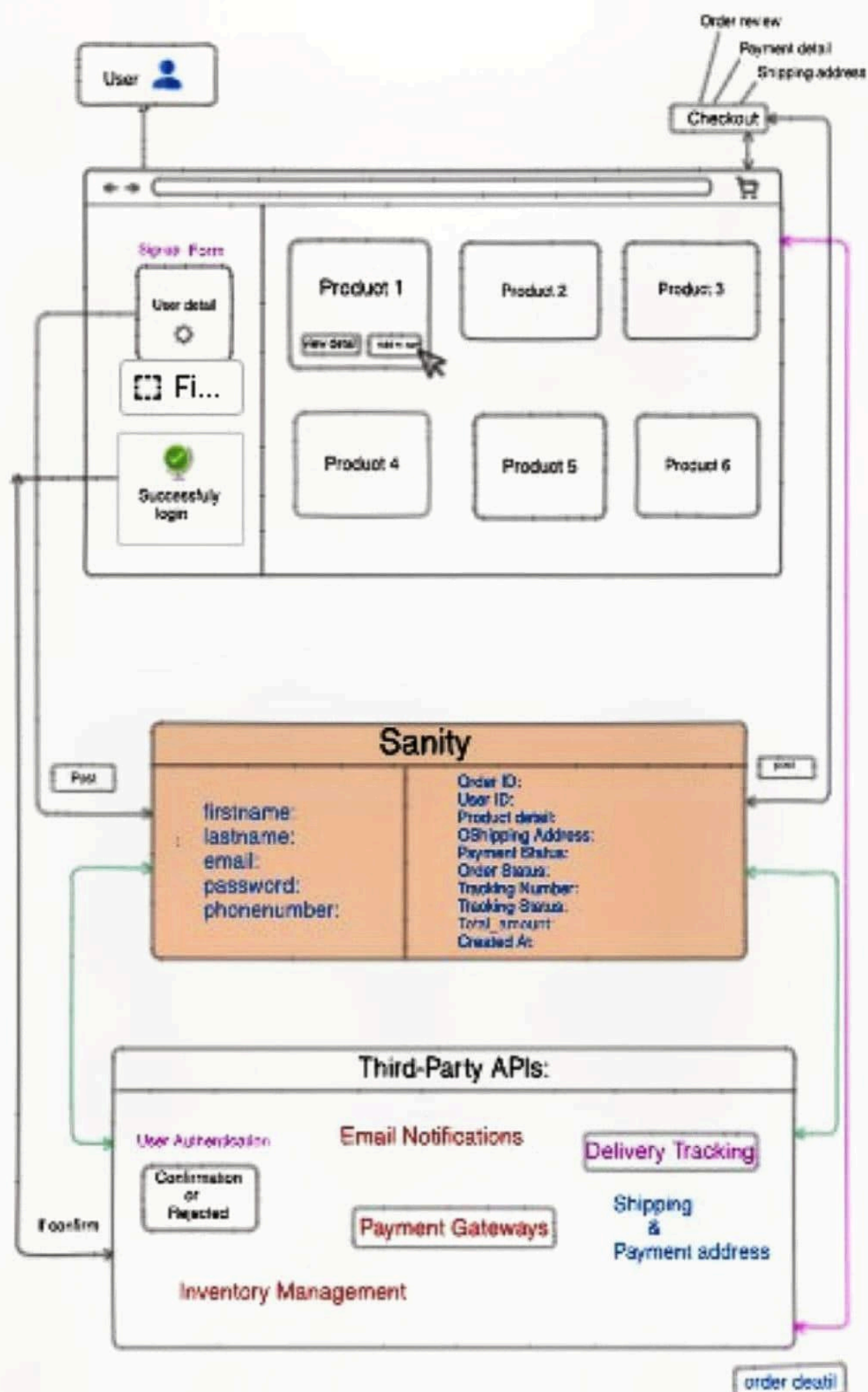
## DesignAura Marketplace

The DesignAura Marketplace is a cutting-edge furniture eCommerce platform that combines modern design with seamless functionality. The platform is built to provide a smooth, intuitive shopping experience, with a responsive Frontend for browsing products and a robust Sanity CMS to manage product listings, customer data, and orders. Additionally, integration with Third-Party APIs ensures secure payment processing and real-time shipment tracking, creating a scalable, efficient, and user-friendly marketplace for furniture shopping.

### System Architecture Overview



### System Architecture Diagram :



# 1. Frontend (User Interface)

The frontend is the face of the platform, enabling users to browse, purchase, and interact with products seamlessly.

## Key Pages:

- **Homepage:** Displays featured products, promotions, and a search bar for easy navigation.
- **Product Pages:** Displays product details, including images, descriptions, prices, and availability.
- **Cart:** A summary of selected items for purchase, allowing users to modify quantities or remove items.
- **Checkout:** Collects billing and shipping information, calculates taxes and delivery, and processes payments.
- **Login/Signup:** Allows customers to register, log in, and manage their accounts.
- **Contact Us:** A page for users to reach out with queries, feedback, or support.
- **Blog:** A section for publishing content like furniture tips, trends, and news.

# 2. Sanity CMS (Backend & Content Management)

Sanity CMS serves as the backend, storing and managing product data, customer info, and orders.

## Key Responsibilities:

- **Product Management:** Store product names, images, descriptions, and stock levels in Product Schema.
- **Order Management:** Store order details, including customer info, product purchased, and payment status.
- **Customer Data:** Manage customer information (e.g., name, email, address) for order processing and communication.
- **Content Management:** Manage dynamic content for Homepage, Blog, and Contact Us pages.

### **Tech Stack:**

Sanity CMS: Content management and data storage.

Sanity Studio: Admin interface for managing data.

API Integrations: Fetch product, order, and customer data.

## **3. Third-Party APIs (External Integrations)**

Third-Party APIs integrate additional services like payment processing and shipment tracking.

### **Key Responsibilities:**

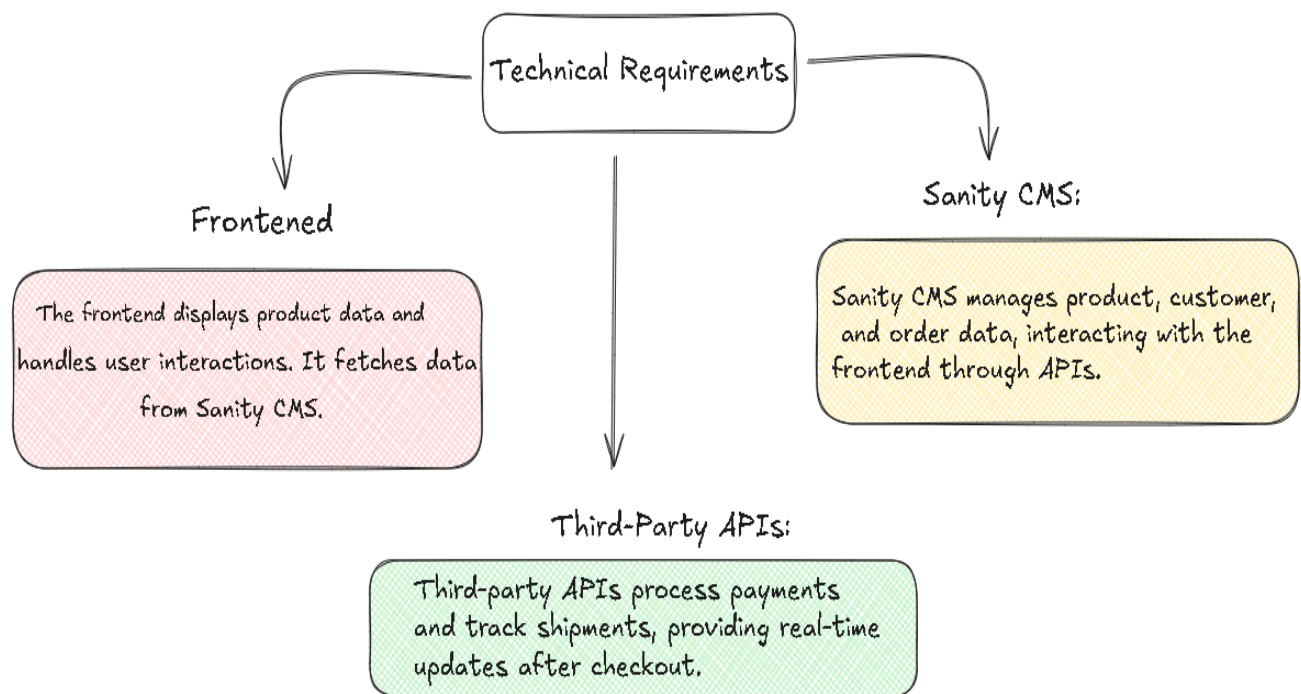
**Payment Gateway API:** Securely handles payments and updates the order status in Sanity CMS.

**Shipment Tracking API:** Provides real-time shipment updates after the order is confirmed.

### Tech Stack:

RESTful APIs: For seamless data exchange between the frontend, Sanity CMS, and third-party services.

## Technical Workflow Diagram:



### Key Workflow:

## **1. User Registration & Login**

- *Registration: Users sign up with email/password or third-party login (e.g., Google). Data is stored in Sanity CMS.*
- *Login: Users log in to access their profile, cart, and order history.*

## **2. Browsing Products**

- *Products are fetched from Sanity CMS and displayed by category (Essential, Deluxe, Premium).*

## **3. Adding Products to Cart**

- *Guest Users: Cart is saved in local storage.*
- *Logged-In Users: Cart data is stored in Sanity CMS and synced across devices.*

## **4. Placing an Order**

- *At checkout, system checks stock and processes payment through the Payment API (e.g., Stripe).*
- *Order details (products, payment, shipping) are saved in Sanity CMS.*

## **5. Tracking Orders**

- *Logged-In Users: View past orders and track shipments via Shipment API.*
- *Guest Users: Track orders using Order ID and email.*

## **Apl Endpoints :**

A	B	C	D
Endpoints	Method	Purpose	Response
/api/register	POST	Register a new user	{"success":true, "userid": "123"}
/api/login	POST	Logs in a user	{"success":true, "token": "abcd"}
/api/products	GET	Fetch all product detail.	{"id": 5, "name": "Sofa", "stock": 5, "price": 300}
/api/cart	POST	Adds item to the cart	{"success":true, "cart": [{"productID": 2, "quantity": 3}]}
/api/checkout	POST	Processes Payment and creates order	{"orderId": "Order345", "status": "Payment Successful", "total": 200}
/api/shipment	POST	Add shipping detail	{"shipment": 24}
/api/orders	GET	order-status	{"orders": [{"orderId": "complete", "status": "Delivered"}]}

**Documentation Author : Aliza Moin**  
**Saturday Slot ( 2 to 5)**  
**Task Given By : Sir Ameen Alam**