**Foundation of General E-Commerce Marketplace:**

Overview:

This **Marketplace E-commerce Website** is a modern platform offering a wide range of stylish and customizable chairs for homes and offices. Built with Next.js for a fast and responsive user experience, it integrates Sanity CMS to manage product details, orders, and customer data efficiently. Secure payment gateways and real-time shipment tracking ensure a seamless and reliable shopping journey, while customization options allow customers to personalize their chairs to suit their needs.

**Table of content:**

01 Frontend: Nextjs

02 Content Management System (CMS): Sanity

03 API Integration: Mockapi

04 Shipment: ShipEng

05 Payment: Stripe

**01 Frontend Nextjs:**

The frontend, built with **Next.js**, serves as the user-facing part of your website where customers browse products, place orders, and track shipments. It dynamically fetches data from Sanity CMS using APIs to display updated product details, stock levels, and order statuses. The responsive design ensures smooth navigation on desktop and mobile devices, providing features like search, filters, cart management, and order checkout for an engaging user experience.

**02 Sanity (CMS):**

Sanity CMS is the backend content management system where all your product, customer, order, and shipment data is stored and managed. Custom schemas define how data is organized, such as products (name, price, stock) and orders (customer info, products, payment status). It integrates seamlessly with the frontend via APIs, ensuring real-time updates when you add or edit content, like adjusting stock levels or updating product descriptions.

**03 API Integration:**

APIs act as the bridge connecting your frontend and backend systems. Product details are fetched via APIs from Sanity CMS, while customer orders and payments are sent to the backend. Third-party APIs are used for shipment tracking and payment processing. For instance, APIs provide live shipment updates to customers and validate payments securely, ensuring smooth communication between all components of your platform.

**04 Shipment:**

The shipment system handles the delivery process, starting from order confirmation to final delivery. Once a customer places an order, a shipment ID is generated, and the order is assigned to the appropriate delivery zone. Real-time shipment tracking data is fetched from a third-party API, allowing customers to monitor the status of their order (e.g., "In Transit" or "Delivered") and view the expected delivery date directly on the website.

**05 Payment:**

The shipment system handles the delivery process, starting from order confirmation to final delivery. Once a customer places an order, a shipment ID is generated, and the order is assigned to the appropriate delivery zone. Real-time shipment tracking data is fetched from a third-party API, allowing customers to monitor the status of their order (e.g., "In Transit" or "Delivered") and view the expected delivery date directly on the website.

**01 Define Technical Requirements:**

1. **Frontend Nextjs:**

**Purpose:**

The **frontend** is the visual and interactive layer of your website, where customers interact with your platform. It ensures a seamless user experience, allowing customers to browse, select, and purchase chairs effortlessly.

Key responsibilities:

* Displaying product catalogs and details dynamically.
* Providing intuitive navigation and filtering options.
* Handling user actions like adding products to the cart, placing orders, and tracking shipments.

### ****How It Works:****

1. **Built Using Next.js:**
   * Next.js is used to create a fast, SEO-friendly, and responsive interface.
   * Features like server-side rendering (SSR) ensure the website loads quickly and displays updated product data.
2. **Dynamic Data Fetching:**
   * The frontend fetches data (e.g., product details, stock levels) from **Sanity CMS** using APIs and displays it to the user.
3. **Customer Actions:**
   * Customers browse chairs through product categories and use search or filter options to narrow their choices.
   * When a product is selected, its detailed information (e.g., price, dimensions, stock) is fetched via APIs.
   * Customers add products to their cart, and the cart is updated dynamically without refreshing the page.
4. **Order Placement:**
   * During checkout, the frontend collects customer details, payment information, and order preferences, sending them to the backend via APIs.
5. **Real-Time Updates:**
   * The frontend displays real-time order statuses, shipment tracking, and estimated delivery times by fetching data from APIs connected to Sanity CMS and third-party services.
6. **Sanity (CMS):**

### ****Purpose:****

The **Sanity CMS** serves as the content management backend, where all website data (products, customers, orders, and shipments) is stored, organized, and updated. It acts as the single source of truth for the website.

Key responsibilities:

* Managing and storing data for products, orders, customers, and shipments.
* Providing real-time updates to the frontend through APIs.
* Offering a scalable and flexible backend for future growth.

### ****How It Works:****

1. **Data Storage:**
   * **Products:** Stores all product-related data, such as names, prices, categories, stock levels, descriptions, and images.
   * **Orders:** Tracks order details, including customer information, purchased products, payment status, and timestamps.
   * **Shipments:** Logs shipment details like tracking IDs, statuses (e.g., "In Transit"), and delivery dates.
2. **Custom Schemas:**
   * Sanity CMS uses custom schemas to define the structure of stored data.
3. **Data Management:**
   * Administrators can add, edit, or delete product details directly in the Sanity CMS dashboard.
   * Real-time updates ensure that changes made in the CMS are instantly reflected on the website via APIs.
4. **APIs for Frontend:**
   * Sanity CMS provides APIs to fetch data dynamically. For instance:
     1. /products: Fetch product details for the catalog.
     2. /orders: Store order details.
     3. /shipments: Retrieve shipment tracking information.
5. **API Integration:**

### ****Purpose:****

**APIs** act as a bridge between the frontend, Sanity CMS, and external services like payment gateways and shipment tracking systems. They facilitate data exchange and ensure that the website operates efficiently and securely.

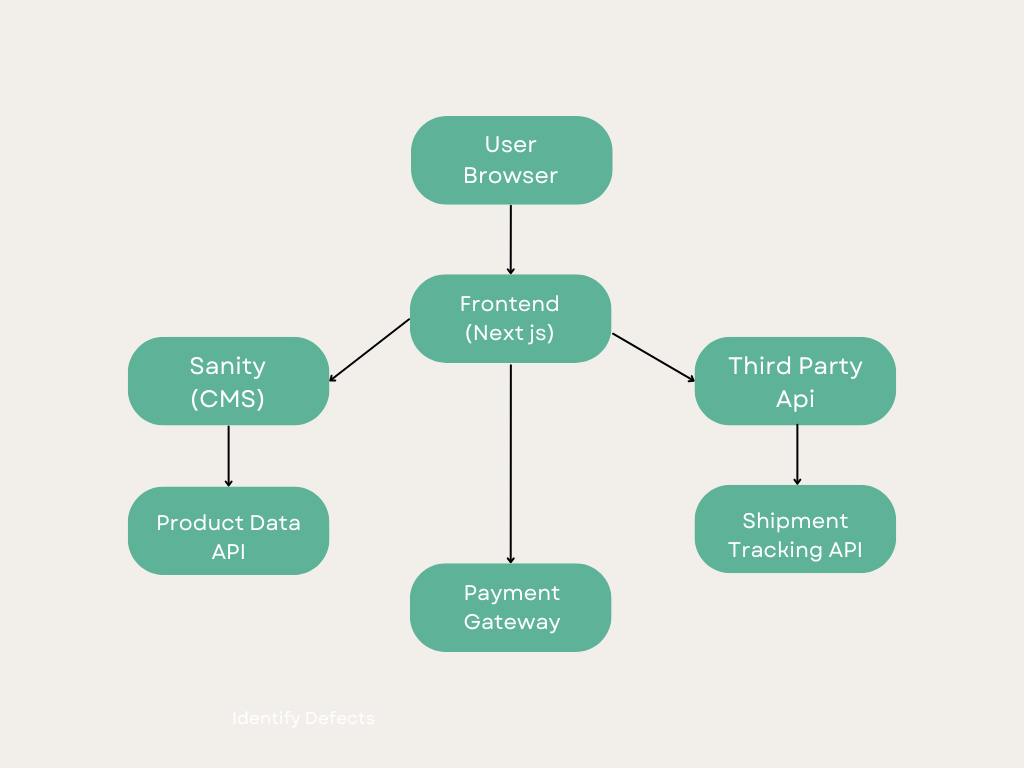
Key responsibilities:

* Retrieving data from Sanity CMS for display on the frontend.
* Sending user input (e.g., orders, payments) from the frontend to the backend.
* Communicating with third-party services for payment processing and shipment tracking.

### ****How It Works:****

1. **Product Data API:**
   * **Endpoint:** /products
   * **Method:** GET
   * **Purpose:** Fetch product details from Sanity CMS to display on the website.
2. **Order Management API:**
   * **Endpoint:** /orders
   * **Method:** POST
   * **Purpose:** Save order details (e.g., customer info, product IDs, total cost) in Sanity CMS.
3. **Shipment Tracking API:**
   * **Endpoint:** /shipment
   * **Method:** GET
   * **Purpose:** Retrieve real-time tracking information for orders.
4. **Payment Gateway API:**
   * **Integration:** Stripe or PayPal.
   * **Workflow:** Processes payment securely, validates transactions, and updates the payment status in the system.

**02 Design System Architecture:**



**03 API Endpoints:**

User Management API:

**Create a New User (POST)**

* **Endpoint:** /api/users
* **Method:** POST
* **Description:** Send a request to register a new user.

**Get All Users (GET)**

* **Endpoint:** /api/users
* **Method:** GET
* **Description:** Retrieve details of all users.

**Get User by ID (GET)**

* **Endpoint:** /api/users/{id}
* **Method:** GET
* **Description:** Fetch the details of a specific user by user ID.

**Update User Details (PUT)**

* **Endpoint:** /api/users/{id}
* **Method:** PUT
* **Description:** Update the information of a user.

**Delete a User (DELETE)**

* **Endpoint:** /api/users/{id}
* **Method:** DELETE
* **Description:** Delete a specific user by user ID.

Product Management API:

**Create a New Product (POST)**

* **Endpoint:** /api/products
* **Method:** POST
* **Description:** Create a new product.

**Get All Products (GET)**

* **Endpoint:** /api/products
* **Method:** GET
* **Description:** Retrieve all products.

**Get Product by ID (GET)**

* **Endpoint:** /api/products/{id}
* **Method:** GET
* **Description:** Fetch the details of a specific product by product ID.

**Update Product Details (PUT)**

* **Endpoint:** /api/products/{id}
* **Method:** PUT
* **Description:** Update the details of a product.

**Delete a Product (DELETE)**

* **Endpoint:** /api/products/{id}
* **Method:** DELETE
* **Description:** Delete a specific product by product ID.

**Conclusion:**

The Foundation of a General E-Commerce Marketplace is a robust and scalable platform designed to provide customers with a seamless shopping experience while offering administrators powerful tools for managing content, orders, and logistics. With a Next.js-powered frontend, Sanity CMS for dynamic data management, and integrations with APIs for payment processing and shipment tracking, the system ensures efficiency, security, and reliability. Its modular design allows for easy customization and expansion, making it an ideal solution for businesses looking to deliver modern and user-friendly e-commerce experiences. This architecture exemplifies a synergy between advanced technologies and user-centric features, paving the way for success in the competitive e-commerce landscape.