**Day 2: Building the Technical Backbone**

**Designing My Marketplace’s Technical Framework**

On the second day of my journey, I focused on outlining the technical architecture needed to bring my furniture marketplace to life. The goal is to ensure that the system is robust, user-friendly, and capable of seamless operations.

**How the System Works**

**High-Level Overview:**

My marketplace architecture is designed to deliver a smooth experience for users while ensuring backend efficiency. Here are the key components:

* **Frontend (Next.js/React.js):** Responsible for creating an interactive, responsive user interface for browsing, managing carts, and processing checkouts.
* **Sanity CMS:** Acts as the database for managing:
  + Product details
  + Customer information
  + Orders
* **Third-Party APIs:** Used for integrating:
  + Payment gateways like Stripe and PayPal.
  + Shipment tracking systems for real-time updates.

**System Flow:**

* The frontend fetches product data and handles user interactions, including checkout and payments.
* Sanity CMS manages and stores core data such as product listings, customer details, and order records.
* APIs handle real-time functionalities like payments and tracking shipments.

**My API Blueprint**

Here’s how I structured key API endpoints:

* **/products (GET):** Retrieves available products with details like name, price, stock, and category.
* **/orders (POST):** Creates new orders by processing customer ID, selected products, and total amounts.
* **/customers (POST):** Adds new customers with their name, contact details, and address.
* **/shipment (GET):** Tracks shipment status, providing updates like delivery estimates and current locations.

**Visualizing the Workflow**

To better understand how my system will function, I created workflows for:

1. Browsing and viewing products.
2. Placing and processing orders.
3. Tracking shipments in real-time.

**Structuring Data with Sanity CMS**

I’ve developed specific schemas to organize data efficiently:

* **Product Schema:** Contains fields like product name, price, stock quantity, category, and image URL.
* **Order Schema:** Includes order ID, customer reference, product list, total amount, and order status.
* **Customer Schema:** Covers customer name, email, phone number, and address details.

**Achievements and Learnings**

By the end of Day 2, I’ve:

1. Designed a clear system architecture aligned with my business objectives.
2. Defined core API endpoints to manage essential functionalities like orders, customers, and shipments.
3. Visualized key workflows for an enhanced user experience.
4. Developed ready-to-use schemas in Sanity CMS for seamless backend operations.

This technical blueprint will serve as the backbone of my marketplace, ensuring a smooth transition to the development phase. I’m excited to take the next steps and make this vision a reality!