

# Key Workflows for our Market place Builder:

**Objective:** Allow users to explore trending, featured, and categorized products effortlessly.

**Steps:**

## 1. Home Page Display:

- User visits the website and clicks "Log In/Sign Up"
- They enter credentials or use social login options (Google, GitHub, etc.)
- Upon success, they access their homepage.
- Users land on a visually engaging homepage showcasing **Trending Products**, **Latest Collections**, and **Featured Products** with high-quality images and detailed descriptions.
- A **dynamic Sales Banner** grabs attention with limited-time deals.

## 2. Category Navigation:

- Users click on a category (e.g., Living Room, Bedroom).
- **Frontend (Next.js)** sends an API request to Sanity CMS to fetch product data specific to the selected category.

## 3. Product Details Page:

- Users click on a product to view details, including images, price, availability, and tags and categories.
- A **"Wishlist" button** allows saving the product for later purchases.
- Related products are dynamically fetched and displayed to keep users engaged.

---

## 2. Cart Management & Checkout

**Objective:** Ensure a smooth cart and checkout experience.

**Steps:**

### 1. Adding to Cart:

- Users add products to their cart from the product or category pages.

- The cart dynamically updates with product quantity and total cost.

## 2. Cart Review:

- Users review their cart with options to modify quantities or remove all item from the cart or remove just one item.
- A **real-time total calculation** (including taxes and discounts) is displayed.

## 3. Checkout Process:

- User clicks "Checkout," triggering the creation of a **Stripe Checkout Session**.
- The backend sends product details (IDs, quantities, prices, and totals) to the Stripe API to generate the session.
- User is redirected to Stripe's secure checkout page.
- User enters payment details (e.g., credit card, Apple Pay).
- Once payment is confirmed, the system updates the order status in Sanity CMS.

---

## 4. Shipment Tracking

- After placing an order, users track shipments in real-time using ShipEngine.
- Once the order is confirmed, the backend sends shipping details (customer address, product weight/dimensions) to ShipEngine API.
- ShipEngine generates a tracking number and updates the shipment status.
- The frontend (Next.js) fetches shipment data from the ShipEngine API.
- Displays the real-time status (e.g., "In Transit," "Delivered") on the Shipment Tracking page.

- User enters the tracking number on the Shipment Tracking page to view the latest status.

**By Wania Azam**