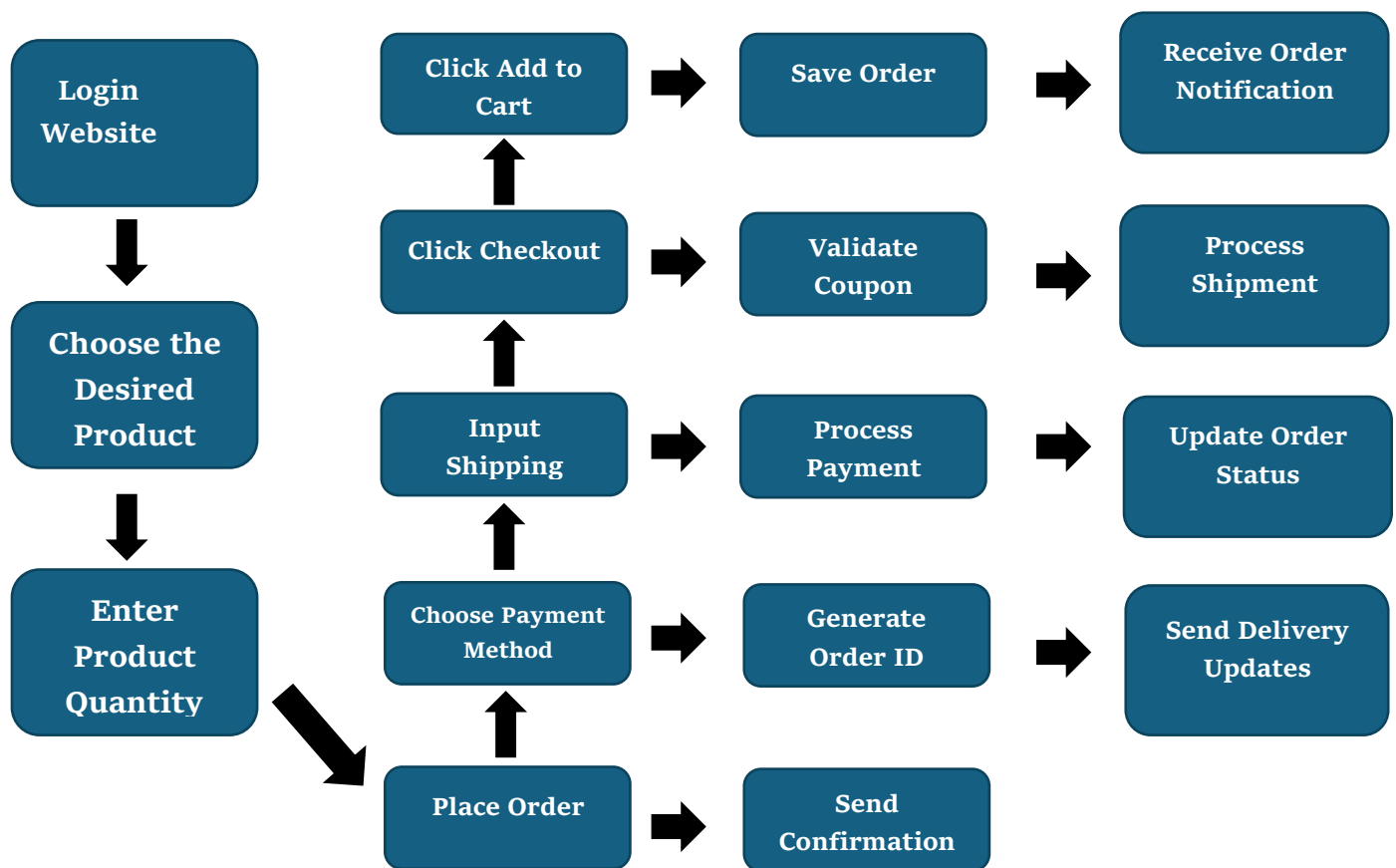


# Marketplace-Technical- Foundation General-E- Commerce Furniture Website

## Key Workflow:



- Key Workflow: Detailed Steps

- Below is a detailed breakdown of workflows for **User Registration**, **Product Browsing**, and **Order Placement**, outlining the roles of each component involved in the process.

- 

- **1. User Registration**

- **Steps:**

- **Frontend (Next.js)**

- User visits the registration page and fills in the form with their name, email, and password.
- The data is validated on the client side (e.g., email format, password strength).
- Upon submission, the data is sent to the backend via an API request.

- **Backend (API)**

- The backend receives the registration request.
- The email is checked for uniqueness in the user database.
- If valid, the user's information is securely stored in the database (e.g., hashed passwords).

- **Sanity CMS**

- If needed, Sanity stores user metadata such as profile preferences, addresses, or wish lists.

- **Response**

- Backend returns a success message and authentication token to the frontend.
- Frontend stores the token (e.g., in cookies or local storage) for authenticated user sessions.

- **Roles of Components:**

- **Next.js:** Provides the UI and client-side validation.
- **Backend API:** Handles user data processing, validation, and storage.
- **Sanity CMS:** Optional storage for user-related metadata.
-

- **2. Product Browsing**

- **Steps:**

- **Frontend (Next.js)**

- User lands on the home or shop page.
- The frontend requests product data from Sanity CMS using **GROQ** queries.
- Category filters, sorting options, and search functionality are presented to the user.

- **Sanity CMS**

- Sanity serves as the source of truth for product details, categories, pricing, and promotional content.
- Queries fetch structured product data and metadata (e.g., available inventory).

- **Caching and Pre-rendering**

- Static pages (e.g., homepage, category pages) are pre-rendered at build time via SSG for faster load times.
- Dynamic pages (e.g., individual product pages) use SSR to fetch the latest content.

- **User Interaction**

- The user browses products, applies filters, and clicks on a product to view its details.
- The frontend dynamically renders the product details using the fetched data.

- **Roles of Components:**

- **Next.js:** Fetches and displays product data, manages state for filters and navigation.
- **Sanity CMS:** Provides real-time product data and category information.
- **API Layer (if used):** Acts as a middleman for complex data manipulation or aggregation.
-

- **3. Order Placement**

- **Steps:**

- **Frontend (Next.js)**

- The user adds products to the cart and navigates to the checkout page.
- The cart details are displayed, including product quantities, prices, and totals.
- User inputs shipping and billing details.

- **Payment Processing (Stripe API)**

- On form submission, payment information is sent securely to Stripe.
- Stripe processes the payment and returns a success or failure status.

- **Backend (API)**

- If payment is successful, the backend creates an order record in the database, including:
  - User ID (or guest information).
  - Products purchased.
  - Total price.
  - Shipping details.
- The backend updates inventory counts in Sanity CMS (if managed there).

- **Sanity CMS**

- Stores order metadata, such as product references and customer details.
- Updates inventory or order status if configured for real-time updates.

- **Order Confirmation**

- The backend sends an order confirmation response to the frontend.
- The frontend redirects the user to a "Thank You" page, displaying order details and a tracking link.

- **Shipment and Tracking**

- Backend triggers a request to the **ShipEngine API** for shipping label creation and tracking number generation.
- Shipment details are linked to the order in the database.

- **Roles of Components:**

- **Next.js:** Manages the user interface, cart state, and checkout flow.
- **Stripe API:** Handles secure payment processing.
- **Backend API:** Manages order creation, inventory updates, and shipping integration.
- **Sanity CMS:** Stores metadata for orders and products.
- **ShipEngine API:** Provides shipment tracking and logistics management.
- 

- **Summary of Roles Across Component:**

Component	Role
Next.js	Frontend UI, client-side validation, data fetching, and dynamic rendering.
Sanity CMS	Content management for products, categories, orders, and inventory.
Backend API	Business logic, data processing, and database management.
Stripe API	Secure payment gateway integration.
ShipEngine API	Shipment tracking and delivery updates.

