## Technical Documentation

---

## 1. System Architecture Overview

The Furniture Ecommerce Website architecture is built to provide an efficient and user-friendly platform for customers to browse, order, and receive furniture products. The system consists of the following main components:

### *Frontend (React/Next.js):*

- **Role**: The frontend represents the user interface where customers interact with the website.
- **Responsibilities**:
    - Browse and search for furniture products.
    - View product details, prices, and availability.
    - Add items to the shopping cart and proceed to checkout.
    - Interact with third-party APIs for payment and shipment tracking.

### *Sanity CMS (Backend):*

- **Role**: Sanity CMS is the content management system and database for the marketplace.
- **Responsibilities**:
    - Manages product catalog, customer information, order details, and stock management.
    - Provides APIs to fetch product details and update orders in real-time.

### *Third-Party APIs:*

- **Payment Gateway API**:
    - **Role**: Processes customer payments securely.
    - **Responsibilities**:
        - Handles transactions, payment verification, and confirmation.
- **Shipment Tracking API**:
    - **Role**: Tracks orders and provides real-time shipment status.
    - **Responsibilities**:
        - Updates customers with delivery statuses and estimated arrival times.

---

## 2. Key Workflows

### *User Registration and Login:*

- **Step 1**: User navigates to the registration or login page.
- **Step 2**: User submits credentials (email, password).
- **Step 3**: Frontend makes an API request to the backend to authenticate the user.
- **Step 4**: If valid, the user is granted access, and session data is stored.

### *Product Browsing and Shopping Cart:*

- **Step 1**: User browses products through categories (e.g., sofas, chairs).
- **Step 2**: Frontend makes a request to the backend (Sanity CMS) for product listings.
- **Step 3**: User adds products to the shopping cart.
- **Step 4**: User proceeds to checkout, where payment and shipment details are gathered.

### *Order Placement and Payment:*

- **Step 1**: User enters payment and shipping information.

- **Step 2**: Frontend communicates with the Payment Gateway API to process payment.
- **Step 3**: Order details are sent to the backend (Sanity CMS) for order creation.
- **Step 4**: Frontend displays order confirmation with estimated delivery date.

---

## 3. Category-Specific Instructions

### *General Ecommerce:*

- **Workflows**:
  - Browsing products, managing cart, and placing orders.
  - Example Endpoint: `/products` (GET) to fetch product listings.

### *Real-Time Shipment Tracking:*

- **Workflows**:
  - Tracking order status in real-time.
  - Example Endpoint: `/shipment/{orderId}` (GET) to track shipment status.

### *Payment Processing:*

- **Workflows**:
  - Handling secure payment for customer orders.
  - Example Endpoint: `/payment` (POST) to process payment.

---

## 4. API Endpoints

| Endpoint | Method | Purpose | Response Example |
|---|---|---|---|
| `/products` | GET | Fetches all product details | `{ "id": 1, "name": "Sofa", "price": 499.99 }` |
| `/orders` | POST | Create new order | `{ "orderId": 123, "status": "Order Created" }` |
| `/payment` | POST | Process payment for an order | `{ "paymentStatus": "Success", "transactionId": "XYZ123" }` |
| `/shipment/{id}` | GET | Track shipment for an order | `{ "shipmentId": "ABC123", "status": "In Transit", "ETA": "2 days" }` |

---

## 5. Sanity CMS Schema Example

For a Product Document:

```
export default {
  name: 'product',
  type: 'document',
  fields: [
    { name: 'name', type: 'string', title: 'Product Name' },
    { name: 'price', type: 'number', title: 'Price' },
    { name: 'stock', type: 'number', title: 'Stock Level' },
    { name: 'description', type: 'text', title: 'Product Description' },
    { name: 'image', type: 'image', title: 'Product Image' }
  ]
};
```

## Conclusion:

This technical documentation outlines the key components, workflows, and API requirements for the Furniture Ecommerce Website. By adhering to this structure, the project can proceed smoothly with clear expectations for functionality and integration across the frontend, backend, and third-party services.