**FURNIRO**

***Marketplace Technical Foundation***

The architecture of the **FURNIRO** marketplace is simple yet effective. It connects several components to ensure a smooth user experience.

Here's how the data flows:

**1. Frontend:** The user interacts with the frontend to browse products, add items to the cart, place orders, and track shipments.

**2. Sanity CMS:** This serves as the backend for managing product details, customer information, and order records.

**3. APIs:** These act as the bridge between the frontend and the CMS, ensuring that the required data is fetched or updated securely and quickly.

The process is as follows:

* When users browse products, the frontend sends a request to the `/products` API.
* When an order is placed, the order details are sent to the `/orders` API, which updates Sanity CMS.
* To track shipments, the frontend communicates with the `/shipment/:orderId` API.

**Key Workflows**

***Browsing Products:***

1. The user visits the marketplace.

2. The frontend calls the `/products` API to fetch the list of products from the CMS.

3. The products, including images, prices, and descriptions, are displayed to the user.

***Adding to Cart:***

1. The user selects a product and clicks "Add to Cart."

2. The product ID is saved in the local state of the application.

***Placing an Order:***

1. The user proceeds to checkout.

2. The frontend sends the order details (customer info, products, and payment details) to the `/orders` API.

3. The CMS stores the order and marks it with a unique ID.

***Tracking Shipment:***

1. The user enters their order ID to track the shipment.

2. The frontend sends a request to `/shipment/:orderId` API.

3. The API returns the current shipping status and estimated delivery time.

***API Endpoints:***

The FURNIRO marketplace relies on several key API endpoints to handle different user actions and ensure smooth operations:

1. **/products (GET)**:  
   This endpoint is used to fetch all product details from the database. It provides information such as the product ID, name, price, stock availability, and any other relevant details required to display products to users.
2. **/orders (POST)**:  
   This endpoint is designed to save new orders into the system. When a customer places an order, their details, along with the selected products and total amount, are sent to this endpoint. The system processes the request and stores the order in the database, returning a success status and a unique order ID.
3. **/shipment/:orderId (GET)**:  
   This endpoint allows users to track their shipments. By providing the order ID, users can retrieve the current shipping status, including whether the item is in transit, delivered, or delayed. Additional information, like the estimated time of arrival (ETA), is also included.

***Sanity CMS Schema:***

The Sanity CMS is configured with schemas that organize and store marketplace data effectively.

1. **Products Schema**:  
   This schema manages all product details, including names, prices, stock levels, images, descriptions, and categories. It ensures the frontend can easily fetch and display products to users.
2. **Orders Schema**:  
   This schema handles customer orders. It stores data such as customer names, emails, purchased products, total order amounts, payment statuses, and shipping statuses. This organization streamlines order management and tracking.
3. **Customers Schema**:  
   The customer schema focuses on user data. It includes fields for customer names, email addresses, delivery addresses, and order history, allowing for a personalized user experience and efficient customer service.