



Marketplace Technical Foundation Documentation Comforty.

Furniro Website

Hassan Farid
hassanfarid667@gmail.com

Contents

| | |
|---------------------------------------|---|
| API Endpoints Documentation..... | 5 |
| 1. Features | |
| 2. Third-Party Integrations | |
| 3. Problems Solved | |
| 4. Benefits | |
| 5. Pages | |
| 6. Technical Architecture | |
| 7. API Endpoints Documentation | |

Marketplace Technical Foundation of Furniro

Overview

Furniro is a modern furniture marketplace that simplifies shopping with an intuitive platform for discovering, customizing, and purchasing furniture. The app delivers a seamless user experience with live search, advanced filtering, and a responsive design.

Features

- **Dynamic Search & Filters:** Quickly find furniture by category, price, and material.
- **Responsive Design:** Optimized for desktops, tablets, and mobile devices.
- **Custom Options:** Personalize furniture with size, color, and material choices.
- **Cart Management:** Real-time cart updates and price calculations.

Third-Party Integrations

- **Secure payments** via Stripe/PayPal.
 - **Real-time shipping estimates** with shipment APIs.
 - **Admin Dashboard:** Manage products, orders, and users effectively.
-

Problems Solved

- Centralized platform for efficient furniture discovery.
 - Enhanced customization for personalized purchases.
 - Simplified checkout with integrated payments and shipping.
-

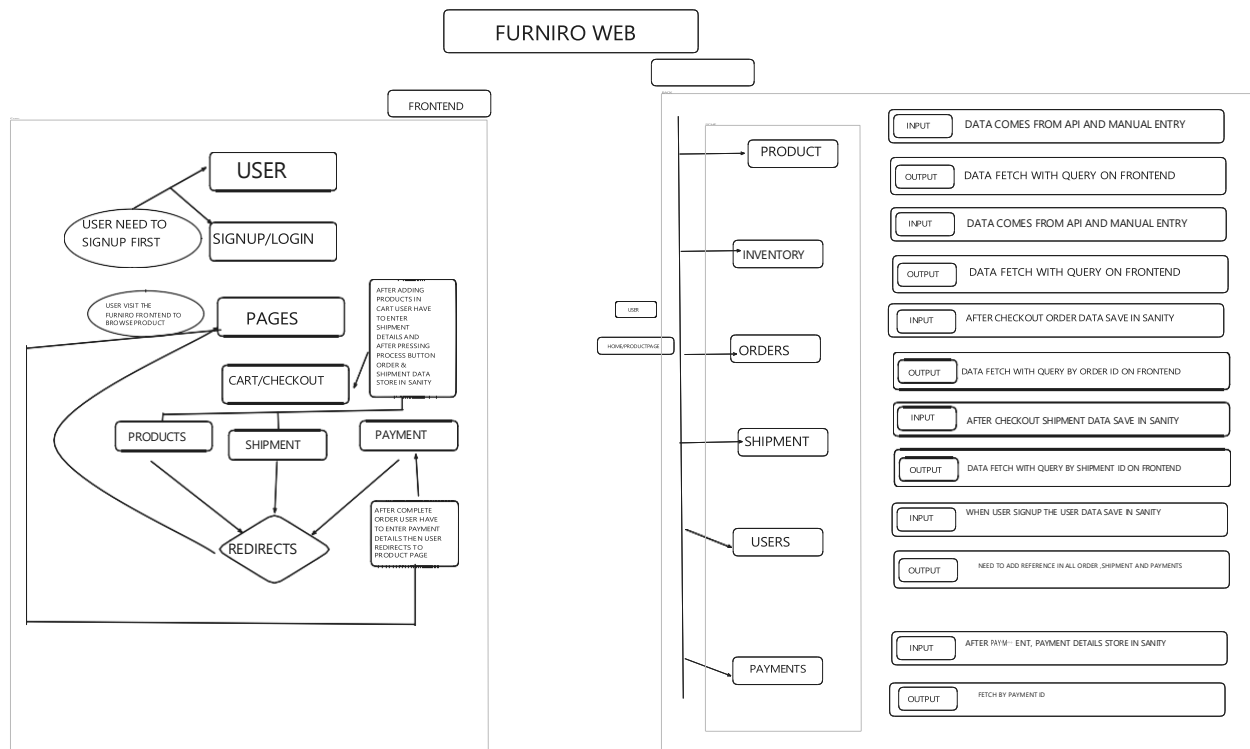
Benefits

- Convenience: Browse, customize, and shop from anywhere.
- Transparency: Detailed product information and live shipment tracking.
- Scalability: Designed to grow with new features and categories.

Furniro aims to redefine furniture shopping by offering a streamlined, user-friendly experience.

Pages

- Home Page
- Contact Page
- Login/Signup Page
- Blog Page
- Shop Page
- Dynamic Product Page
- Cart Page
- Account Page
- Wishlist Page
- Checkout Page



Technical Architecture

Frontend (Next.js)

The frontend of the Furniro marketplace focuses on delivering a seamless user experience through a responsive and user-friendly interface.

- **Essential Features:**
 - Home Page: Showcase featured products and categories.
 - Product Listing: Display all available products with search and filter options.
 - Product Details: Detailed view with customization options (color, size, etc.).
 - Cart Page: Real-time cart management with price calculations.
 - Checkout Page: Collect user details and integrate payment options.
 - Order Confirmation: Display order summary and tracking details.
- **Additional Features:**
 - Live Search and Filters: Allow users to search and filter products dynamically.
 - Fully Responsive Design: Ensures compatibility across mobile, tablet, and desktop devices.
 - Interactive UI/UX: Modern, engaging design with intuitive navigation.

Backend

The backend of the Furniro marketplace is powered by Sanity CMS and additional tools to handle data efficiently and provide secure services.

- **Sanity CMS:**
 - Manage product data, including titles, descriptions, prices, images, and categories.
 - Store customer details such as names, addresses, and contact information.
 - Record and manage order details, including products, quantities, and statuses.
- **Backend Functionalities:**
 - User Authentication: Secure login and registration for customers.
 - Payment Handling: Process transactions securely through integrated APIs.
 - Order Tracking: Manage order statuses and provide tracking details.

Third-Party API Integrations

To enhance the functionality and provide a smooth shopping experience, the Furniro marketplace integrates the following APIs:

1. **Payment Gateway APIs:**
 - **Stripe:** For secure and reliable payment processing.
 - Features: Accept multiple payment methods, handle refunds, and provide transaction receipts.
2. **Shipment APIs:**
 - Real-time shipment tracking and delivery status updates.
 - Features: Calculate shipping costs based on user location and delivery speed.
3. **Product APIs:**
 - Use external APIs for populating product data dynamically in Sanity
 - Features: Fetch and display product details and categories efficiently from Sanity.

Tools and Technologies

- **Frontend:**
 - Framework: Next.js
 - Language: TypeScript
 - Styling: Tailwind CSS
 - State Management: Context API
 - Responsive Design: Flexbox, Grid Layout

- **Backend:**
 - CMS: Sanity CMS
 - APIs: RESTful APIs for dynamic content management
 - Security: Clerk for user authentication

Testing & Deployment

Test the website for responsiveness and functionality.

Deploy using platforms like Vercel.

Final Deliverable

A fully functional, user-friendly furniture website with dynamic features and integrated APIs.

API Endpoints Documentation

| Endpoint Name | Method | Description | Payload | Response Example |
|---------------|--------|--|--|---|
| /products | GET | Fetch all available products. | N/A | [{ "id": 1, "title": "Wooden Chair", "price": 500, "stock": 20, "image": "url", "category": "Furniture " }] |
| /product/:id | GET | Fetch details of a specific product by ID. | N/A | { "id": 1, " title ": "Wooden Chair", "price": 500, "stock": 20, "image": "url", "category": "Furniture " } |
| /orders | POST | Create a new order. | { "customerInfo": { "name": "John", "email": "john@example.com", "phone": "1234567890" }, "orderDetails": [{ "productId": 1, "quantity": 2 }], "paymentStatus": "Paid" } | { "orderId": 101, "status": "Success", "message": "Order created successfully" } |

| | | | | |
|-------------|------|---|--|--|
| /order/:id | GET | Fetch status and details of an order by ID. | N/A | { "orderId": 101, "status": "In Transit", "expectedDeliveryDate": "2025-01-20", "items": [...] } |
| /categories | GET | Fetch all product categories. | N/A | [{ "id": 1, "name": "Furniture" }, { "id": 2, "name": "Chairs" }] |
| /shipment | GET | Track the status of an order using a third-party API. | { "shipmentId": 201, "orderId": 101 } | { "status": "In Transit", "ETA": "15 mins" } |
| /checkout | POST | Process payment and finalize the order. | { "cartId": 123, "paymentMethod": "Credit Card", "paymentDetails": { "cardNumber": "xxxx", "expiryDate": "xx/xx" } } | { "status": "Success", "orderId": 101, "message": "Payment processed and order placed successfully." } |