Marketplace Technical Foundation - Furniro

1. System Architecture Overview

Front-End:

- Responsive design to ensure compatibility with mobile, tablet and desktop screens.
- Developed using modern web technologies.
- User friendly interface across all pages.

Essential Pages:

1) **Home Page**: Central hub showcasing featured furniture, new arrivals etc.
2) **Shop**: Displays all available furniture categories.
3) **Single Product Page**: Detailed view of individual furniture pieces, including descriptions, sizes and customization options.
4) **Cart**: User-friendly cart page to review items, adjust quantities.
5) **Checkout**: A secure checkout process for reviewing the order and completing payment.
6) **Login/Signup**: User authentication pages for customers to register and manage their accounts.
7) **Contact Page**: Form for customers to get in touch with inquiries, support or feedback.
8) **Blog**: A section for blog posts related to furniture trends.
9) **Profile Page**: A personalized dashboard for customers.
10) **Product Comparision**: Enables customers to compare different furniture pieces
11) **Order Traking**: Allows customers to track the status of their orders and view shipment updates.

Sanity CMS:

Using Sanity CMS to manage product data, customer details, and order records. Sanity acts as the database for our marketplace.

Sanity CMS will be used for managing:
□

- **Products**: Categories, prices, stocks, and customizations. □
- **Blogs**: Title, content, images, and tags for articles.
- **Customer Information**: Storing User details, saved addresses, and order history.
- **Orders**: Providing Real-time updates on order status.

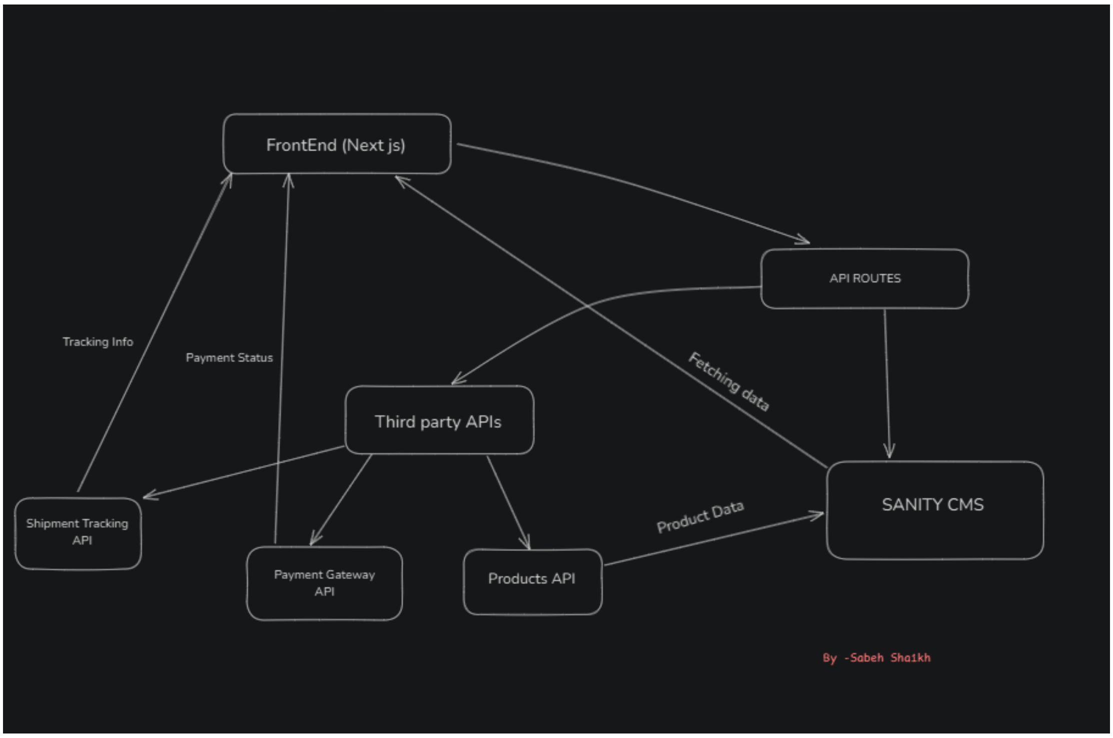
Third-Party API Integrations:

Integrate APIs for shipment tracking, payment gateways, and other required backend services.

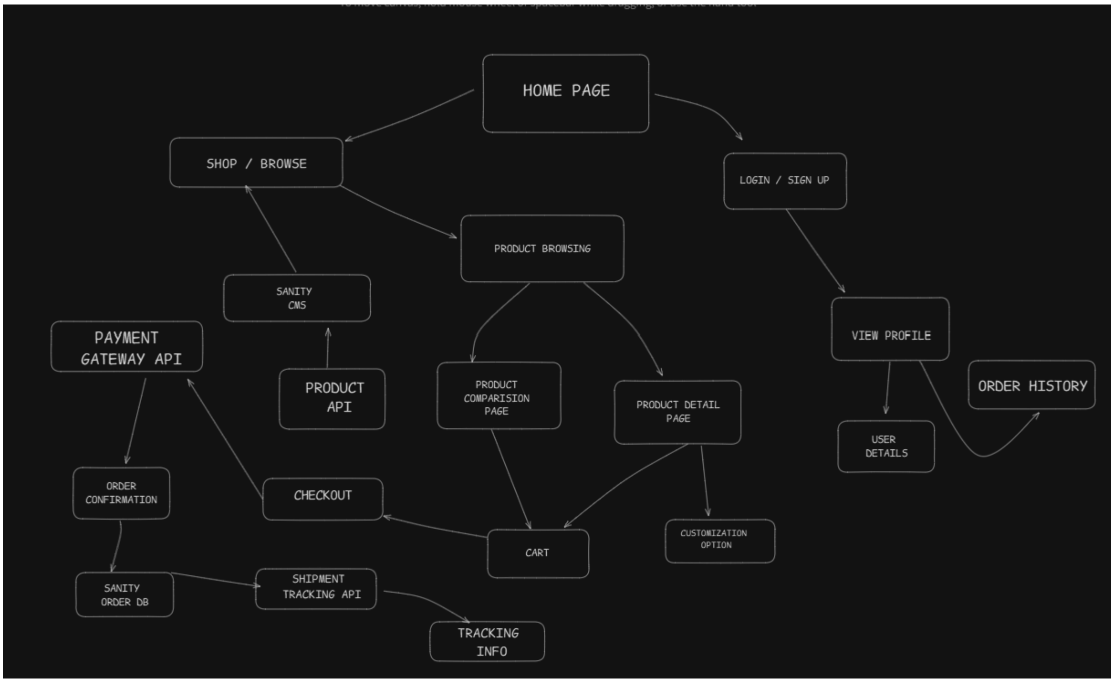
APIs will be used for:

- **Payment Solutions:** Secure gateways for processing transactions.
- **Logistics:** Integration with delivery partners for efficient shipping and tracking.
- **Authentication:** Secure sign-in/sign-up methods.
- **Tracking:** Real-time order and delivery tracking systems.

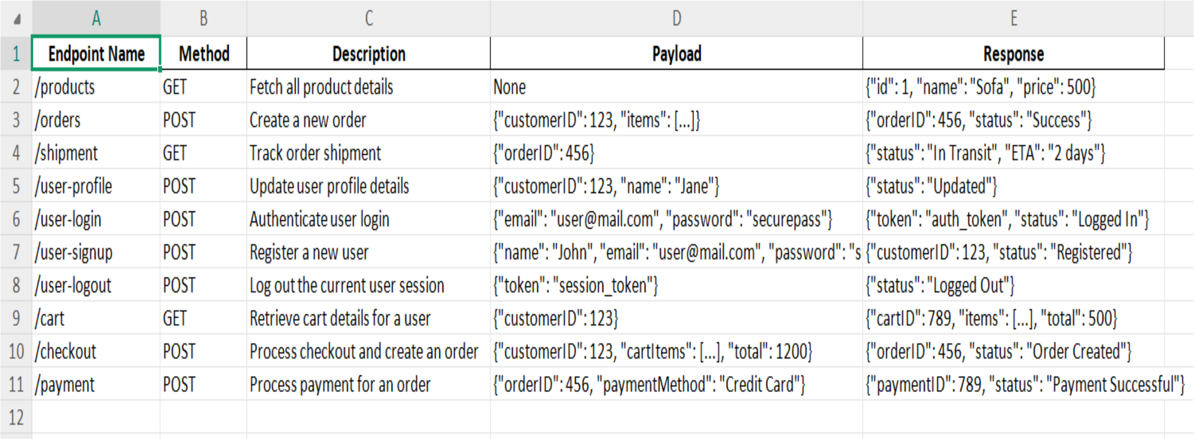
System Architecture Diagram:



Workflow Diagram:



API Endpoints:



	A	B	C	D	E
1	**Endpoint Name**	**Method**	**Description**	**Payload**	**Response**
2	/products	GET	Fetch all product details	None	{"id": 1, "name": "Sofa", "price": 500}
3	/orders	POST	Create a new order	{"customerID": 123, "items": [...]}	{"orderID": 456, "status": "Success"}
4	/shipment	GET	Track order shipment	{"orderID": 456}	{"status": "In Transit", "ETA": "2 days"}
5	/user-profile	POST	Update user profile details	{"customerID": 123, "name": "Jane"}	{"status": "Updated"}
6	/user-login	POST	Authenticate user login	{"email": "user@mail.com", "password": "securepass"}	{"token": "auth_token", "status": "Logged In"}
7	/user-signup	POST	Register a new user	{"name": "John", "email": "user@mail.com", "password": "s	{"customerID": 123, "status": "Registered"}
8	/user-logout	POST	Log out the current user session	{"token": "session_token"}	{"status": "Logged Out"}
9	/cart	GET	Retrieve cart details for a user	{"customerID": 123}	{"cartID": 789, "items": [...], "total": 500}
10	/checkout	POST	Process checkout and create an order	{"customerID": 123, "cartItems": [...], "total": 1200}	{"orderID": 456, "status": "Order Created"}
11	/payment	POST	Process payment for an order	{"orderID": 456, "paymentMethod": "Credit Card"}	{"paymentID": 789, "status": "Payment Successful"}
12					