# FRONTEND REQUIREMENTS

## FLOW CHART BY MUHAMMAD HUZAIFA

ROLL NO : 00149806

DAYS/TIME: SUNDAY -09:00AM - 12:00PM

This document describes the user journey flow for an eCommerce marketplace, focusing on key steps and their technical implementation. It is tailored for building a robust and user-friendly platform.

### Flowchart Overview

Below is the complete user journey flow for an eCommerce marketplace:

### Home Page:

User lands on the homepage.

Displays featured categories, popular products, and a search bar.

### Product Browsing:

User selects a category or uses the search bar

### Product Details:

User clicks on a product to view its details.

Page includes product description, price, availability, and user reviews.

### Add to Cart:

User adds the product to the cart.

Cart updates dynamically with quantity and price.

### Checkout:

User proceeds to the checkout page.

Provides shipping address and selects a delivery option.

### Payment:

User enters pavment details.

Secure payment gateway processes the transaction.

### Order Confirmation:

Order details are displayed and sent via email.

Order status is updated in the backend.

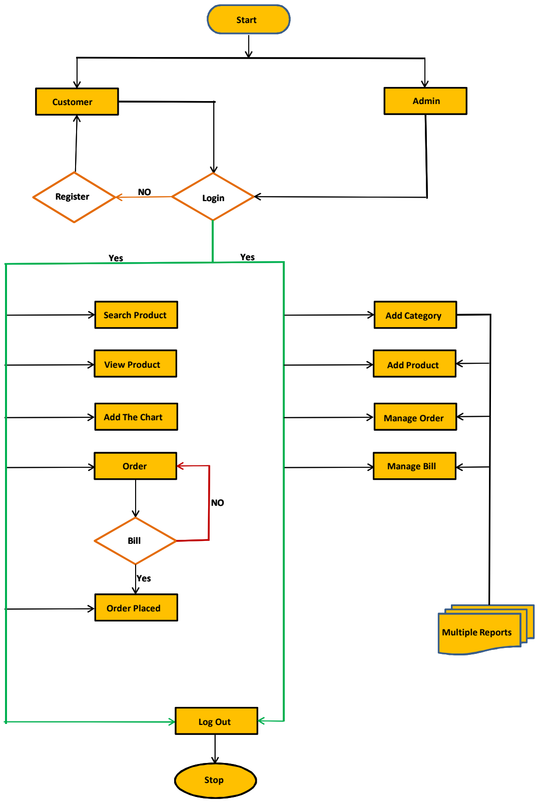
### Shipment Tracking:

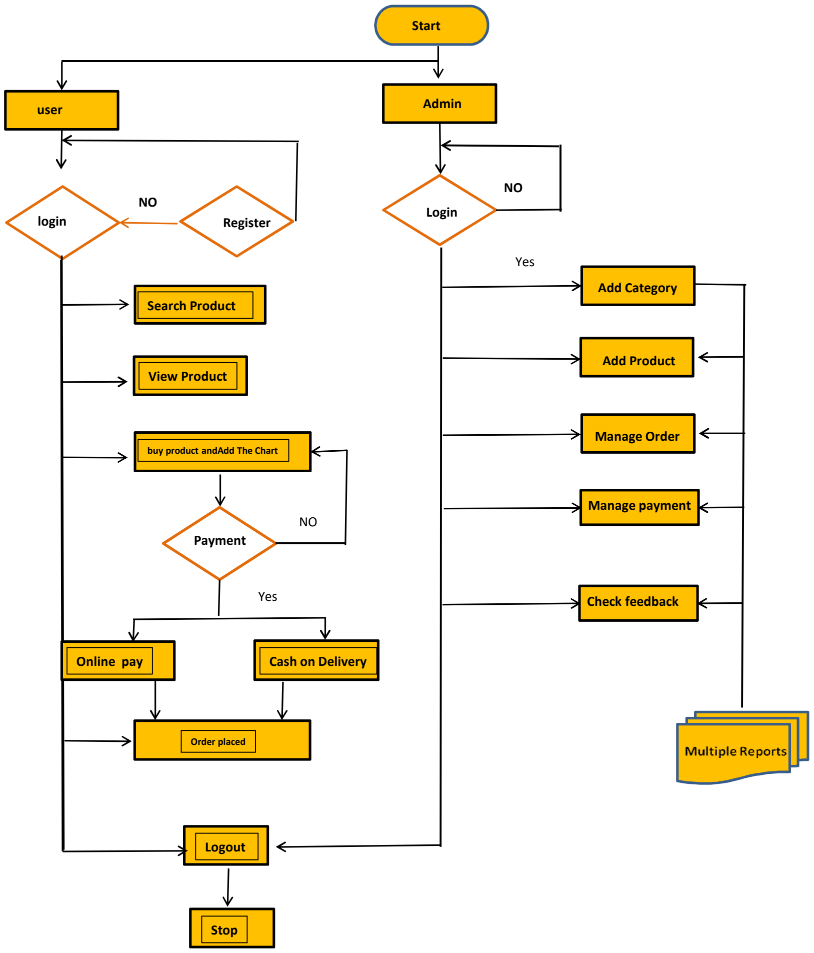
User visits the "Order History" section.

Real-time shipment tracking is enabled via API integration.

### Delivery:

Product is delivered to the user's address





## Frontend Requirements:

Cher receives a notification and can leave a review

User friendly mterface for browsing products

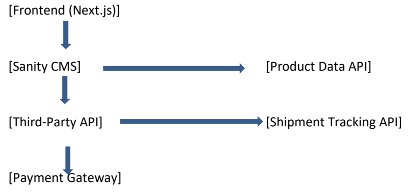
Responsive design for moble and desktop users

Essential pages Home, Product Listing, Product Details, Cart Checkout, and Order Confirmation

## Design System Architecture

Create a high-level diagram showing how your system components itaract. Use too's the pen and paper software like Lucidchart, Figma or Excaldraw For example, a more detalend architecture might include workflows such as

### System Architecture Overview Diagram:



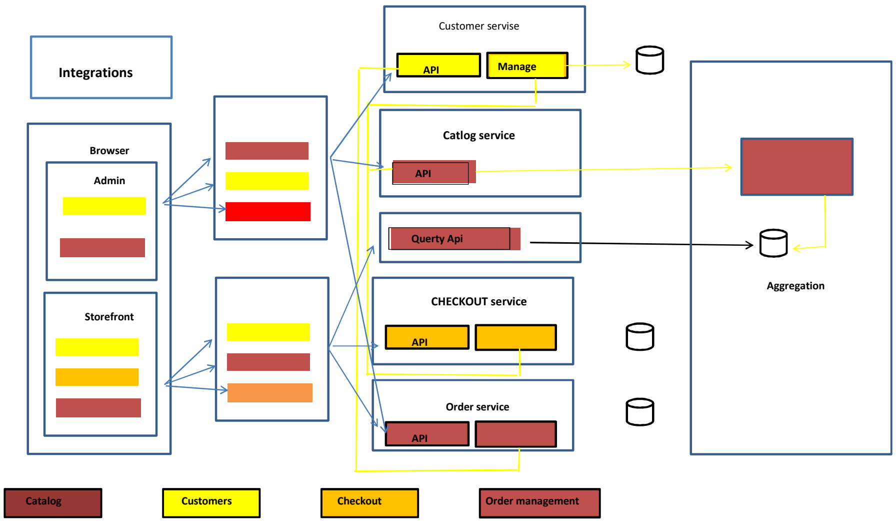
[Frontend (Next.js)]

[Sanity CMS] [Product Data API]

[Third-Party API] [Shipment Tracking API]

[Payment Gateway]

# System Architecture



### Components and Roles:

Frontend (Next.js);

1. Displays the user interface for browsing products, managing the cart, and placing orders.

2. Handles user interactions and communicates with backend services via APIs.

Sanity CMS:

1. Acts as the primary backend to manage product data, customer details, and order records.

2. Provides APIs for the frontend to fetch and update data.

Product Data API:

Provides endpoints to fetch product listings, details, and inventory status.

Third-Party APIs

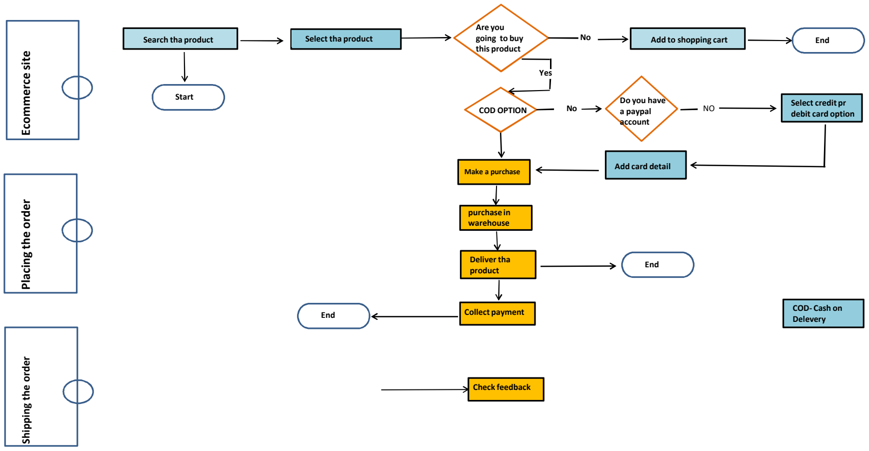
Provides endpoints to fetch product listings, details, and inventory status.

Third-Party APIs:

Integrates services like shipment tracking and payment processing.

Payment Gateway:

Processes user payments securely and provides transaction confirmation


Ecommerce site
Placing the order
Shipping the order
Search tha product
Start
Select tha product
Are you going to buy this product
No
Add to shopping cart
End
Yes
COD OPTION
No
Do you have a paypal account
NO
Select credit pr debit card option
Add card detail
Make a purchase
purchase in warehouse
Deliver tha product
End
Collect payment
End
COD- Cash on Delevery
Check feedback

**Also make this EDR diagram in your project.**

**This diagram will define the relationships between entities in your database.**

## Example Entities:

**1. Food Items:**

**Fields: id, name, price, description, image, categoryld**

**2. Categories:**

**Fields id, nate.**

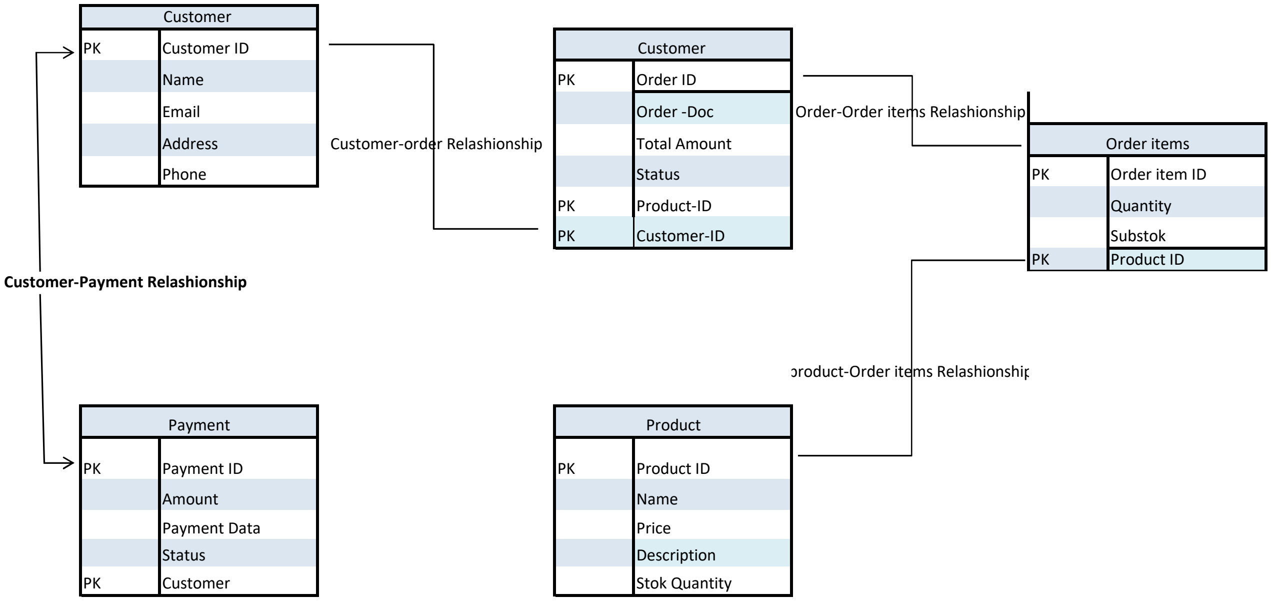
**3. Users:**

**Fields: 10, nase, email, password**

**4. Orders:**

**Fields, id, user lit**

**order Date, status**



| Customer | |
|---|---|
| PK | Customer ID |
| | Name |
| | Email |
| | Address |
| | Phone |

| Customer | |
|---|---|
| PK | Order ID |
| | Order -Doc |
| | Total Amount |
| | Status |
| PK | Product-ID |
| PK | Customer-ID |

| Order items | |
|---|---|
| PK | Order item ID |
| | Quantity |
| | Substok |
| PK | Product ID |

| Payment | |
|---|---|
| PK | Payment ID |
| | Amount |
| | Payment Data |
| | Status |
| PK | Customer |

| Product | |
|---|---|
| PK | Product ID |
| | Name |
| | Price |
| | Description |
| | Stok Quantity |

Customer-order Relashionship
Order-Order items Relashionship
**Customer-Payment Relashionship**
product-Order items Relashionship

| ENDPOINT | Method | Description | Parameters | Response Example |
|---|---|---|---|---|
| | | | | |
| /api/furniture | GET | Fetch all food items | None | (id: I, name: "Syltherine"} |
| | | | | |
| /api/furnitures/id | GET | Fetch a single furniture item | id (Path) | { id: 1, name: "Syltherine"} |
| | | | | |
| /api/furnitures | POST | Add a new furniture item | name, price, category (Body) | (success: true, id: 5 } |
| | | | | |
| /api/furnitures/:id | PUT | Update a furniture item | id (Path), name, price (Body) | {success: true } |
| | | | | |
| /api/furnitures/id | DELETE | Delete a furniture item | id (Path) | {success: true} |
| | | | | |
| /api/categories | GET | Fetch all furniture categories | None | { categories: ["Jane Smith"]} |