"Marketplace Technical Foundation - [Furniture Website]"

1: Frontend Requirements:

User-friendly Interface:

- Navigation: Clear and intuitive navigation for easy browsing.
- Search Functionality: Robust search bar with filter options (by category, price, etc.

Responsive Design:

- Ensure the website is fully responsive and optimized for both mobile and desktop users.

Essential Pages:

Home: Showcasing featured products,search bar, popular products and new arrivals.

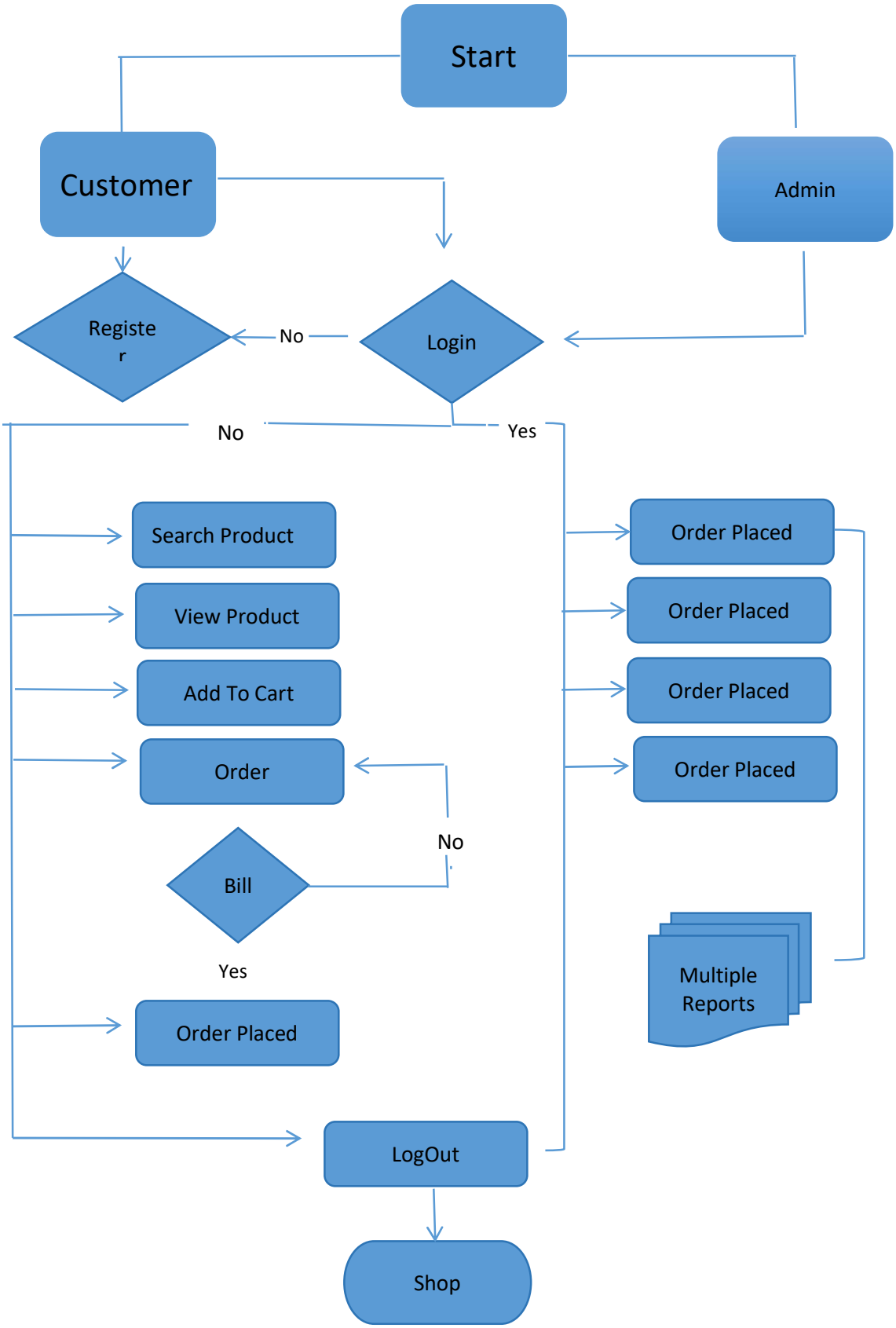
Product Listing:Products are view with sorting and filtering options by price, brand or rating.

Product Details: High-quality images, detailed descriptions, customer reviews, and related products.

Add to Cart: Overview of selected items with the option to update quantities or remove items.

Checkout: Secure checkout process with multiple payment options.

Order Confirmation: Order detailed are displayed and send via email, order status is updated in the back-end.


Start
Customer
Admin
Registe
r
No
Login
No
Yes
Search Product
View Product
Add To Cart
Order
Bill
No
Yes
Order Placed
Order Placed
Order Placed
Order Placed
Order Placed
Multiple
Reports
LogOut
Shop

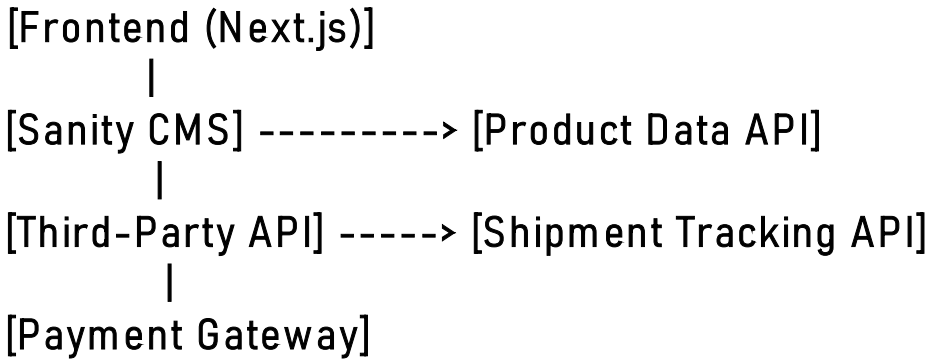
Sanity CMS as Backend:

- **Manage Product Data:** Create a schema for product details (name, description, price, dimensions, materials, images, stock).
- **Customer Details:** Design a schema for customer info (names, contacts, addresses, purchase history).
- **Order Records:** Set up a schema for orders (order ID, customer ID, product IDs, status, payment, shipping).
- **Integration:** Use Sanity's APIs to connect with your frontend, keeping your website dynamic.
- **Security:** Include validation and permissions to protect data; manage user roles and access control.

Third-Party APIs:

- **Shipment Tracking APIs:** Integrate APIs from shipping providers to allow customers to track their orders directly on your site.
- **Payment Gateways:** Use payment gateway APIs (e.g., PayPal, Stripe) for secure and reliable transactions.
- **Other Backend Services:** Integrate necessary APIs for features like tax calculations, inventory management, and customer support.
- **Ensure Frontend Functionality:** Make sure the APIs provide all required data for a smooth user experience on the frontend, such as order statuses and payment confirmations.

2. Design System Architecture



Component Roles:

Frontend (Next.js): Handles user interactions and requests data.

Sanity CMS: Manages content, products, customer details, and orders.

Product Data API: Provides product information to the Frontend.

Shipment Tracking API: Manages shipment tracking.

Payment Gateway: Processes secure payments.

3.Plan API Requirements:

" General eCommerce"

API Endpoints:

1. Endpoint Name :Product:

.Method: GET

.Description: Fetch all available product from sanity.

. Response Example :

```
{
 "id": 1,
 "name": "Product A",
 "price": 100,
 "stock": 50,
 "image": "image_url"
}
```

2. Endpoint Name ::Orders

.Method: Post

.Description:Create a new order in Sanity.

. Payload Example :

```
{
 "customerInfo": {
  "name": "John Doe",
  "email": "john@example.com",
  "address": "123 Main St, Anytown, USA"
 },
 "product Details": [
  {
   "productId": 1,
   "quantity": 2
  }
 ],
```

"paymentStatus": "Paid"
}

3. Endpoint Name :Product:

.Method: Shipment

.Description: Track order status via third-party API.

. Response Example :

{
"shipmentId": "SHIP123",
"orderId": 1,
"status": "In Transit",
"expectedDeliveryDate": "2025-01-20"
}

4. API Endpoints:

EndpointResponse	Method	Purpose	Response Example
/products	GET	Fetches all product details	{ "customerInfo": {..}, "product Details": [..]}
/orders	POST	Create a new order in Sanity	{ "customerInfo": {..}, "productDetails": [..]}
/shipment	GET	Track order status via third-party	{ "shipmentId": "SHIP123", "status": "In Transit"}

5. Sanity Schema Example:

```
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' }
  ]
};
```