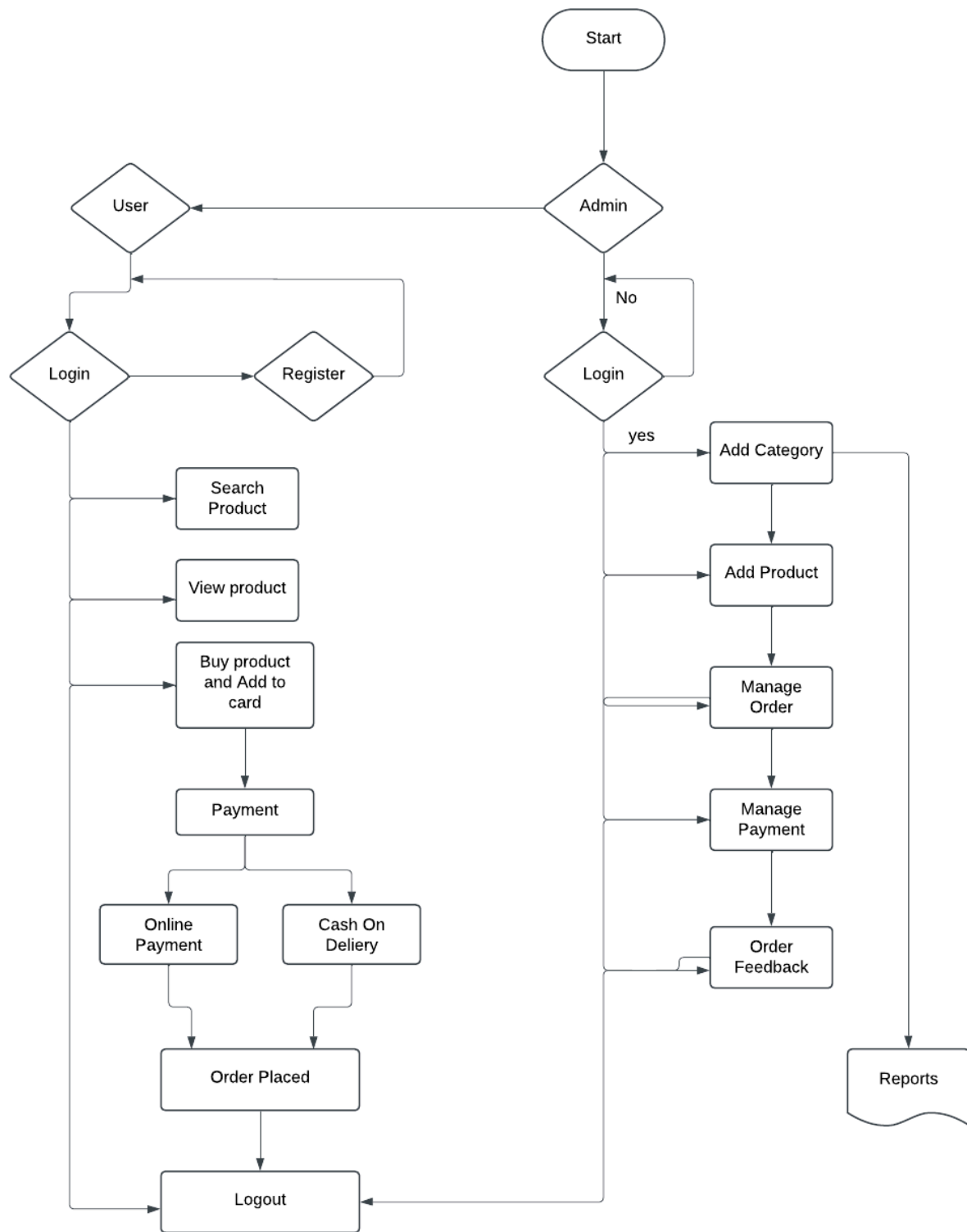


Day_2 (Planing the Technical Foundation) Marketplace Technical Foundation - Furniture Bazaar



1. Define Technical Requirements:

Frontend Requirements:

- User-friendly Interface for Browsing Products:
 - A clean and intuitive layout for easy navigation and product discovery.
 - Category filters, search functionality, and product sorting options.
- Responsive Design for Mobile and Desktop Users:
 - Responsive CSS frameworks or libraries like Tailwind CSS or Bootstrap to ensure consistent user experience across devices.
 - Mobile-first approach to prioritize mobile usability.
- Essential Pages:
 - Home Page: A landing page showcasing featured products, categories, and promotional offers.
 - Product Listing Page: Displays a list of products with options to filter and sort.
 - Product Details Page: Detailed view of a single product, including images, description, price, and reviews.
 - Cart Page: Summary of selected products, allowing users to modify quantities or remove items.
 - Checkout Page: Form for users to enter shipping and payment details.
 - Order Confirmation Page: Displays order summary and confirmation message after successful checkout.

Sanity CMS as Backend:

● Sanity CMS for Managing Data:

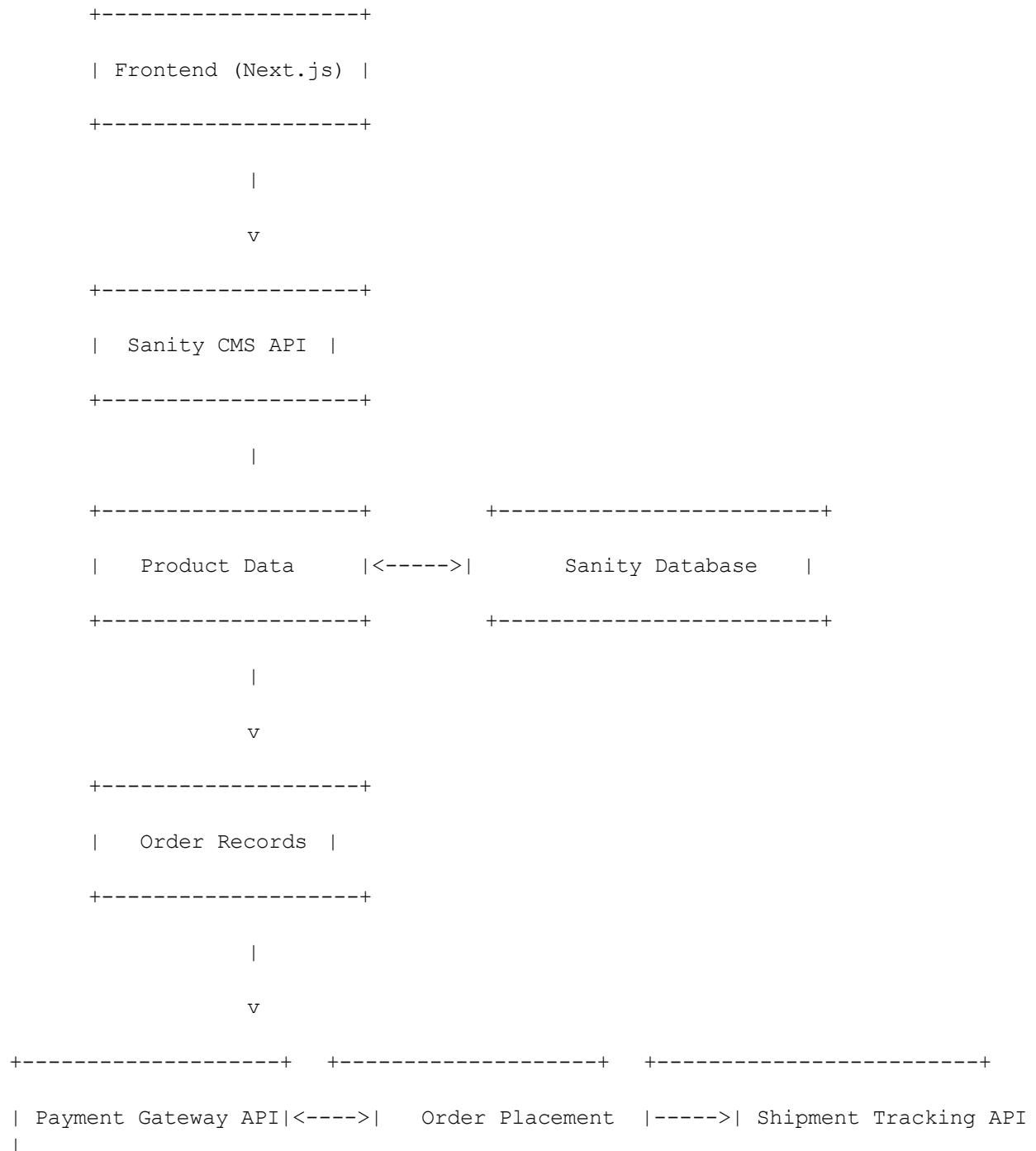
- Use Sanity CMS to handle product data (name, price, stock, description, images), customer details, and order records.
- Design and implement schemas in Sanity for:
 - Products: Including fields for ID, name, price, stock, description, images, categories, and tags.
 - Orders: Capturing order details such as customer information, product details, payment status, and timestamps.
 - Customers: Storing customer data like names, contact info, addresses, and order history.

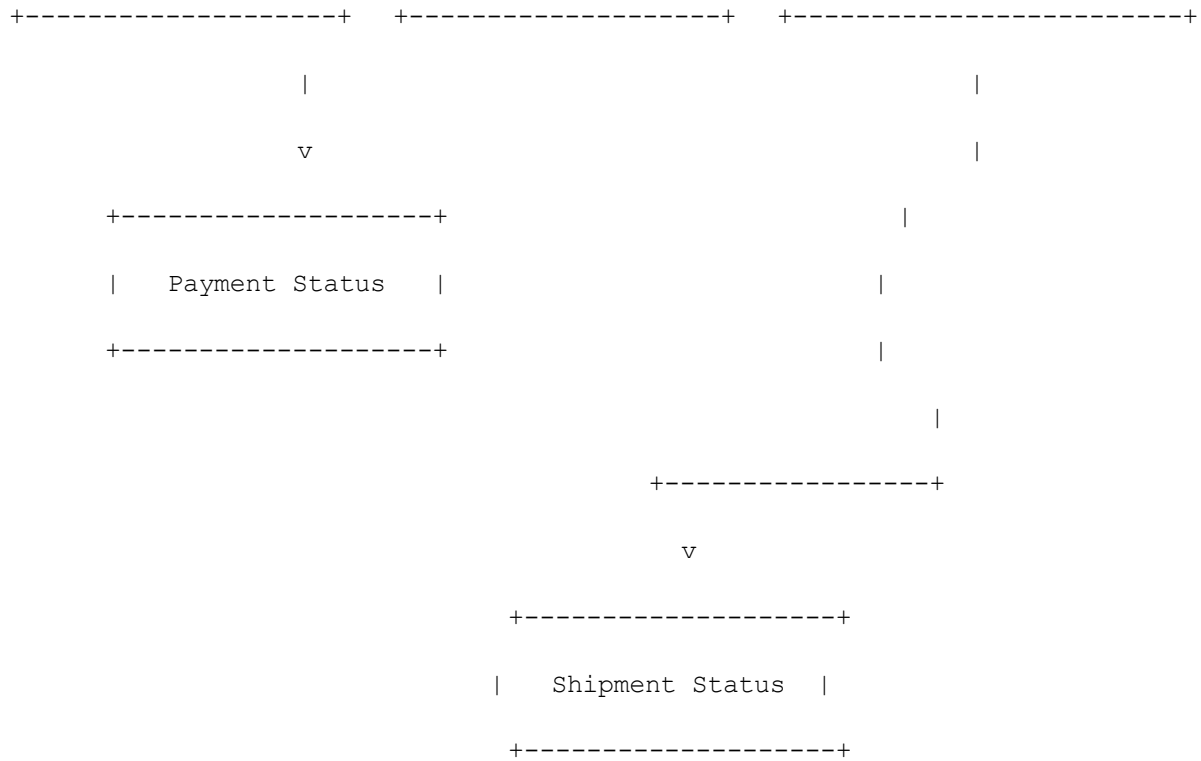
Third-Party APIs:

- Integrate APIs for Essential Services:
 - Shipment Tracking API (e.g., ShipEngine): For real-time shipment updates. API should provide tracking ID, order ID, status, and estimated delivery time.
 - Payment Gateway (e.g., Stripe): Secure payment processing with fields for payment method, transaction status, and refund capabilities.
 - Authentication (e.g., Clerk): For user registration, login, and identity management. API should handle user sessions and secure data exchange.
- Ensure APIs Provide Necessary Data:

- APIs should support both frontend and backend requirements, ensuring seamless data flow for tasks like displaying product availability, tracking shipments, and processing payments.

Design System Architecture





Key Workflows:

1. User Registration:

- User signs up on the frontend.
- Data is stored in Sanity CMS.
- Confirmation is sent to the user.

2. Product Browsing:

- User views product categories on the frontend.
- Sanity CMS API fetches data from the Sanity Database.
- Products are displayed dynamically on the frontend.

3. Order Placement:

- User adds items to the cart and proceeds to checkout.
- Order details are saved in Sanity CMS.
- Payment information is processed through the Payment Gateway API.
- Order confirmation and details are updated in Sanity CMS.

4. Shipment Tracking:

- Shipment status is updated via the Shipment Tracking API.

- Real-time shipment information is displayed to the user.

API Endpoints:

	A	B	C	D	E
1	Endpoint	Method	Description	Parameters	Response Example
2	/api/products	GET	Fetch all available products from Sanity	None	{ id: 1, name: 'Product A', price: 10,000 }
3	/api/products/:id	GET	Fetch details of a specific product	id (Path)	{ id: 1, name: 'Product A', price: 10,000 }
4	/api/orders	POST	Create a new order in Sanity	customer info, product details, payment status (Body)	{ success: true, orderId: 123 }
5	/api/orders/:id	PUT	Update an existing order	id (Path), updated order details (Body)	{ success: true }
6	/api/shipment/:id	GET	Track order status via third-party API	id (Path)	{ shipmentId: 456, orderId: 123, status: 'In Transit', expectedDeliveryDate: '2025-01-20' }
7					
8					
9					
10					
11					