289436 - Sadaf Shahab - (Monday 2:00 to 5:00)

Day 2 -- PLANNING THE TECHNICAL FOUNDATION

What was the problems of marketplace?

1. **No Flexibility:** Customers need temporary furniture but can only buy.
2. **High Costs:** Furniture is expensive for short-term use.
3. **Limited Quality Rentals:** Rental furniture options may not be good quality.
4. **Complicated Delivery & Returns:** Renting furniture can be hard to manage logistically.
5. **Untrustworthy Rentals:** Not enough reliable rental services.
6. **Complex Returns:** Rental return processes can be difficult.
7. **Short-Term Housing Solutions:** People moving temporarily struggle to find affordable furniture.
8. **Limited Rental Variety:** Few platforms offer diverse rental furniture options.
9. **Lack of Eco-Friendly Choices:** Limited availability of sustainable furniture.
10. **Trust Issues:** Customers worry about the quality and durability of rental furniture.
11. **Market Confusion:** Customers get confused with platforms offering only rentals or purchases.

What is the solution for marketplace?

1. **Flexibility:** Rent furniture for temporary use instead of buying.
2. **Lower Costs:** Rent furniture for a fraction of the purchase price.
3. **Access to Quality:** Rent high-quality furniture without long-term commitment.
4. **Easy Delivery & Pickup:** Flexible delivery and return options.
5. **Trusted Rentals:** Centralized, reliable platform for renting or buying.
6. **Simplified Returns:** Clear return policies and easy furniture pickup.
7. **Short-Term Housing Needs:** Furnish temporary spaces without a large upfront cost.
8. **More Rental Variety:** Offer a wide range of rental furniture options.
9. **Sustainability:** Provide eco-friendly rental furniture options.
10. **Trust in Quality:** Detailed product info and reviews to ensure quality.
11. **Simplified Shopping:** Clear options for both renting and buying, reducing market confusion.

Target Audience:

1. Young Professionals
2. Students
3. Travelers/Expats
4. Small Families
5. Eco-Conscious Consumers
6. Budget-Conscious Shoppers
7. Interior Designers/Real Estate Agents
8. Homeowners for Seasonal Updates
9. Online Shoppers

Unique Value Proposition:

1. Flexible Options
2. Affordable Pricing
3. Sustainability
4. Convenience
5. Wide Selection
6. Trust and Quality

Key Features:

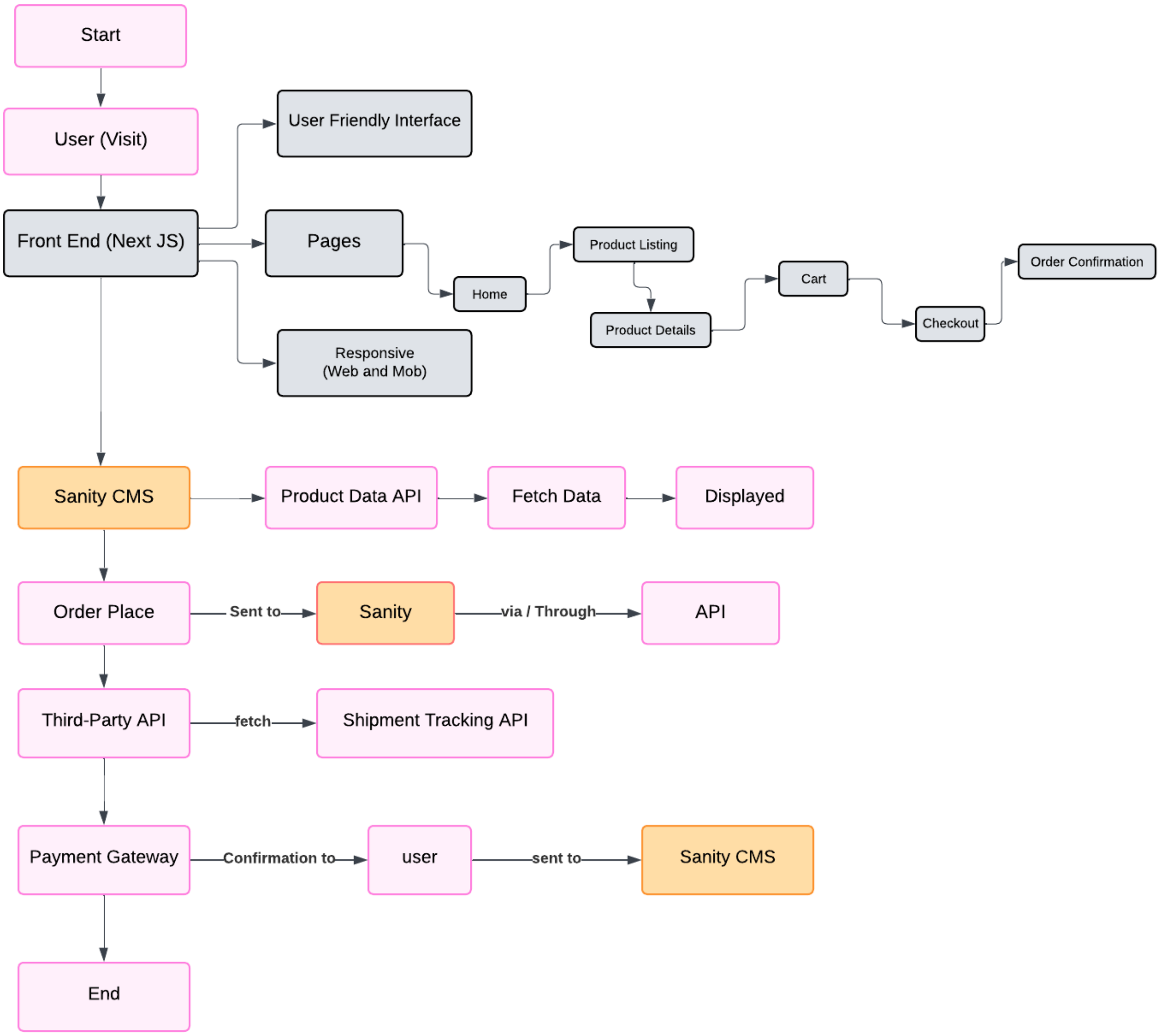
1. **Rent or Buy Options:** Allow customers to choose between renting or purchasing furniture.
2. **3D Product Visualization/AR:** Let customers visualize furniture in their space with AR or 3D models.
3. **Flexible Delivery & Pickup:** Offer flexible delivery and return options for both rental and purchased items.
4. **Product Customization:** Allow customization of size, color, and material for both rented and purchased furniture.
5. **Easy Checkout Process:** Simple and secure checkout with payment options for both rental and purchase.
6. **Subscription & Rental Plans:** Provide different rental duration plans (daily, monthly, etc.).
7. **Eco-Friendly Section:** Highlight sustainable, eco-friendly furniture options.
8. **Customer Reviews & Ratings:** Allow customers to review products and share their experiences.
9. **Clear Return Policies:** Transparent return and exchange policies for both rentals and purchases.
10. **Search & Filter Options:** Advanced search and filters to help customers easily find the right furniture based on their needs (style, price, type, etc.).
11. **Live Chat Support:** Instant support for customers through live chat to answer questions and assist with decisions.
12. **Rental Agreement:** Easy-to-understand rental agreements for customers to review before renting.
13. **Wishlist & Favorites:** Let customers save their favorite items for future reference or purchase.

289436 - Sadaf Shahab - (Monday 2:00 to 5:00)

Transitioning to Technical Planning

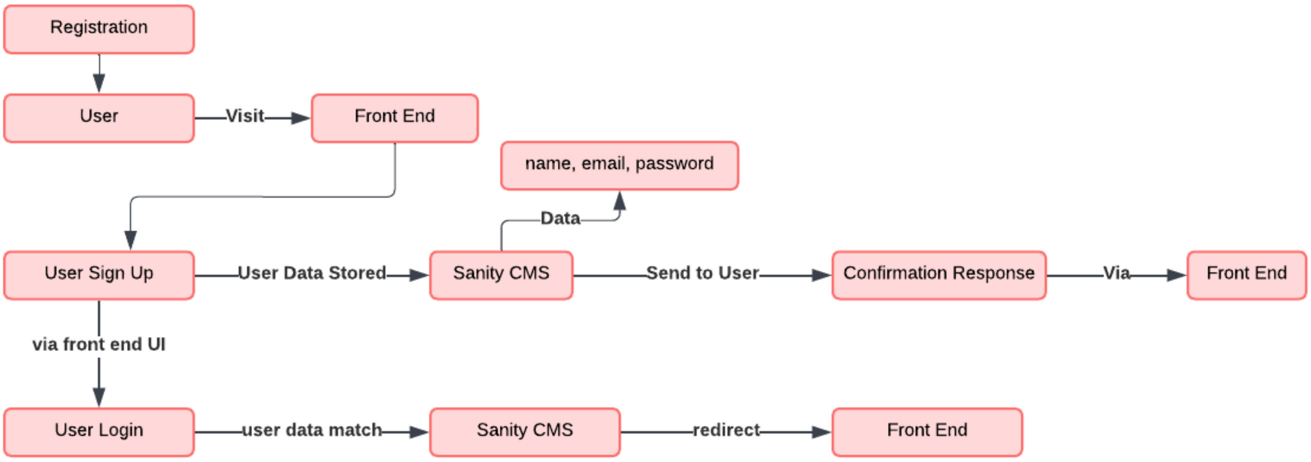
Technical Requirements

- Frontend Requirements
- Sanity CMS as Backend
- Third-Party APIs



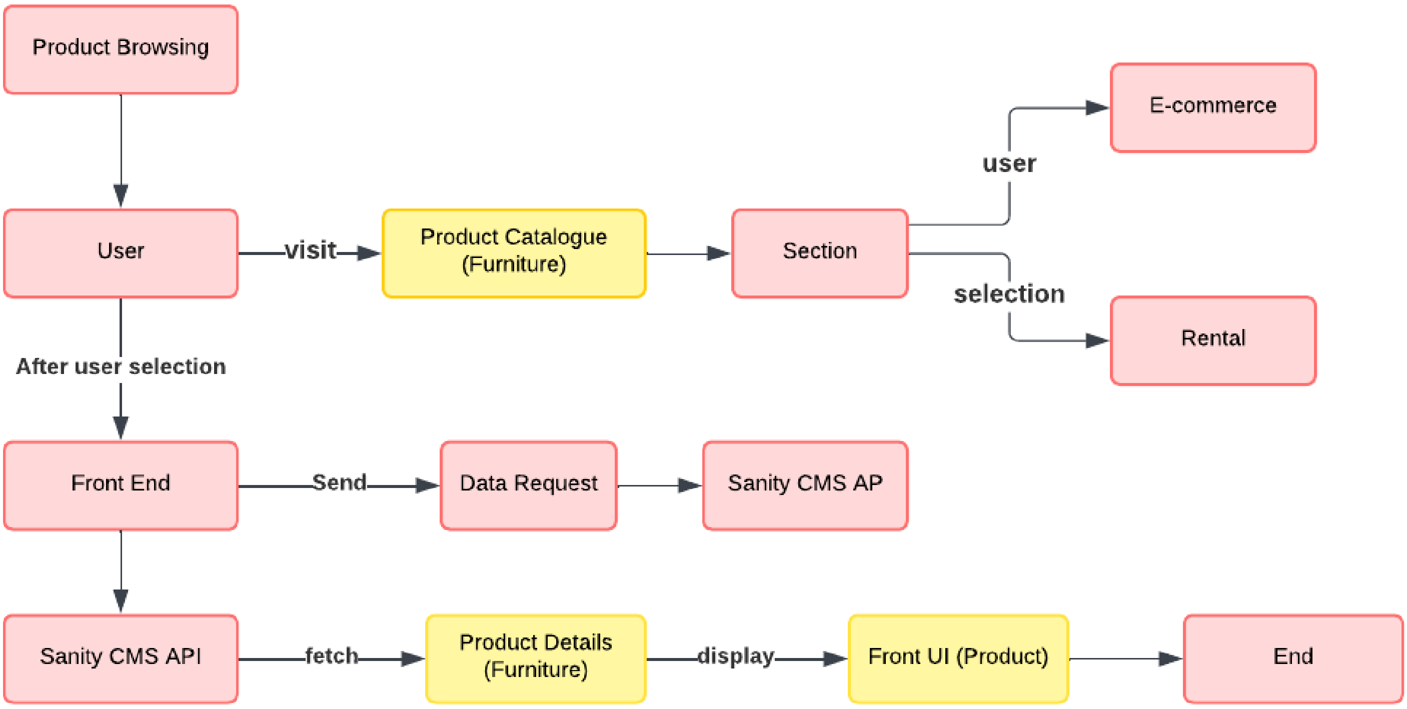
Key Workflows:

User Registration/Login:

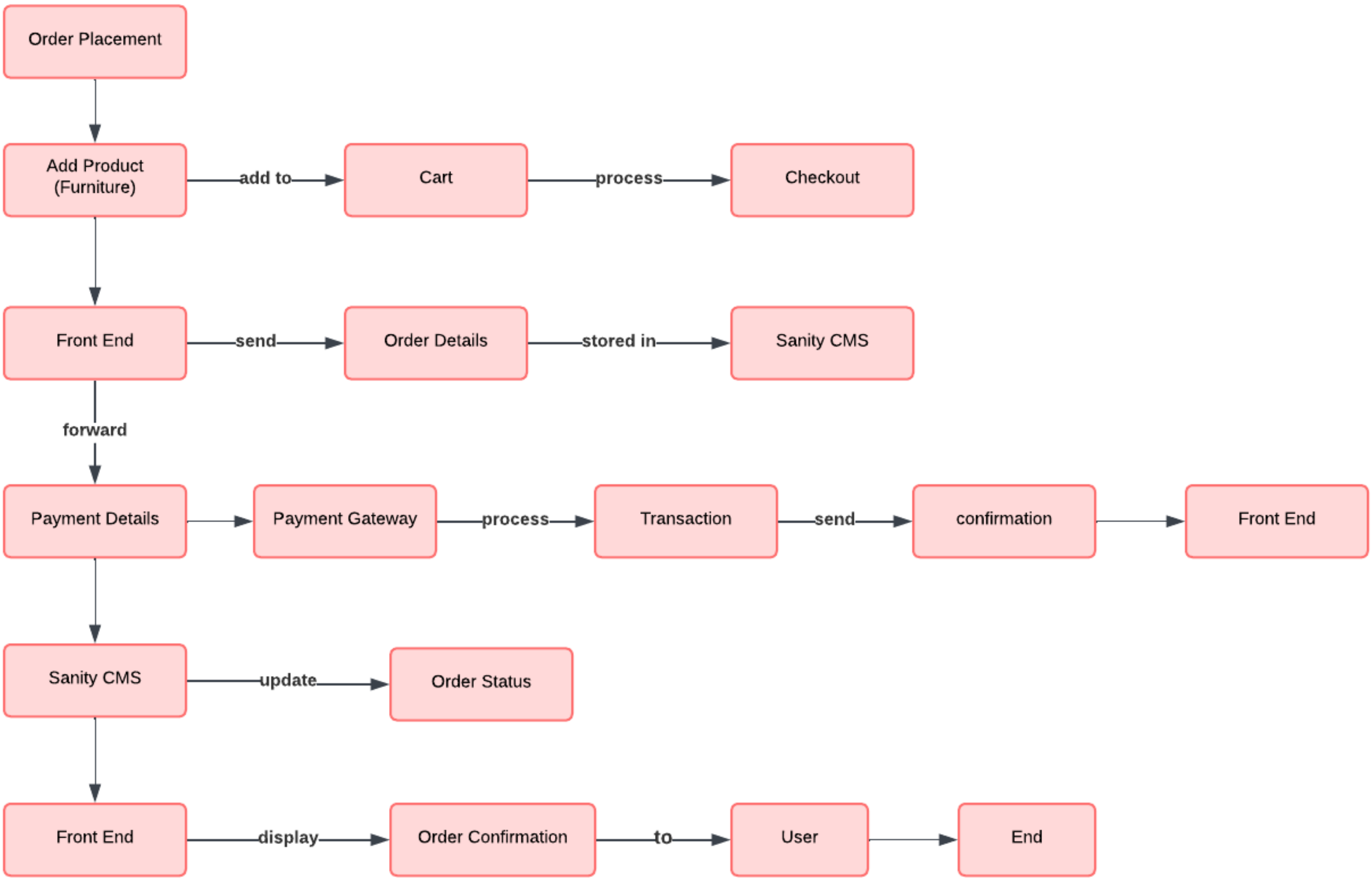


289436 - Sadaf Shahab - (Monday 2:00 to 5:00)

Product Browsing:

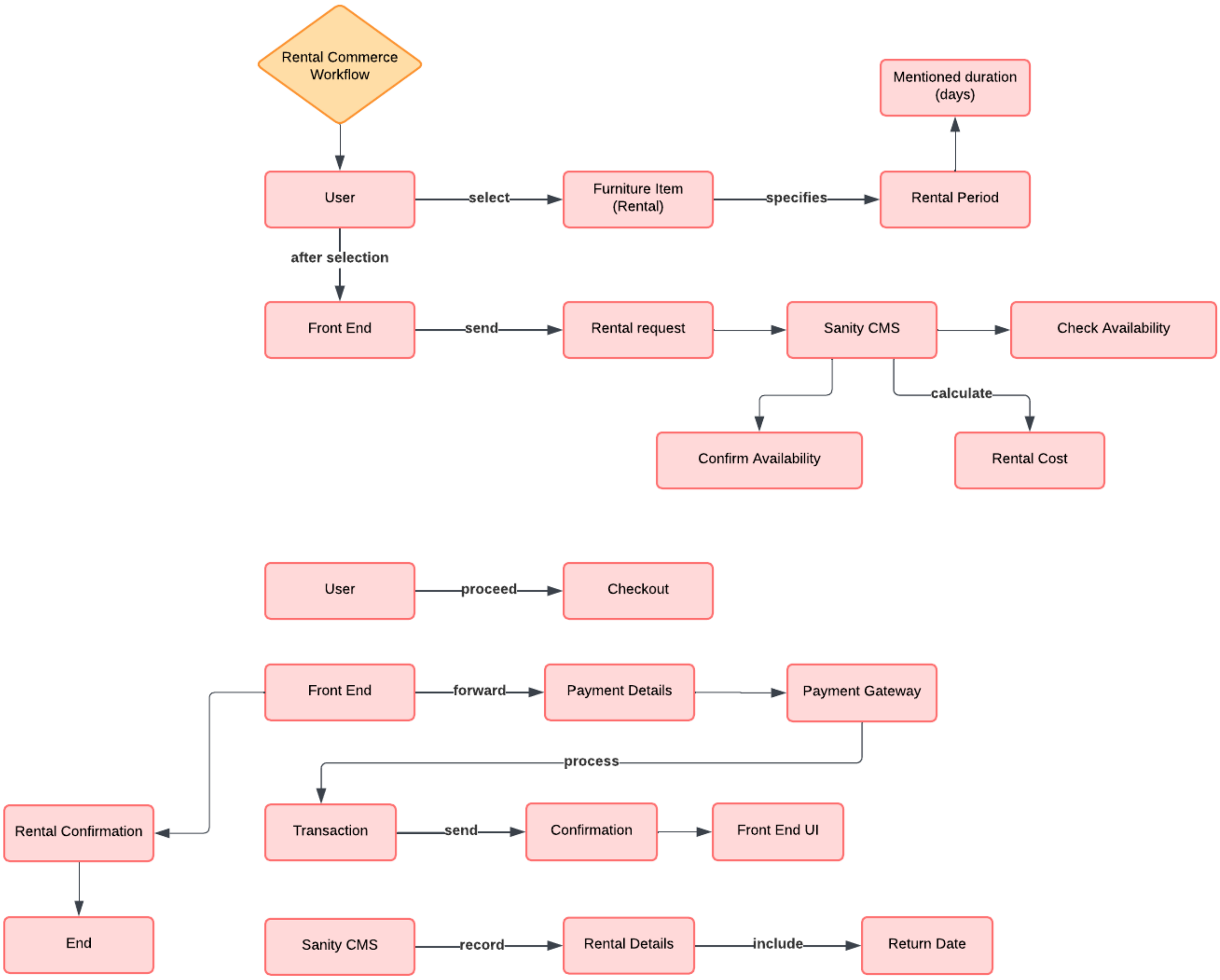


E-Commerce Order Placement Workflow:

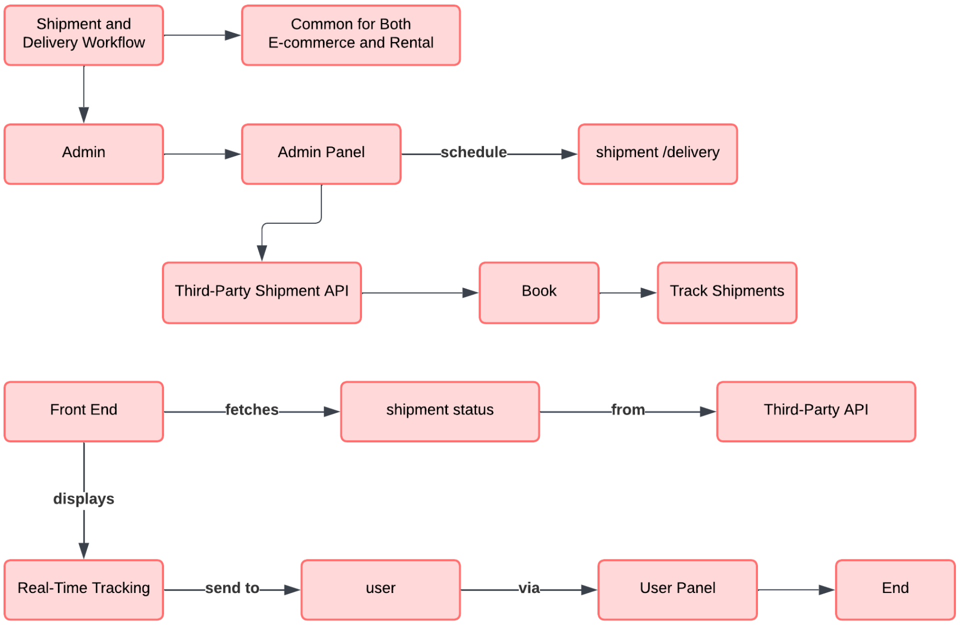


289436 - Sadaf Shahab - (Monday 2:00 to 5:00)

Rental Commerce Workflow:

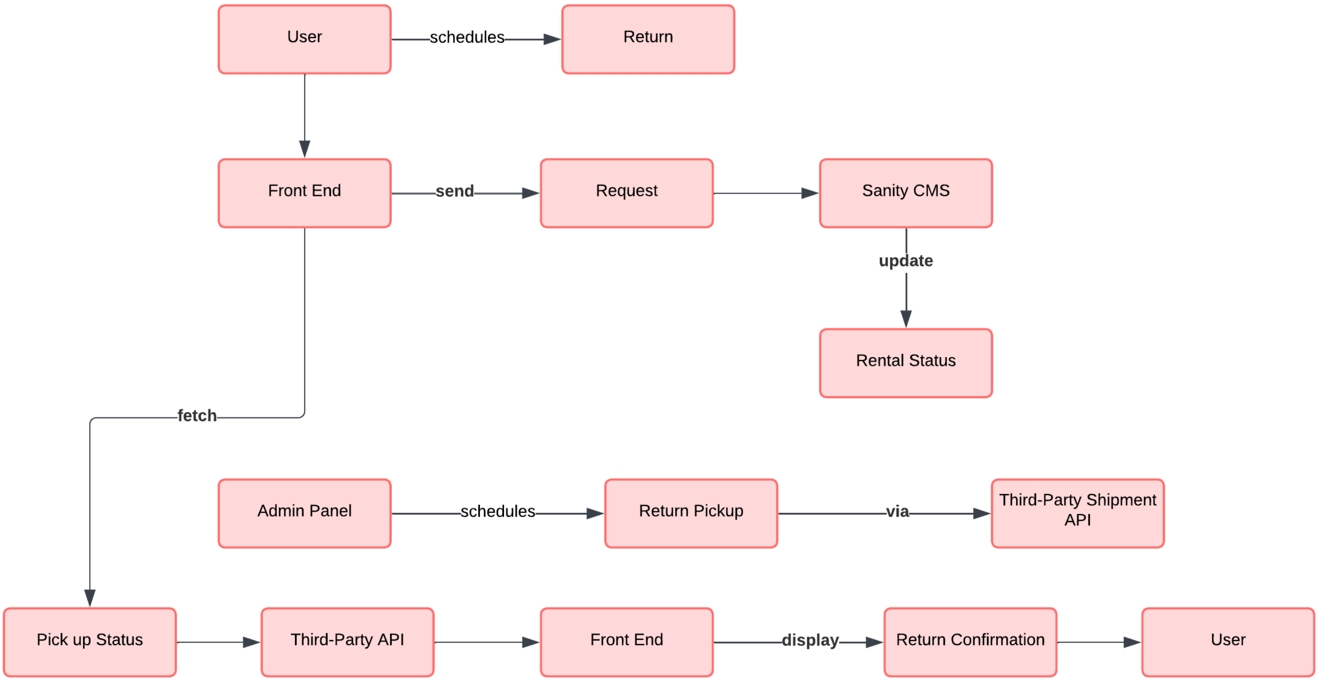


Shipment and Delivery Workflow:

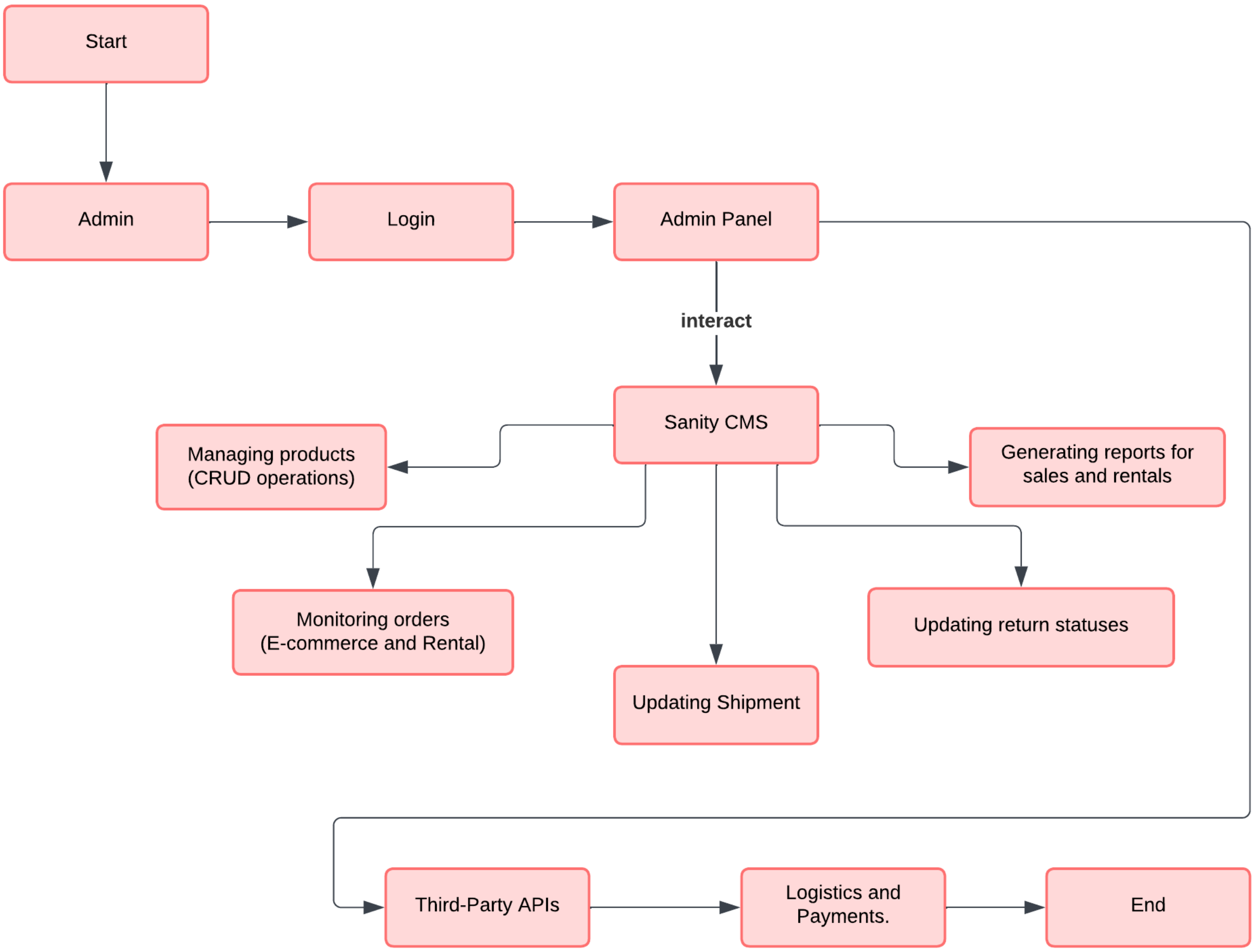


289436 - Sadaf Shahab - (Monday 2:00 to 5:00)

Return Workflow (For Rentals Only):



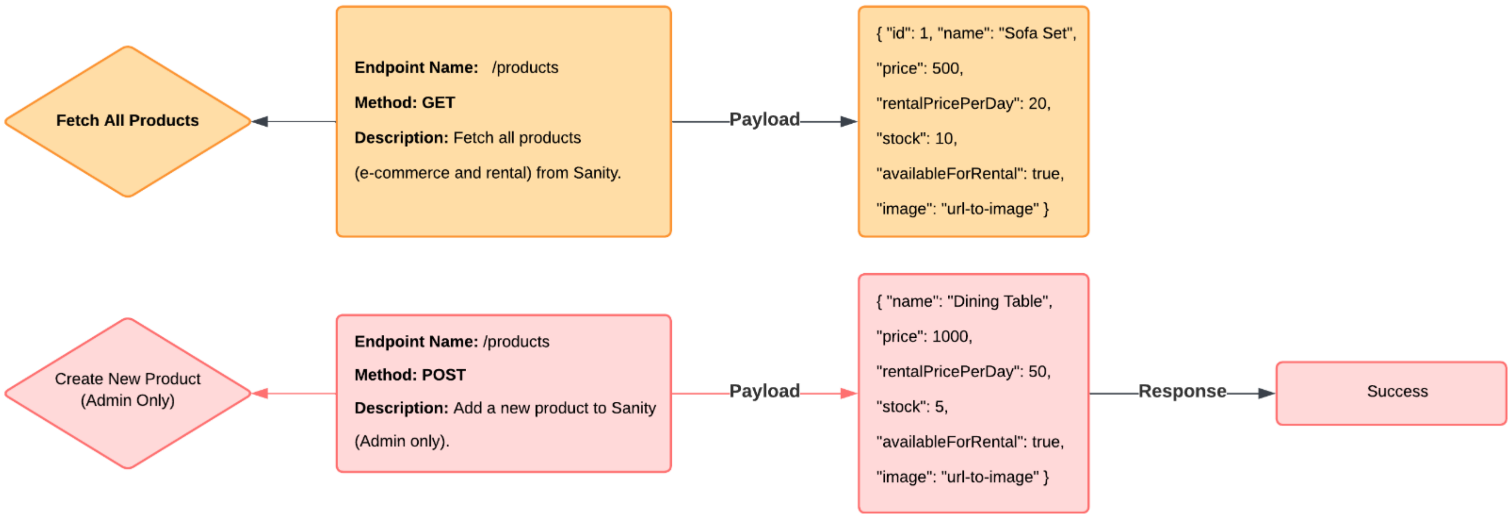
Admin Management Workflow:



289436 - Sadaf Shahab - (Monday 2:00 to 5:00)

Plan API Requirements

Product Management



Fetch All Products

Endpoint Name: /products
Method: GET
Description: Fetch all products (e-commerce and rental) from Sanity.

Payload

```
{ "id": 1, "name": "Sofa Set",
"price": 500,
"rentalPricePerDay": 20,
"stock": 10,
"availableForRental": true,
"image": "url-to-image" }
```

Create New Product (Admin Only)

Endpoint Name: /products
Method: POST
Description: Add a new product to Sanity (Admin only).

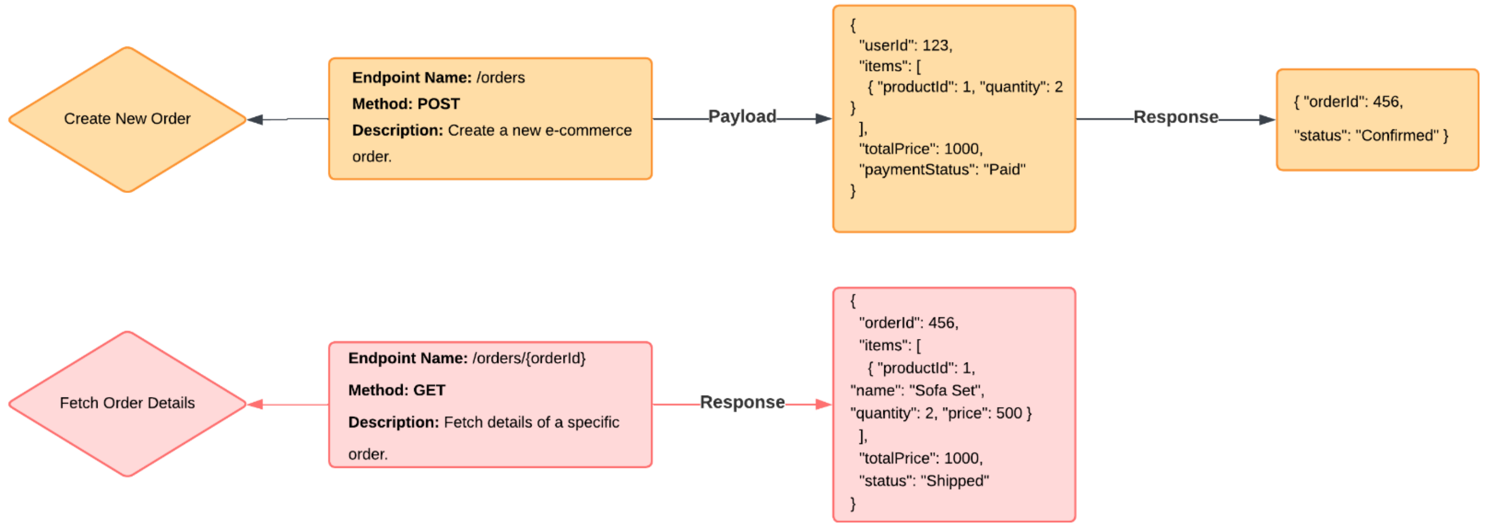
Payload

```
{ "name": "Dining Table",
"price": 1000,
"rentalPricePerDay": 50,
"stock": 5,
"availableForRental": true,
"image": "url-to-image" }
```

Response

Success

Order Management



Create New Order

Endpoint Name: /orders
Method: POST
Description: Create a new e-commerce order.

Payload

```
{
  "userId": 123,
  "items": [
    { "productId": 1, "quantity": 2
}
  ],
  "totalPrice": 1000,
  "paymentStatus": "Paid"
}
```

Response

```
{ "orderId": 456,
"status": "Confirmed" }
```

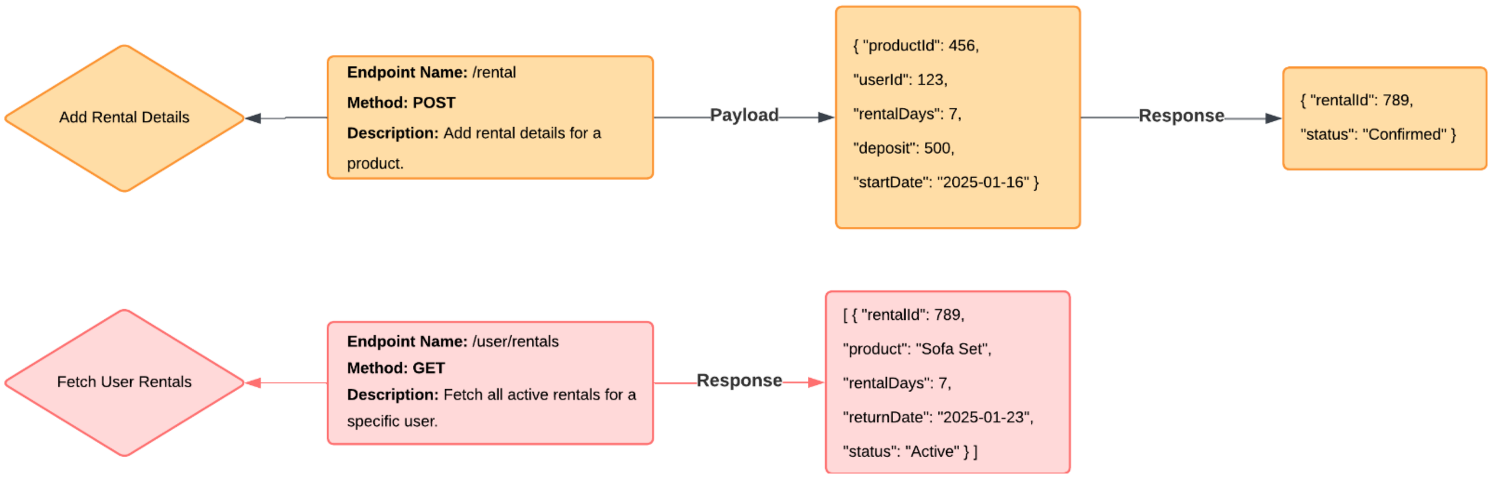
Fetch Order Details

Endpoint Name: /orders/{orderId}
Method: GET
Description: Fetch details of a specific order.

Response

```
{
  "orderId": 456,
  "items": [
    { "productId": 1,
"name": "Sofa Set",
"quantity": 2, "price": 500 }
  ],
  "totalPrice": 1000,
  "status": "Shipped"
}
```

Rental Management



Add Rental Details

Endpoint Name: /rental
Method: POST
Description: Add rental details for a product.

Payload

```
{ "productId": 456,
"userId": 123,
"rentalDays": 7,
"deposit": 500,
"startDate": "2025-01-16" }
```

Response

```
{ "rentalId": 789,
"status": "Confirmed" }
```

Fetch User Rentals

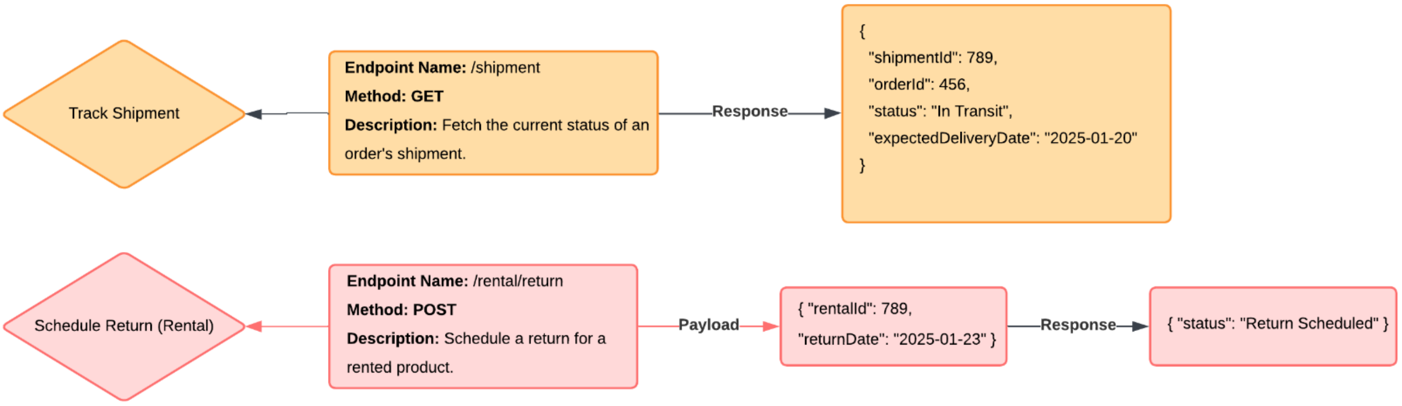
Endpoint Name: /user/rentals
Method: GET
Description: Fetch all active rentals for a specific user.

Response

```
[ { "rentalId": 789,
"product": "Sofa Set",
"rentalDays": 7,
"returnDate": "2025-01-23",
"status": "Active" } ]
```

289436 - Sadaf Shahab - (Monday 2:00 to 5:00)

Shipment and Returns



Endpoint Name: /shipment
Method: GET
Description: Fetch the current status of an order's shipment.

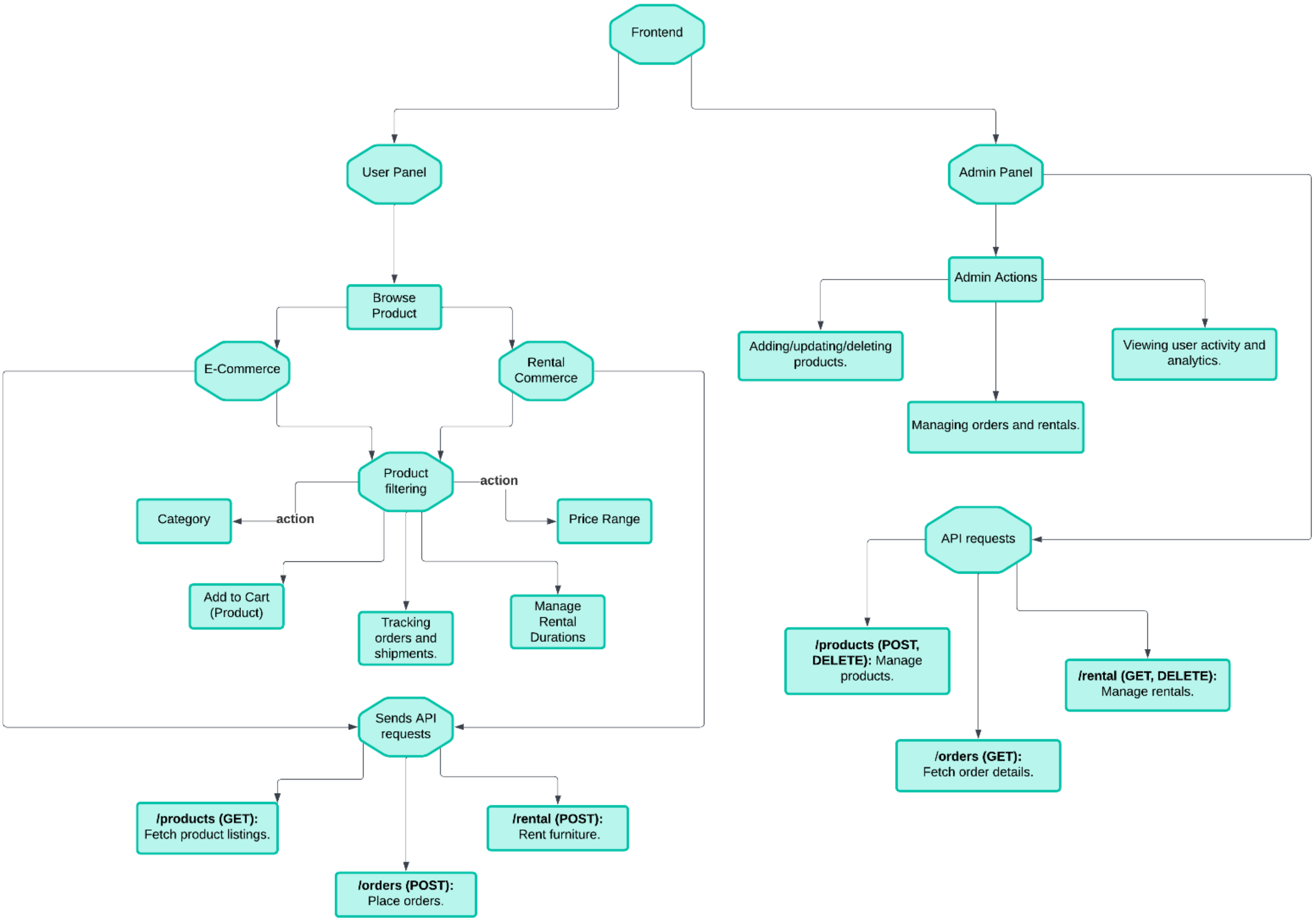
```
{
  "shipmentId": 789,
  "orderId": 456,
  "status": "In Transit",
  "expectedDeliveryDate": "2025-01-20"
}
```

Endpoint Name: /rental/return
Method: POST
Description: Schedule a return for a rented product.

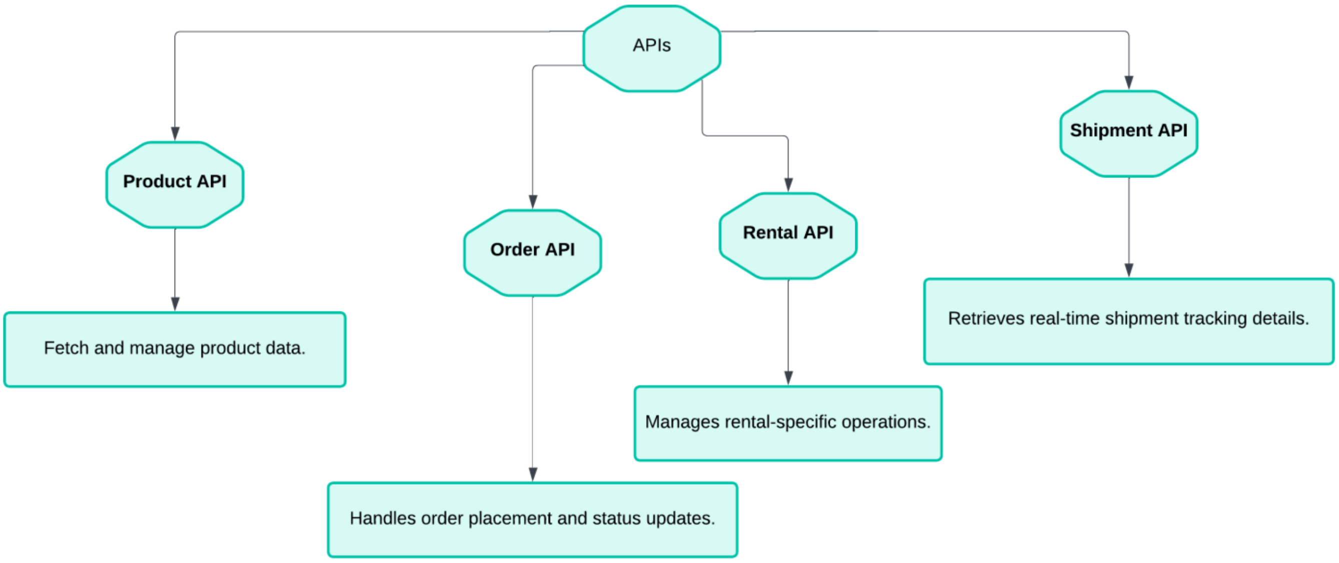
```
{ "rentalId": 789,
"returnDate": "2025-01-23" }
```

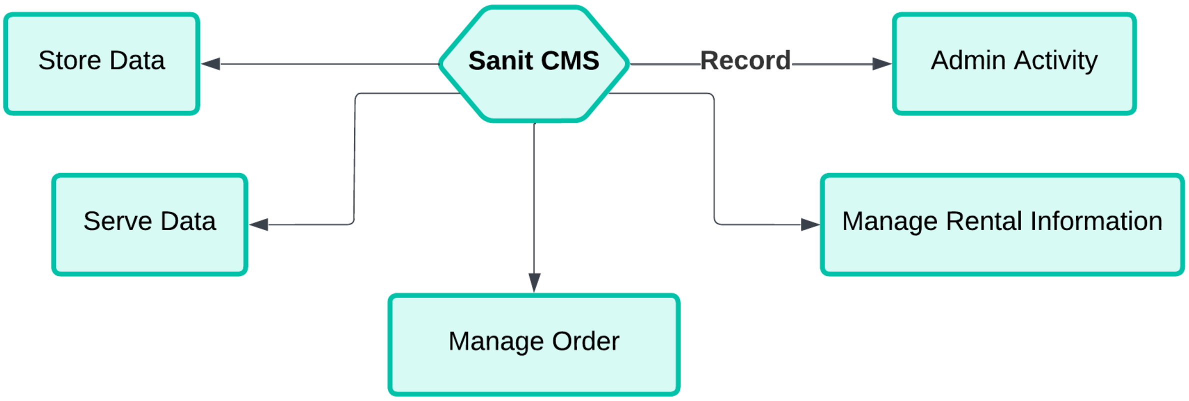
```
{ "status": "Return Scheduled" }
```

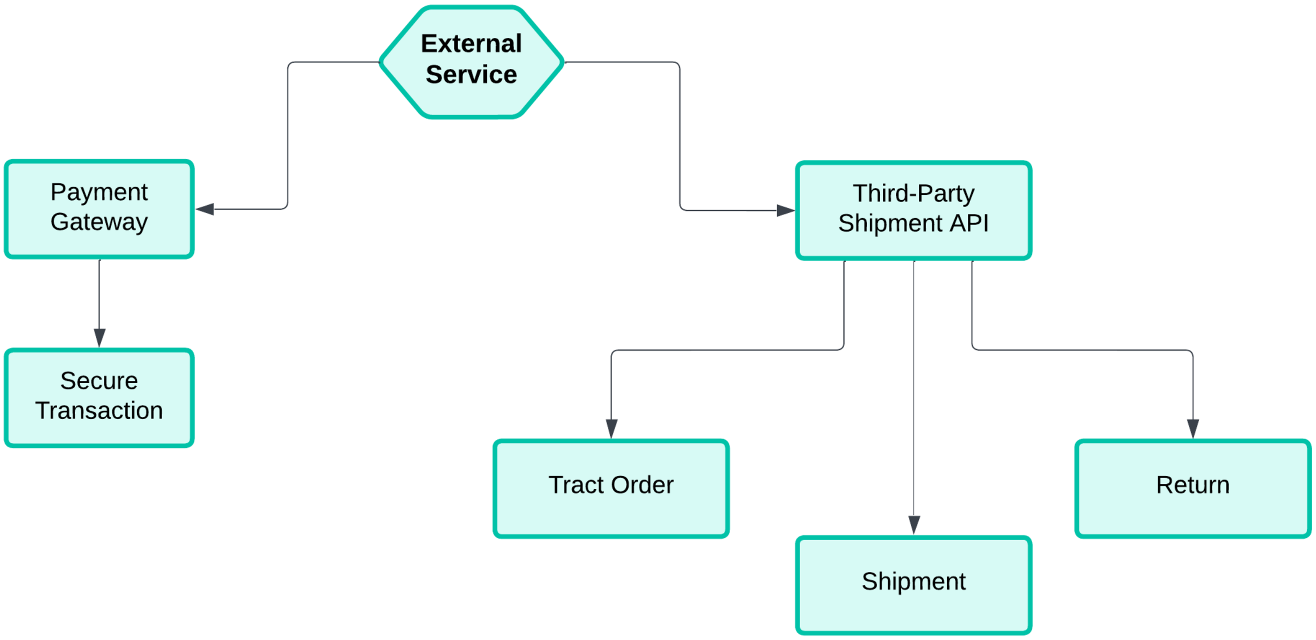
System Architecture Overview


Frontend
User Panel
Admin Panel
Browse Product
Admin Actions
Adding/updating/deleting products.
Viewing user activity and analytics.
Managing orders and rentals.
E-Commerce
Rental Commerce
Product filtering
action
action
Category
Price Range
Add to Cart (Product)
Tracking orders and shipments.
Manage Rental Durations
API requests
/products (POST, DELETE): Manage products.
/rental (GET, DELETE): Manage rentals.
/orders (GET): Fetch order details.
Sends API requests
/products (GET): Fetch product listings.
/rental (POST): Rent furniture.
/orders (POST): Place orders.

289436 - Sadaf Shahab - (Monday 2:00 to 5:00)


APIs
Product API
Order API
Rental API
Shipment API
Fetch and manage product data.
Handles order placement and status updates.
Manages rental-specific operations.
Retrieves real-time shipment tracking details.



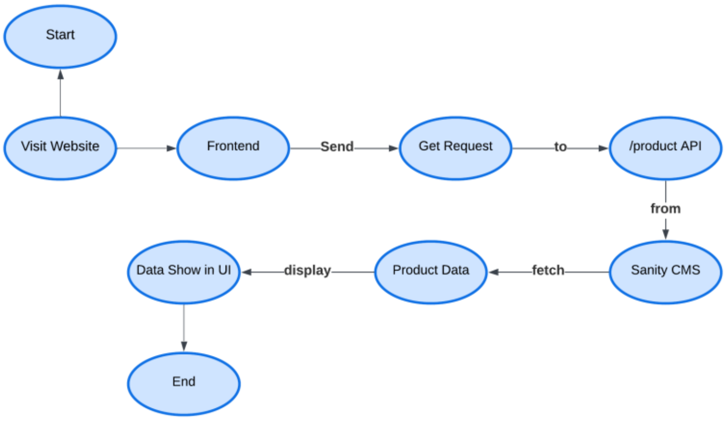


2. Key Workflows:

User Workflows

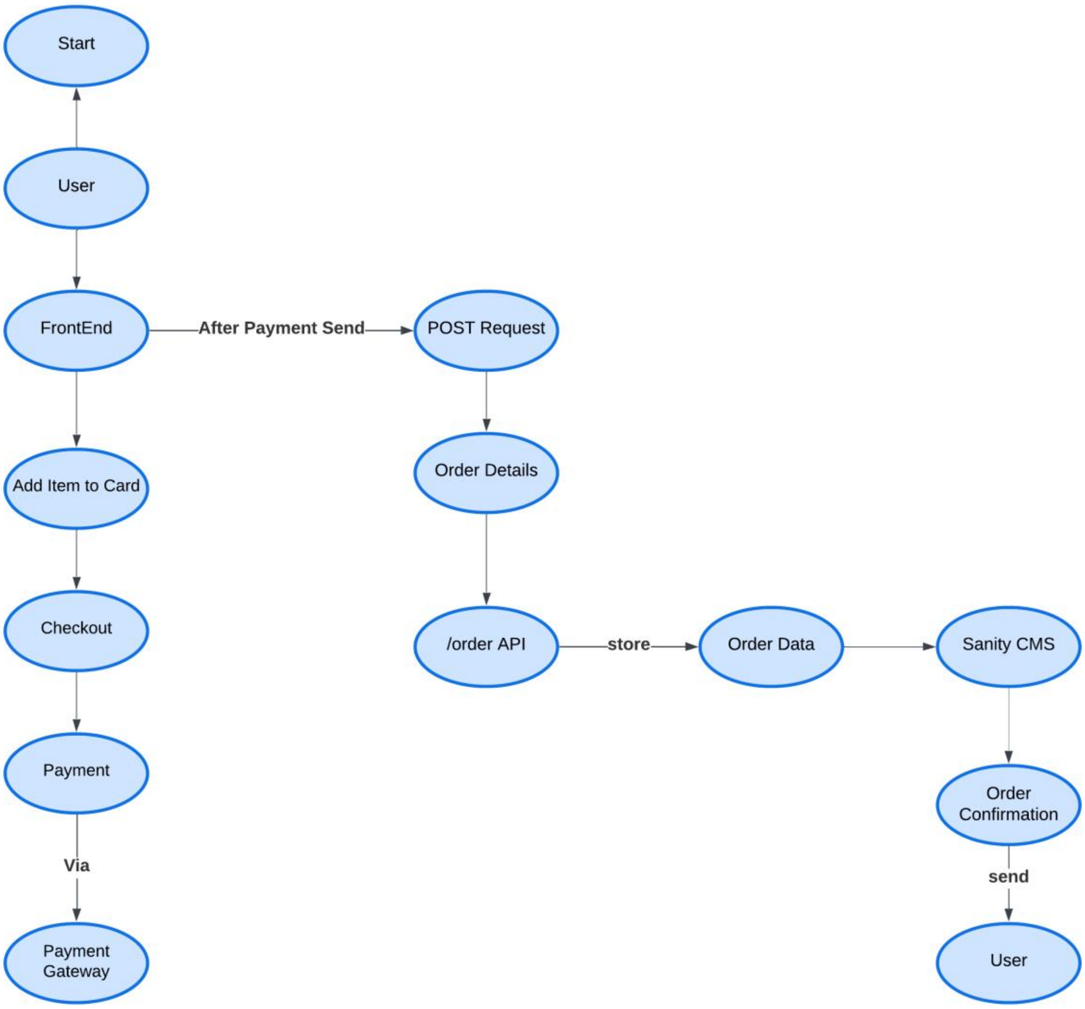
***Product Browsing**:*

1. *User visits the website.*
2. *Frontend sends a GET request to /products API.*
3. *API fetches product data from Sanity CMS.*
4. *Data is displayed on the website for user interaction.*



***Placing an Order**:*

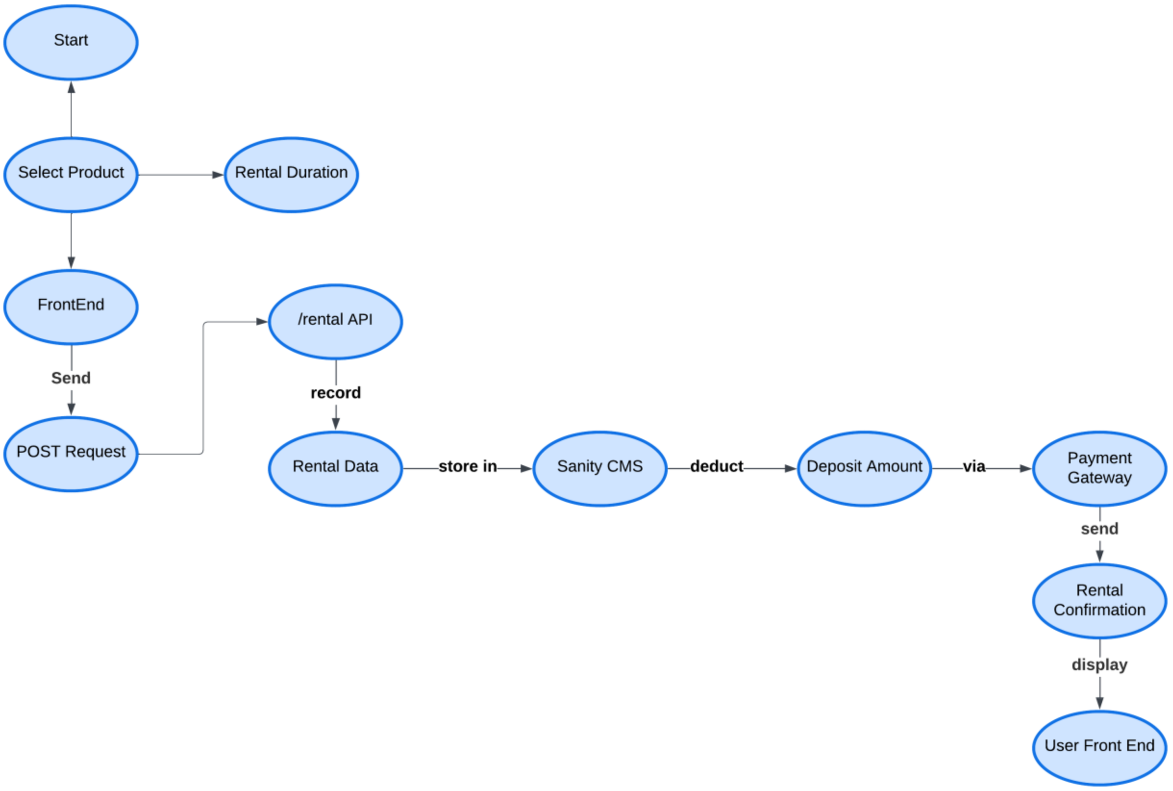
1. *User adds items to cart.*
2. *User proceeds to checkout and completes payment via Payment Gateway.*
3. *Frontend sends a POST request to /orders with order details.*
4. *API stores order information in Sanity CMS.*
5. *Confirmation is displayed to the user.*



289436 - Sadaf Shahab - (Monday 2:00 to 5:00)

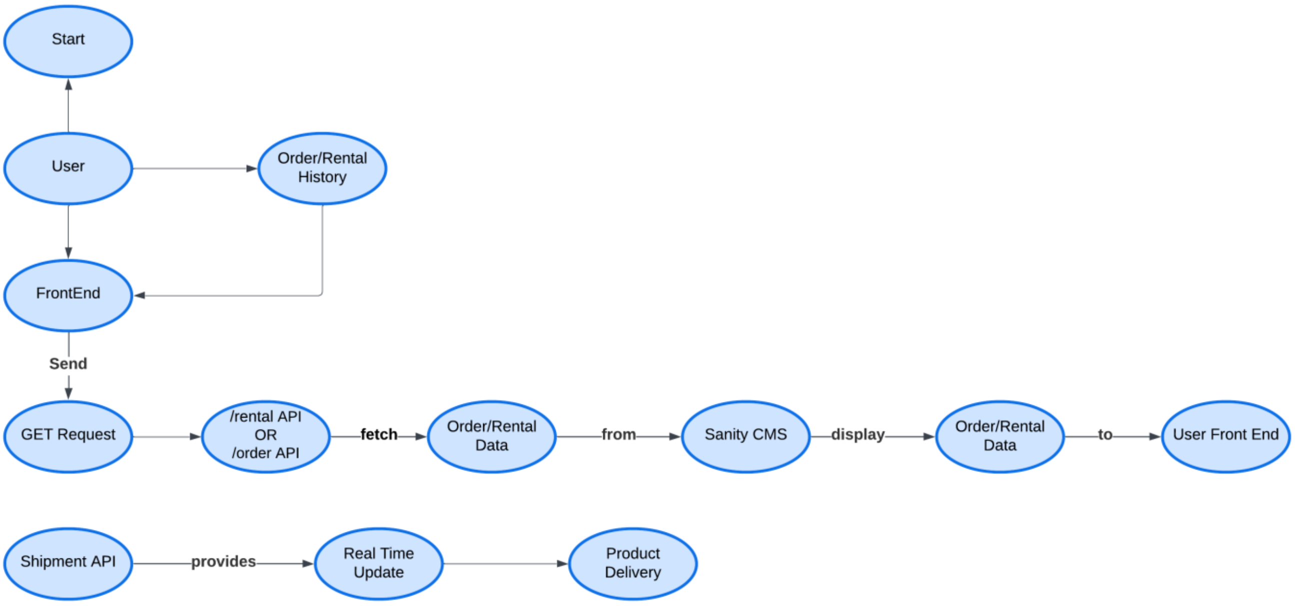
Renting Furniture:

1. *User selects a product and rental duration.*
2. *Frontend sends a POST request to /rental API.*
3. *API records rental details in Sanity CMS and deducts the deposit amount via Payment Gateway.*
4. *Rental confirmation is displayed to the user.*



Tracking Orders and Rentals:

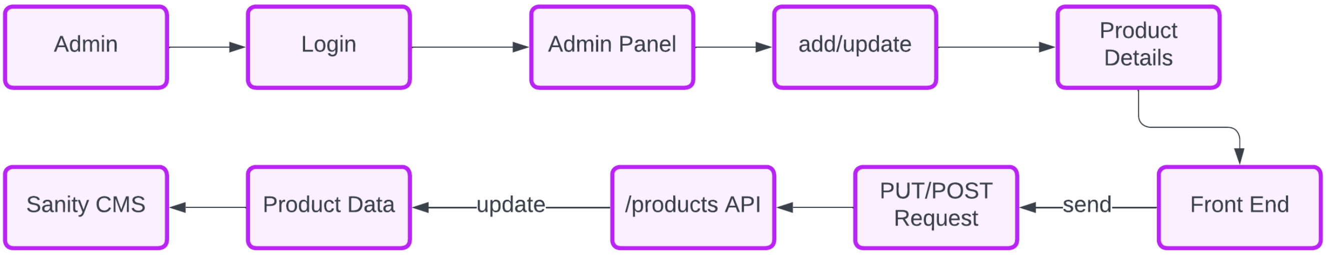
1. *User accesses their order/rental history.*
2. *Frontend sends GET requests to /orders or /rental.*
3. *API retrieves and displays order or rental data.*
4. *Shipment API is used to provide real-time updates for deliveries.*



Admin Workflows

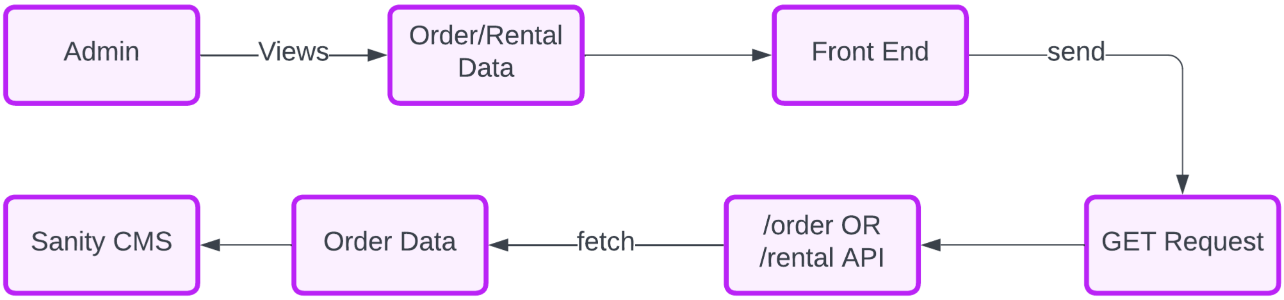
Managing Products:

1. *Admin logs into the admin panel.*
2. *Admin adds or updates product details.*
3. *Frontend sends POST/PUT requests to /products API.*
4. *API updates product data in Sanity CMS.*



Monitoring Orders and Rentals:

1. *Admin views order or rental data.*
2. *Frontend sends GET requests to /orders or /rental.*
3. *API retrieves data from Sanity CMS for admin action.*



3. **Category-Specific Instructions**

Rental E-Commerce:

- Include workflows for rental duration and deposit management.
- **Example schema field:**
  - Rental Duration: Specifies the rental period (e.g., "7 days").
  - Deposit Amount: Stores the refundable deposit.
  - Condition Status: Tracks the condition of returned items.

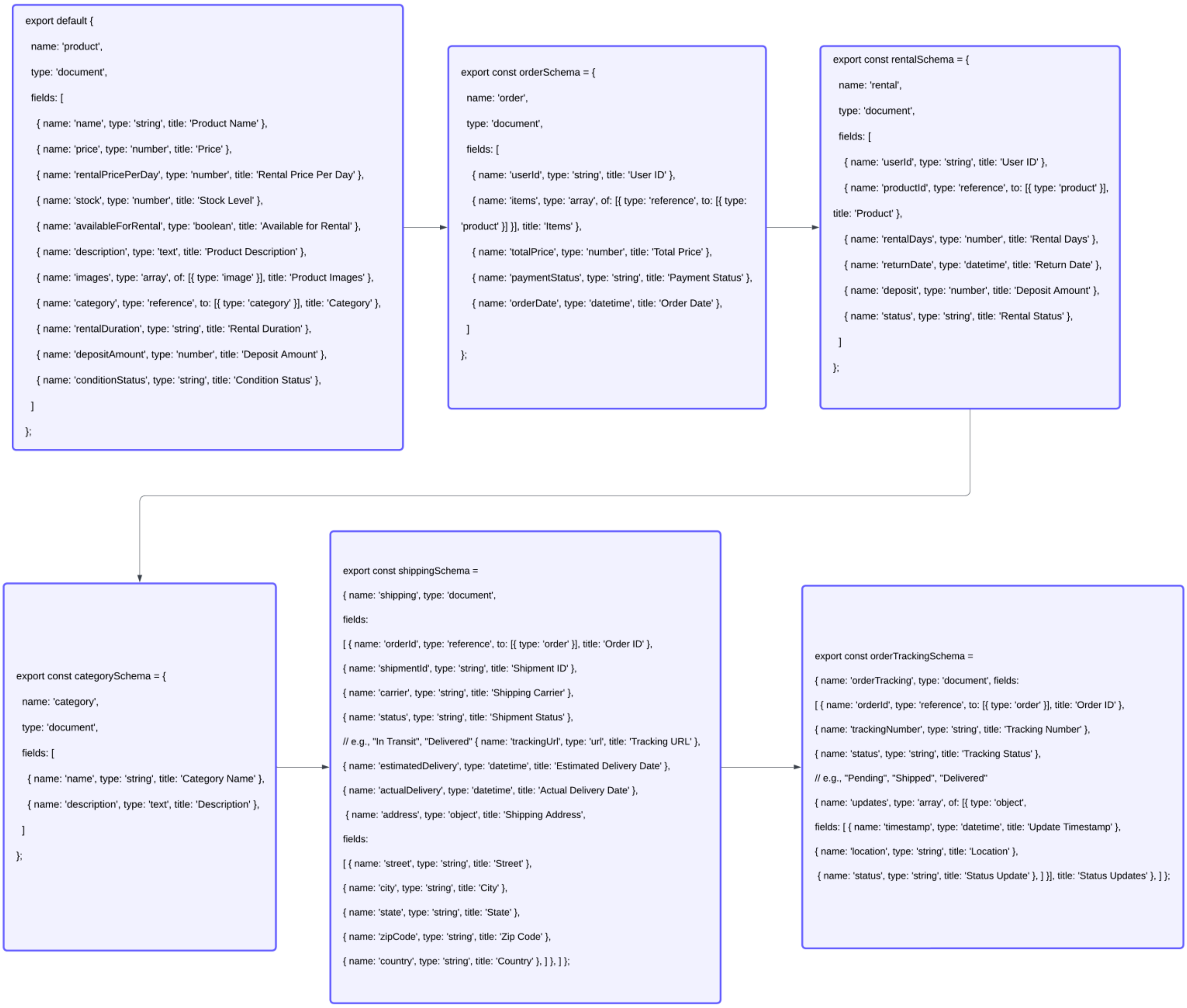
General E-Commerce

- Focus on product browsing, cart management, and order placement workflows.
- Example endpoint: /products to fetch available items.

4. API Endpoints

Endpoint	Method	Purpose	Payload	Response Example
/products	GET	Fetches all product details	None	{ "id": 1, "name": "Table", "price": 500 }
/orders	POST	Creates a new order	{ "userId": 123, "items": [{...}] }	{ "orderId": 456, "status": "Success" }
/rental	POST	Adds rental details	{ "productId": 789, "duration": "7" }	{ "confirmationId": 321, "status": "Confirmed" }
/shipment	GET	Fetches shipment tracking data	{ "orderId": 456 }	{ "status": "In Transit", "ETA": "2 days" }

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: 'rentalPricePerDay', type: 'number', title: 'Rental Price Per Day' },
    { name: 'stock', type: 'number', title: 'Stock Level' },
    { name: 'availableForRental', type: 'boolean', title: 'Available for Rental' },
    { name: 'description', type: 'text', title: 'Product Description' },
    { name: 'images', type: 'array', of: [{ type: 'image' }], title: 'Product Images' },
    { name: 'category', type: 'reference', to: [{ type: 'category' }], title: 'Category' },
    { name: 'rentalDuration', type: 'string', title: 'Rental Duration' },
    { name: 'depositAmount', type: 'number', title: 'Deposit Amount' },
    { name: 'conditionStatus', type: 'string', title: 'Condition Status' },
  ]
};
```

```
export const orderSchema = {
  name: 'order',
  type: 'document',
  fields: [
    { name: 'userId', type: 'string', title: 'User ID' },
    { name: 'items', type: 'array', of: [{ type: 'reference', to: [{ type:
'product' }] }], title: 'Items' },
    { name: 'totalPrice', type: 'number', title: 'Total Price' },
    { name: 'paymentStatus', type: 'string', title: 'Payment Status' },
    { name: 'orderDate', type: 'datetime', title: 'Order Date' },
  ]
};
```

```
export const rentalSchema = {
  name: 'rental',
  type: 'document',
  fields: [
    { name: 'userId', type: 'string', title: 'User ID' },
    { name: 'productId', type: 'reference', to: [{ type: 'product' }],
title: 'Product' },
    { name: 'rentalDays', type: 'number', title: 'Rental Days' },
    { name: 'returnDate', type: 'datetime', title: 'Return Date' },
    { name: 'deposit', type: 'number', title: 'Deposit Amount' },
    { name: 'status', type: 'string', title: 'Rental Status' },
  ]
};
```

```
export const categorySchema = {
  name: 'category',
  type: 'document',
  fields: [
    { name: 'name', type: 'string', title: 'Category Name' },
    { name: 'description', type: 'text', title: 'Description' },
  ]
};
```

```
export const shippingSchema =
{ name: 'shipping', type: 'document',
fields:
[ { name: 'orderId', type: 'reference', to: [{ type: 'order' }], title: 'Order ID' },
{ name: 'shipmentId', type: 'string', title: 'Shipment ID' },
{ name: 'carrier', type: 'string', title: 'Shipping Carrier' },
{ name: 'status', type: 'string', title: 'Shipment Status' },
// e.g., "In Transit", "Delivered" { name: 'trackingUrl', type: 'url', title: 'Tracking URL' },
{ name: 'estimatedDelivery', type: 'datetime', title: 'Estimated Delivery Date' },
{ name: 'actualDelivery', type: 'datetime', title: 'Actual Delivery Date' },
 { name: 'address', type: 'object', title: 'Shipping Address',
fields:
[ { name: 'street', type: 'string', title: 'Street' },
{ name: 'city', type: 'string', title: 'City' },
{ name: 'state', type: 'string', title: 'State' },
{ name: 'zipCode', type: 'string', title: 'Zip Code' },
{ name: 'country', type: 'string', title: 'Country' }, ] }, ] };
```

```
export const orderTrackingSchema =
{ name: 'orderTracking', type: 'document', fields:
[ { name: 'orderId', type: 'reference', to: [{ type: 'order' }], title: 'Order ID' },
{ name: 'trackingNumber', type: 'string', title: 'Tracking Number' },
{ name: 'status', type: 'string', title: 'Tracking Status' },
// e.g., "Pending", "Shipped", "Delivered"
{ name: 'updates', type: 'array', of: [{ type: 'object',
fields: [ { name: 'timestamp', type: 'datetime', title: 'Update Timestamp' },
{ name: 'location', type: 'string', title: 'Location' },
 { name: 'status', type: 'string', title: 'Status Update' }, ] }], title: 'Status Updates' }, ] };
```

289436 - Sadaf Shahab - (Monday 2:00 to 5:00)

Technical Roadmap for Furniture Marketplace

Week 1: Planning & Design

- **Deliverables:**
  - Finalize business requirements and technical specifications.
  - Design wireframes and UI/UX for:
    - Homepage
    - Product listing and details pages
    - Cart and checkout
    - Rental management
    - Admin dashboard
  - Create a detailed **System Architecture Diagram**:
    - Frontend (Next.js)
    - Backend (Node.js with APIs)
    - Sanity CMS integration
    - ShipEngine for shipment tracking.
  - Plan database schemas for:
    - Products, rentals, orders, users, and shipment tracking.

Week 2: Frontend Development - Core UI

- **Deliverables:**
  - Build responsive UI components using **Next.js** and **Tailwind CSS**:
    - Homepage: Showcase featured furniture and categories.
    - Product Pages:
      - General product details.
      - Rental-specific information (e.g., rental duration, deposit).
    - Cart and Checkout:
      - Enable add-to-cart, rental duration selection, and checkout flows.
    - Order Tracking Page:
      - Display real-time tracking status (integrated with ShipEngine).
  - Implement **React Context API** or **Zustand** for state management.
  - Integrate **React Hook Form + Zod** for input validation in forms.

Week 3: Backend Development - API & CMS

- **Deliverables:**
  - Develop backend API:
    - /products: Fetch product and rental information.

        - /orders: Create and manage orders.
        - /rental: Handle rental-specific data like duration, deposits, and condition tracking.
        - /shipment: Integrate ShipEngine for order tracking.
    - Integrate **Sanity CMS**:
        - Set up schemas for products, orders, rentals, and tracking details.
        - Build APIs to fetch/update CMS data.
    - Implement authentication (e.g., Firebase or Clerk):
        - User login/signup for customers.
        - Admin authentication for product/order management.

Week 4: Integration - Frontend with Backend

- **Deliverables:**
    - Connect frontend UI components to backend APIs:
        - Product listing page fetches data from /products.
        - Cart and checkout send order data to /orders.
        - Order tracking page fetches data from /shipment.
    - Add error handling and loading states for APIs.
    - Test rental-specific flows:
        - Verify deposit and duration handling.

Week 5: Shipment & Order Tracking

- **Deliverables:**
    - Integrate **ShipEngine API**:
        - Configure API to create shipment labels and retrieve tracking information.
        - Add /shipment endpoint to backend for:
            - Tracking shipment status (e.g., "In Transit," "Delivered").
            - Estimating delivery time.
    - Display real-time tracking updates on the **Order Tracking Page**.
    - Implement email notifications:
        - Send order confirmation and shipment updates to users.

Week 6: Admin Panel Development

- **Deliverables:**
    - Build a responsive admin panel using Next.js:
        - Product Management: Add/edit/delete products.
        - Order Management: View and update order statuses.

    - Rental Management: Monitor rental returns and condition tracking.
  - Integrate analytics for admin dashboard:
    - View sales data, rental trends, and user activity.

Week 7: Testing & Optimization

- **Deliverables:**
  - Conduct end-to-end testing:
    - Verify UI flows for both general purchases and rentals.
    - Ensure accurate shipment tracking and order updates.
  - Optimize API performance:
    - Add caching for frequently accessed data (e.g., products).
  - Test for scalability:
    - Simulate high traffic for product and order APIs.
  - Fix UI/UX bugs identified during testing.

Week 8: Deployment & Launch

- **Deliverables:**
  - Deploy frontend and backend:
    - Use **Vercel** for Next.js frontend.
    - Use **Sanity CMS** as backend.
  - Set up CI/CD pipelines for seamless updates.?
  - Launch the platform:
    - Announce the launch and onboard initial users.

Post-Launch Maintenance

- Monitor system performance and fix bugs.
- Roll out enhancements:
  - Add advanced filters and sorting options.
  - Introduce loyalty programs or subscription models for frequent renters.