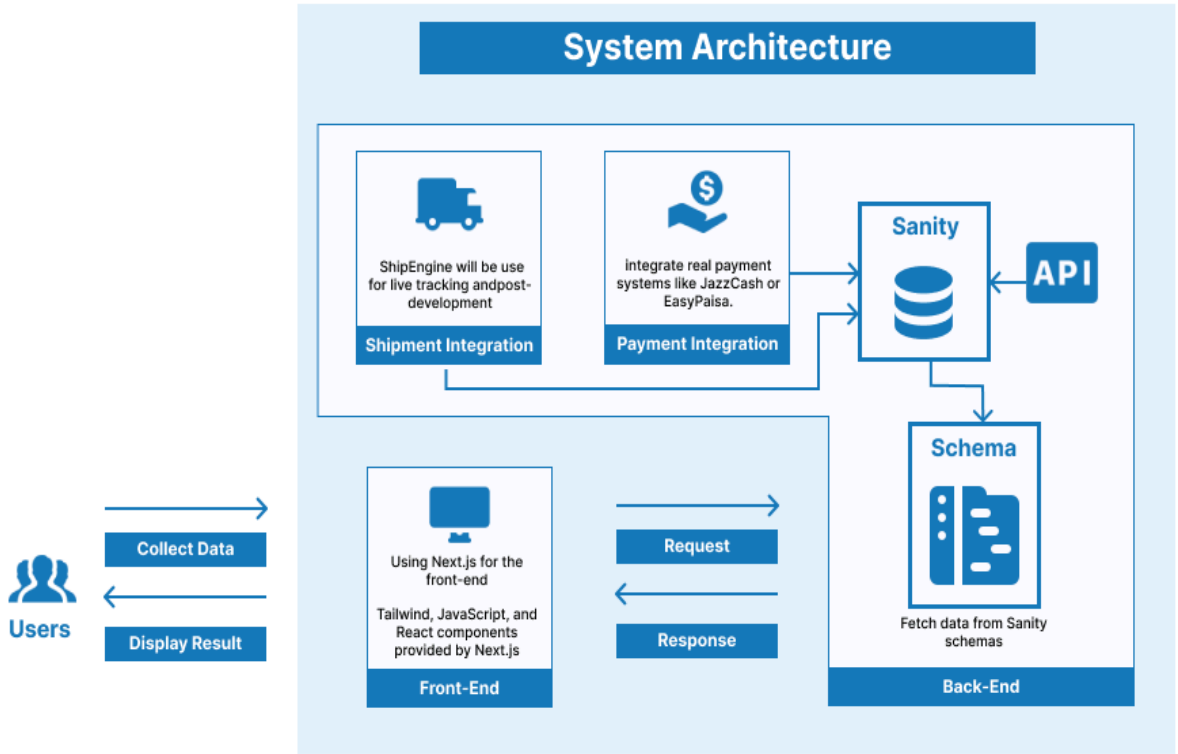
Technical Plan for Rental Furniture Marketplace

1. Technical Requirements

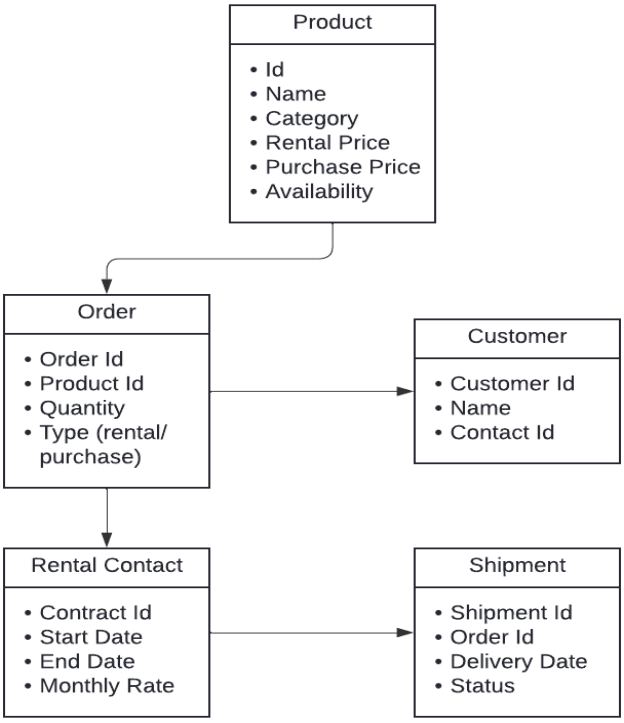
- **Frontend Requirements:**
  - User-friendly interface for browsing and customizing furniture.
  - Responsive design for both mobile and desktop users.
  - Pages: Home, Product Listing, Product Details, Rental Options, Cart, Checkout, and Order Confirmation.
- **Backend Requirements:**
  - Use Sanity CMS for:
    - Managing product inventory (rental and purchase).
    - Storing customer data and order records.
- **Third-Party APIs:**
  - Payment Gateway (e.g., jazzCash EasyPaisa) for secure transactions.
  - Shipment Tracking API for real-time order tracking.

2. System Architecture

Architecture Diagram:



Data Schema Design

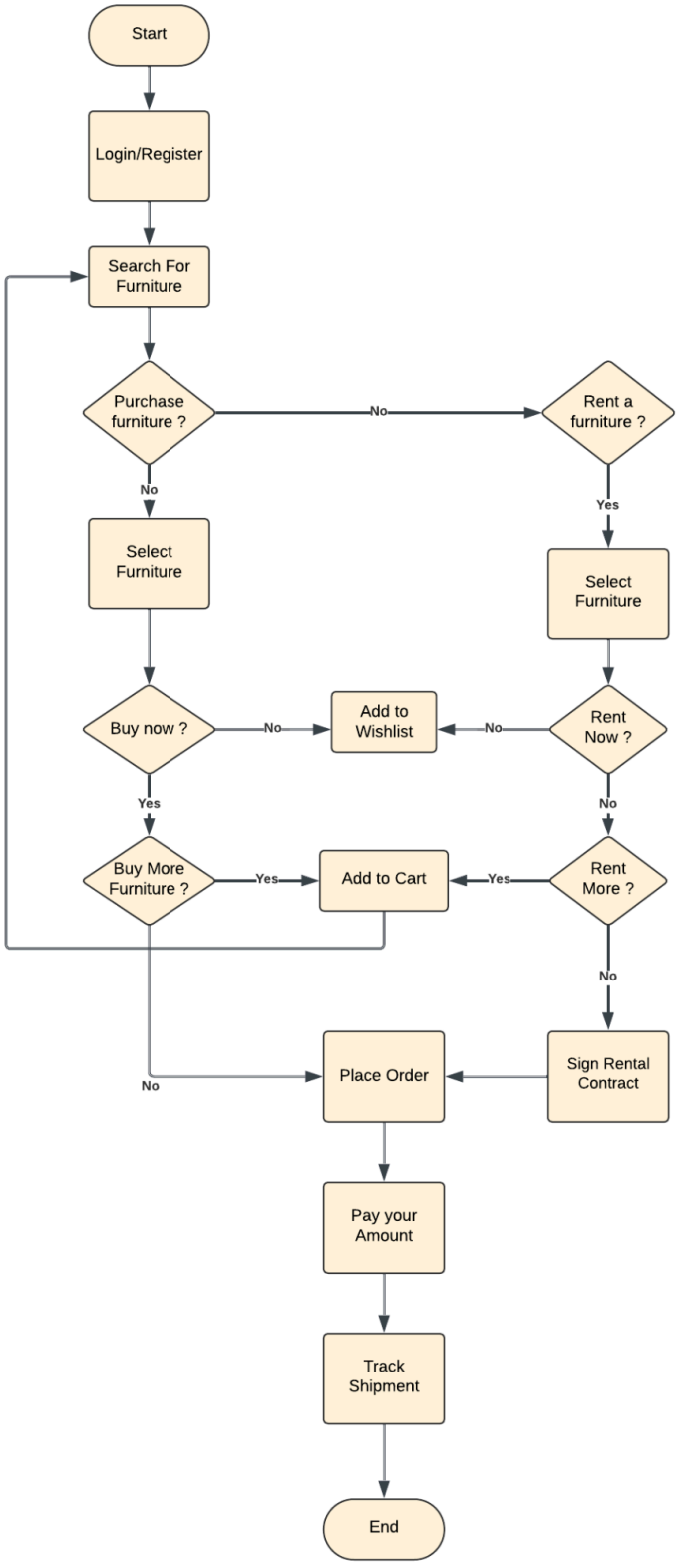


Workflows:

Key Workflows to Include:

1. **User Registration:**
   - User signs up -> Data is stored in Sanity CMS -> Confirmation email sent to the user.
2. **Product Browsing:**
   - User navigates to product categories (Rental or Purchase) -> Sanity API fetches product details -> Products are displayed dynamically on the front-end.
3. **Product Search and Filtering:**
   - User searches or applies filters (e.g., category, price, rental terms) -> Sanity API processes query -> Relevant products are displayed on the front-end.
4. **Order Placement:**
   - User adds items to the cart -> Proceeds to checkout -> Payment processed through a fake payment API or JazzCash/EasyPaisa -> Order details and payment status are stored in Sanity CMS.
5. **Shipment Tracking:**
   - Shipment details are sent to the shipment API (fake API during development, Shippo/ShipEngine in production) -> Order status updates fetched -> Tracking information displayed to the user.
6. **Payment Processing:**
   - User selects a payment option (JazzCash, EasyPaisa, or credit card) -> Payment API validates the transaction -> Transaction status is saved in Sanity CMS.

WorkFlow Diagram:



3. API Requirements

Endpoint	Method	Purpose	Payload/Response
/products	GET	Fetch all available furniture products.	Response: { "id": 1, "name": "Sofa", "rentalPrice": 100, "purchasePrice": 500 }
/orders	POST	Create a new order.	Payload: { "customerId": 123, "productId": 456, "type": "rental", "duration": "7 days" }
/rental-duration	POST	Add rental-specific details for an order.	Payload: { "orderId": 789, "startDate": "2025-01-20", "endDate": "2025-01-27" }
/shipment	GET	Fetch real-time shipment tracking details.	Response: { "shipmentId": 101, "status": "In Transit", "expectedDelivery": "Jan 21" }
/payment-status	GET	Fetch payment confirmation status.	Response: { "orderId": 789, "status": "Paid" }

4. Sanity Schema Design

Product Schema:

```
export default {
  name: 'product',
  type: 'document',
  fields: [
    { name: 'name', type: 'string', title: 'Product Name' },
    { name: 'rentalPrice', type: 'number', title: 'Rental Price' },
    { name: 'purchasePrice', type: 'number', title: 'Purchase Price'
},
    { name: 'category', type: 'string', title: 'Category' },
    { name: 'condition', type: 'string', title: 'Condition
(New/Refurbished)' },
  ],
};
```

Order Schema:

```
export default {
  name: 'order',
  type: 'document',
  fields: [
    { name: 'customerId', type: 'string', title: 'Customer ID' },
    { name: 'productId', type: 'string', title: 'Product ID' },
    { name: 'type', type: 'string', title: 'Order Type
(Rental/Purchase)' },
    { name: 'duration', type: 'string', title: 'Rental Duration' },
    { name: 'status', type: 'string', title: 'Order Status' },
  ],
};
```

5. Technical Roadmap

1: Planning & Setup

- Define workflows and requirements.

2: Frontend Development

- Build responsive UI using Next.js and Tailwind CSS.
- Develop key pages: Home, Product Listing, User Auth, Cart, and Checkout.
- Integrate fake API.

3: Backend Development

- Connect Sanity CMS for data handling.
- Set up APIs for product data, and orders.

4: CMS Configuration

- Design schemas in Sanity for products, users, and orders.
- Populate data for development.

5: Payment & Shipment (Week 5-6)

- Integrate 3rd party APIs for payment such as JazzCash/EasyPaisa and ShipEngine.