

# Marketplace Technical Documentation – Rent-a-Read

---

## 1. System Architecture Overview

### Architecture Diagram

```
[Frontend (Next.js)]
    |
[Sanity CMS] -----> [Rental Data API]
    |
[Payment Gateway] -----> [Deposit and Rental Payments]
    |
[Third-Party API] -----> [Rental Notifications/Reminders]
```

### Component Descriptions

- **Frontend (Next.js):**
    - Customer-facing interface for browsing rental products, managing bookings, and tracking orders.
  - **Sanity CMS:**
    - Backend system to store and manage rental product details, customer orders, and rental history.
  - **Payment Gateway:**
    - Handles secure payments for rental charges and deposits.
  - **Third-Party API:**
    - Sends reminders for rental deadlines, overdue items, or pickup/drop-off notifications.
- 

## 2. Key Workflows

### Workflow 1: Browsing Rental Products

1. User visits the website.
2. Frontend fetches available rental products from `/products` API endpoint.
3. Products are displayed with images, rental price, availability, and terms.

### Workflow 2: Booking a Rental

1. User selects a product and specifies rental duration.
2. Rental details are added to the cart and synced via the `/rental-cart` API.
3. At checkout, the frontend sends booking details to `/rental-order` API.
4. Sanity CMS stores the order, and a confirmation is displayed.

**Workflow 3: Rental Payment**

- 1. At checkout, the frontend integrates with the payment gateway.
- 2. Payment is processed for rental charges and deposits.
- 3. Payment status is updated in the CMS and confirmation is sent to the user.

**Workflow 4: Returning the Rental**

- 1. User initiates the return process from their account.
- 2. Condition of the item is assessed and updated via `/rental-return` API.
- 3. Refund for the deposit (if applicable) is processed through the payment gateway.

**Workflow 5: Reminders and Notifications**

- 1. Notifications for rental return deadlines are fetched via `/rental-reminders` API.
- 2. Users are reminded via email/SMS about upcoming deadlines.

**3. API Endpoints**

Endpoint	Method	Purpose	Response Example
<code>/products</code>	GET	Fetches all available rental products.	<code>{ "id": 1, "name": "Camera", "price": 50 }</code>
<code>/rental-order</code>	POST	Creates a new rental booking.	<code>{ "orderId": 123, "status": "Confirmed" }</code>
<code>/rental-cart</code>	POST	Adds rental details to the cart.	<code>{ "cartId": 234, "status": "Added" }</code>
<code>/rental-return</code>	POST	Processes a rental return.	<code>{ "returnId": 456, "status": "Completed" }</code>
<code>/rental-reminders</code>	GET	Fetches reminders for upcoming returns.	<code>{ "rentalId": 789, "dueDate": "2025-02-01" }</code>

**4. Sanity Schema Example**

**Rental Product Schema**

```
export default {
  name: 'rentalProduct',
  type: 'document',
  fields: [
    { name: 'name', type: 'string', title: 'Product Name' },
    { name: 'price', type: 'number', title: 'Rental Price per Day' },
    { name: 'deposit', type: 'number', title: 'Deposit Amount' },
    { name: 'availability', type: 'boolean', title: 'Available for Rent' },
  ],
}
```

```
    { name: 'category', type: 'string', title: 'Category' }  
  ]  
};
```

## Rental Order Schema

```
export default {  
  name: 'rentalOrder',  
  type: 'document',  
  fields: [  
    { name: 'customerId', type: 'string', title: 'Customer ID' },  
    { name: 'product', type: 'reference', to: [{ type: 'rentalProduct' }],  
      title: 'Rented Product' },  
    { name: 'rentalDuration', type: 'string', title: 'Rental Duration' },  
    { name: 'deposit', type: 'number', title: 'Deposit Paid' },  
    { name: 'rentalStatus', type: 'string', title: 'Rental Status' }  
  ]  
};
```

---

## 5. Technical Roadmap

### Phase 1: Backend Setup

- Configure Sanity CMS schemas for rental products and orders.

### Phase 2: Frontend Development

- Create pages for browsing products, managing carts, and booking rentals.

### Phase 3: Payment Integration

- Implement payment gateway for processing rental payments and deposits.

### Phase 4: API Integration

- Build and integrate `/products`, `/rental-order`, `/rental-return`, and `/rental-reminders` APIs.

### Phase 5: Notifications

- Integrate a third-party API for reminders (e.g., Twilio for SMS or SendGrid for email).

### Phase 6: Testing and Deployment

- Test all workflows end-to-end.
- Deploy the system and monitor for any bugs.