

# Overview

This Document describe the technical and logical elements needed to build the MORENT, a rental e-commerce website.

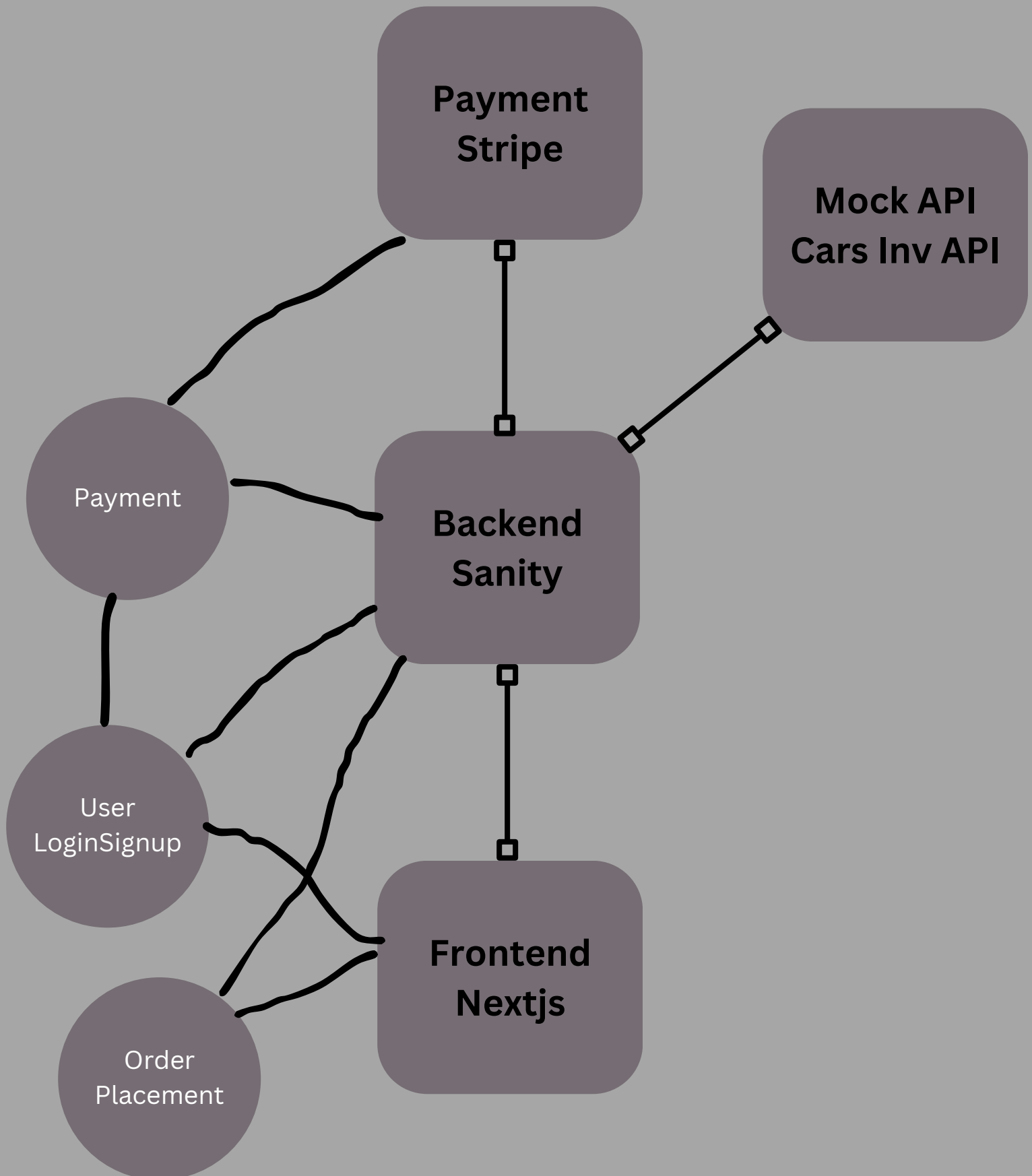
## Tech Stack

- **Frontend:** Using Next.js, Shadcn UI, and Tailwind CSS for responsive, dynamic, and beautiful user interfaces
- **Backend:** Sanity for managing car inventory, offering a headless CMS with easy data structuring and scalability.
- **Payment:** Integrate Stripe for secure, user-friendly, and global payment processing with seamless checkout flows.
- **Authentication:** Clerk for easy, secure authentication with features like social logins, multi-tenancy, and session management.
- **Notifications:** OneSignal for push notifications, ensuring users stay updated about offers, bookings, and reminders.

## Pages

- Home page
- Popular Cars/Recommended Cars/Searched Cars
- Car details page/Buy service Page
- Form Fill/Payment Page
- User Dashboard for analytics

# Technical Architecture Diagram for MORANT Rental Platform



## **User Authentication Flow:**

- Users can sign up or log in via the frontend (Next.js) using Clerk for authentication.
- Clerk manages user sessions and provides secure tokens, which are verified on each request to the backend (Sanity) to ensure authorized actions.

## **Car Inventory Management Flow:**

- The car inventory is stored and managed in the backend (Sanity), which serves as the central database.
- The Mock API Cars Inventory API provides external car details, which are synced with Sanity for up-to-date information.
- The frontend (Next.js) fetches car availability and details in real-time from Sanity to display to users.

## **Order Placement Flow:**

- Once the user selects a car, the frontend (Next.js) sends the booking details to the backend (Sanity).
- The backend processes the booking, updates the inventory, and stores the order information.
- A confirmation is sent back to the frontend, and the order details are displayed to the user.

## **Payment Flow:**

- After booking a car, the user proceeds to payment through the frontend (Next.js).
- Payment requests are securely processed by Stripe, which sends the payment status to the backend (Sanity).
- Upon successful payment, the backend updates the order status and notifies the user.

## **Notifications Flow:**

- Notifications for events like booking confirmations, reminders, or payment updates are triggered by the backend (Sanity).
- These notifications are delivered to the user through OneSignal, which integrates with the frontend (Next.js) for seamless user updates.

# API Endpoints for MORANT Rental Website

- **/products** – Fetch all available products from Sanity.
- **/product/{id}** – Fetch details of a specific product by ID.
- **/latest-products** – Fetch the latest added products.
- **/popular-products** – Fetch the most popular products.
- **/recommended-products** – Fetch recommended products for the user.
- **/orders** – Create a new order in Sanity.
- **/order/{id}** – Fetch details of a specific order by ID.
- **/rental-duration** – Add rental details for a product (duration and deposit).
- **/shipment** – Track order shipment status via third-party API.
- **/auth/signup** – Handle user signup via Clerk.
- **/auth/login** – Handle user login via Clerk.
- **/auth/logout** – Log the user out from Clerk.
- **/payment/initiate** – Initiate payment for an order via Stripe.
- **/payment/status/{id}** – Check payment status for a specific transaction.
- **/notifications** – Trigger notifications using OneSignal.
- **/notifications/{id}** – Fetch notification details by ID.
- **/reviews** – Add or fetch reviews for a product.
- **/reviews/{productId}** – Fetch all reviews for a specific product.

# Possible Sanity Schemas

**name:** 'product'

**title:** 'Product'

**type:** 'document'

**fields:**

- **name:** 'title', title: 'Title', type: 'string'
- **name:** 'description', title: 'Description', type: 'text'
- **name:** 'price', title: 'Price', type: 'number'
- **name:** 'category', title: 'Category', type: 'string'
- **name:** 'image', title: 'Image', type: 'image'
- **name:** 'available', title: 'Available', type: 'boolean'
- **name:** 'rentalDuration', title: 'Rental Duration', type: 'string'

**name:** 'category'

**title:** 'Category'

**type:** 'document'

**fields:**

- name: 'name', title: 'Category Name', type: 'string'
- name: 'slug', title: 'Slug', type: 'slug', options: { source: 'name', maxLength: 96 }

**name:** 'customer'

**title:** 'Customer'

**type:** 'document'

**fields:**

- name: 'firstName', title: 'First Name', type: 'string'
- name: 'lastName', title: 'Last Name', type: 'string'
- name: 'email', title: 'Email', type: 'string'
- name: 'phone', title: 'Phone', type: 'string'
- name: 'address', title: 'Address', type: 'string'

# Possible Sanity Schemas

**name:** 'order'

**title:** 'Order'

**type:** 'document'

**fields:**

- name: 'orderId', title: 'Order ID', type: 'string'
- name: 'product', title: 'Product', type: 'reference', to: [{ type: 'product' }]
- name: 'customer', title: 'Customer', type: 'reference', to: [{ type: 'customer' }]
- name: 'rentalStartDate', title: 'Rental Start Date', type: 'datetime'
- name: 'rentalEndDate', title: 'Rental End Date', type: 'datetime'
- name: 'totalPrice', title: 'Total Price', type: 'number'
- name: 'status', title: 'Order Status', type: 'string'

**name:** 'review'

**title:** 'Review'

**type:** 'document'

**fields:**

- name: 'product', title: 'Product', type: 'reference', to: [{ type: 'product' }]
- name: 'customer', title: 'Customer', type: 'reference', to: [{ type: 'customer' }]
- name: 'rating', title: 'Rating', type: 'number'
- name: 'comment', title: 'Comment', type: 'text'
- name: 'date', title: 'Date', type: 'datetime'