Marketplace Technical Foundation “Morent Car Rental”

1. Technical Plan Aligned with Business Goals:

Frontend Requirements:

- Next.js Framework with Typescript, Tailwind.
- Responsive design for mobile and desktop.
- Pages:

  Home: Showcase popular cars, offers, and categories.
  Car Listings: Display cars with filter options (price, model, etc.).
  Car Details: Detailed view of each car.
  Cart: View selected cars.
  Checkout: Payment and rental details.
  Order Confirmation: Booking details and confirmation.

Backend Requirements using Sanity CMS:

- **Sanity CMS:** To manage car data, user details, and bookings.
- **Schemas:** Car (name, price, seats), User (name, email), Booking (car, user, rental duration, total cost).

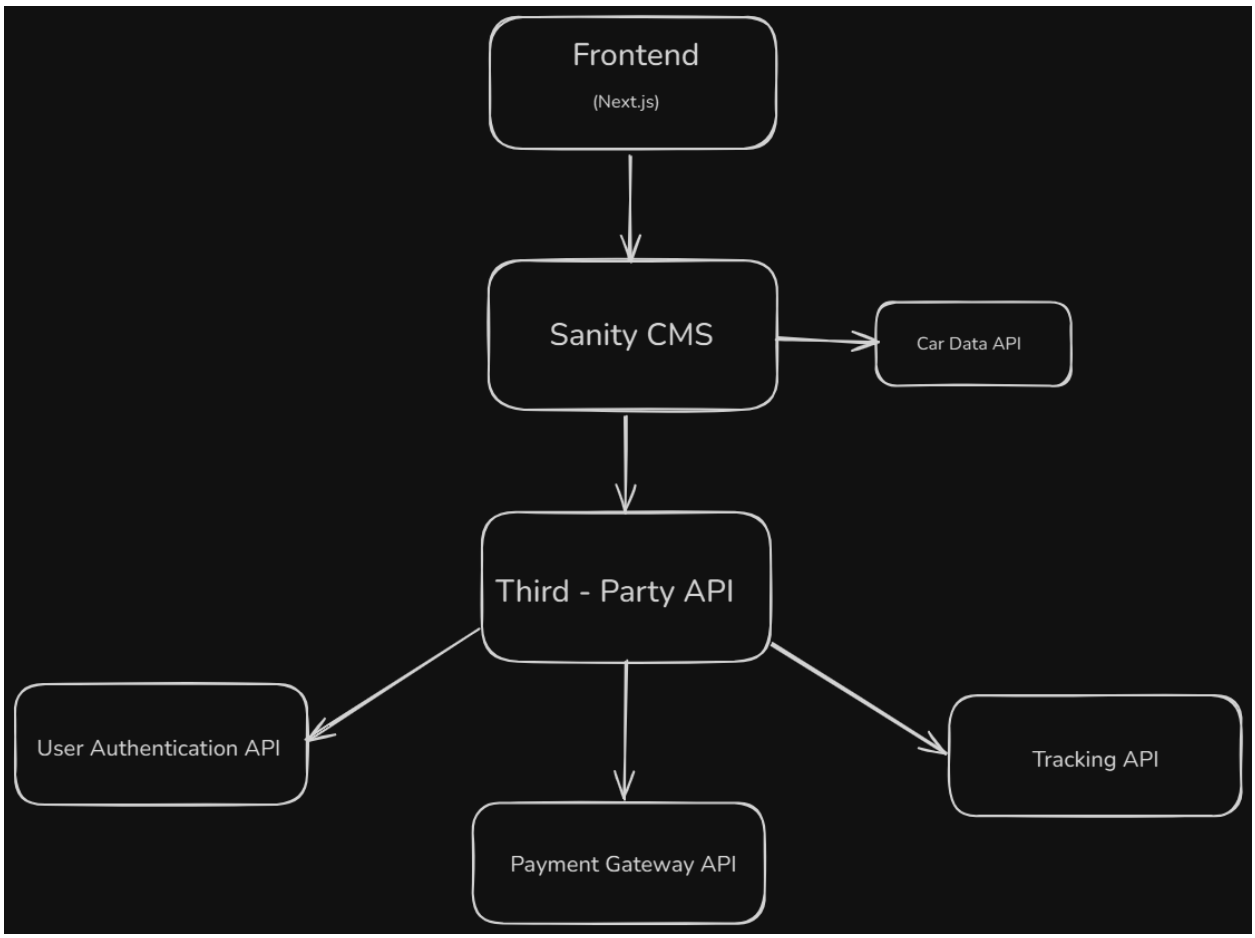
Third-Party APIs:

- **Shipment Tracking API:** For real-time car status.
- **Payment Gateway API:** Payment processing.

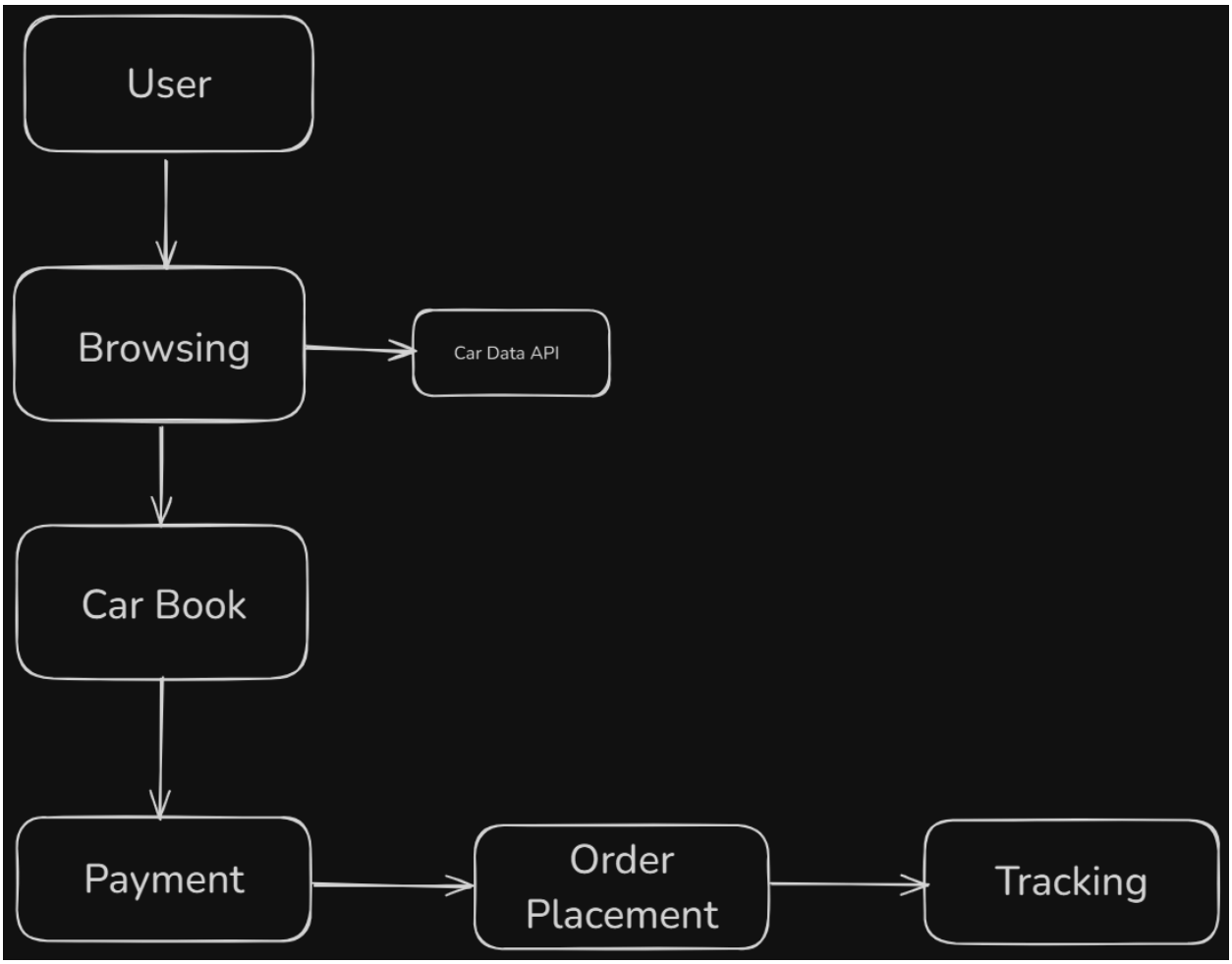
Availability API: Real-time car availability.

2. System Architecture Overview:

Architecture Diagram:



3. Key Workflows:



4. API Endpoints:

Endpoint	Method	Purpose	Response
/cars	GET	Fetch all cars.	{ "id": 1, "name": "Toyota", "price": 80, "seats": 4 }
/bookings	POST	Create new booking.	{ "carId": 1, "userId": 123, "bookingId": 456, "duration": "7 days", "totalPrice": 560 }
/tracking	POST	Tracking cars.	{"carId": 1, "userId": 123, "bookingId": 456}
/signup	POST	Register a new user.	{"message": Registration Successful, "userId": 123}
/payment	POST	Process payment.	{ "bookingId": 456, "paymentStatus": "Success" }

5. Sanity Schema Example:

```
export default {
  name: 'car',
  type: 'document',
  fields: [
    { name: 'name', type: 'string' },
    { name: 'price', type: 'number' },
    { name: 'seats', type: 'number' },
    { name: 'image', type: 'image' }
  ]
};
```