Marketplace Technical Foundation

EliteRide Rentals

1. Frontend Requirements

The frontend will serve as the user-facing interface, ensuring a seamless and user-friendly experience.

User-Friendly Interface

- **Browse Products**: Users should be able to view available cars with filters (e.g., category, price, car type).

- **Search Functionality**:
Include search bar for quick car searches.

Responsive Design

- Fully responsive design for optimal usage on:
  - Mobile devices.
  - Tablets.
  - Desktop screens.
- Ensure touch-friendly interactions for mobile users.

Essential Pages

- **Home Page**:
Highlight car categories, promotions, and call-to-action (CTA) buttons.

- **Car Listing Page**:
List cars with filtering and sorting options.

- **Car Details Page**:
Display detailed information such as car features, availability, and pricing.

- **Checkout Page**:
Collect user information (pickup/drop-off location, rental duration, etc.) and process payment.

- **Order Confirmation Page**:

Display booking details and a unique booking ID.

2. Backend Requirements Using Sanity CMS

Sanity CMS will act as the backend to manage and store data related to cars, customers, and bookings.

2.1 Sanity Schemas

Design the following schemas to align with business goals:

- **Cars Schema**:
  - Fields: `id`, `name`, `category`, `rental_price`, `features`, `condition`,`availability`.
  - Enable sorting and filtering options (e.g., by category, price range).

- **Customer Schema**:
  - Fields: `customer_id`, `name`, `contact_info`, `address`, `rental_history`.
  - Include validation for mandatory fields (e.g., `name`, `contact_info`).

- **Booking Schema**:
  - Fields: `booking_id`, `car_info`, `customer_info`, `rental_period`, `pickup_location`, `status`.
  - Track booking statuses: Confirmed, Ongoing, Completed.

- **Payments Schema**:
  - Fields: `payment_id`, `booking_info`, `amount`, `status`.
  - Track payment statuses: Paid, Pending.

2.2. Data Relationships

- Establish relationships between `Bookings`, `Cars`, `Customers`, and `Payments` for seamless data retrieval.
- Implement dynamic updates to reflect real-time availability.

3. Third-Party API Integrations

Integrate third-party APIs for essential backend services to enhance functionality.

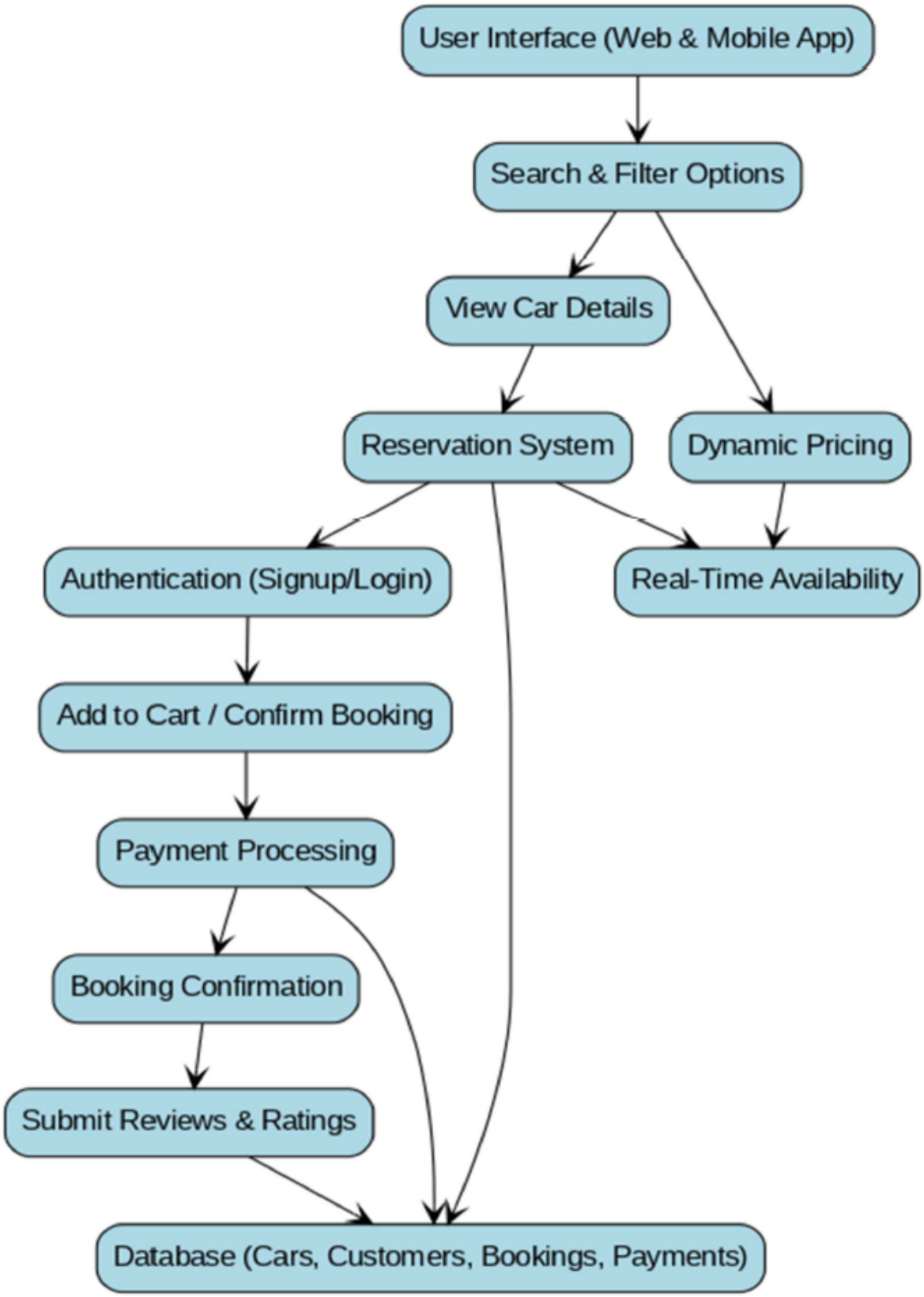
Shipment Tracking API

- Allow users to track car pickups and drop-offs in real time.
- Provide tracking links on the Order Confirmation page.

Payment Gateway API

- Support secure and seamless transactions using:
  - Credit/Debit cards.
  - Bank transfers (if applicable).
- Enable payment status updates for booking records in real time.

System Architecture Diagram



API Endpoints for EliteRide Rentals Platform

Endpoint	Method	Purpose	Response Example
/cars	GET	Fetch car details.	*{ "id": 1, "name": "Toyota Corolla", "category": "Economy", "price": {"hourly": 10, "daily": 50, "weekly": 300} }*
/cars/availability	GET	Check real-time car availability.	*{ "id": 1, "available": true, "location": "Downtown Branch" }*
/bookings	POST	Create a new booking in Sanity CMS.	*{ "bookingId": 123, "status": "Confirmed", "pickupLocation": "Downtown", "rentalPeriod": {"start": "2025-01-20", "end": "2025-01-22"} }*
/bookings/status	GET	Fetch booking status.	*{ "bookingId": 123, "status": "Ongoing", "returnLocation": "Airport" }*
/payments	POST	Process payment for a booking.	*{ "paymentId": 456, "status": "Paid", "amount": 150 }*
/shipment	GET	Fetch shipment status for vehicle delivery.	*{ "shipmentId": 789, "status": "Dispatched", "ETA": "3 hours" }*
/rental-duration	POST	Add rental duration details for a car.	*{ "confirmationId": 321, "status": "Success", "duration": {"hours": 48} }*

Sanity Schema Examples for EliteRide Rentals

Car Schema

```
{
 name: 'car',
 type: 'document',
 fields: [
  { name: 'name', type: 'string', title: 'Car Name' },
  { name: 'category', type: 'string', title: 'Category' },
  { name: 'priceHourly', type: 'number', title: 'Hourly Price' },
  { name: 'priceDaily', type: 'number', title: 'Daily Price' },
  { name: 'priceWeekly', type: 'number', title: 'Weekly Price' },
  { name: 'availability', type: 'boolean', title: 'Availability' },
  { name: 'location', type: 'string', title: 'Location' },
 ],
};
```

Booking Schema

```
{
 name: 'booking',
 type: 'document',
 fields: [
  { name: 'bookingId', type: 'string', title: 'Booking ID' },
  { name: 'car', type: 'reference', to: [{ type: 'car' }], title: 'Car Reference' },
  { name: 'startDate', type: 'datetime', title: 'Start Date' },
  { name: 'endDate', type: 'datetime', title: 'End Date' },
  { name: 'pickupLocation', type: 'string', title: 'Pickup Location' },
  { name: 'returnLocation', type: 'string', title: 'Return Location' },
  { name: 'status', type: 'string', title: 'Status' },
 ],
};
```

Payment Schema

```
{
 name: 'payment',
 type: 'document',
 fields: [
  { name: 'paymentId', type: 'string', title: 'Payment ID' },
  { name: 'amount', type: 'number', title: 'Amount' },
  { name: 'status', type: 'string', title: 'Payment Status' },
  { name: 'booking', type: 'reference', to: [{ type: 'booking' }], title:
'Booking Reference' },
 ],
};
```

Rental Duration Schema

```
{
 name: 'rentalDuration',
 type: 'document',
 fields: [
  { name: 'car', type: 'reference', to: [{ type: 'car' }], title: 'Car Reference' },
  { name: 'hours', type: 'number', title: 'Duration (Hours)' },
  { name: 'days', type: 'number', title: 'Duration (Days)' },
  { name: 'confirmationId', type: 'string', title: 'Confirmation ID' },
  { name: 'status', type: 'string', title: 'Status' },
 ],
};
```