"MARKETPLACE TECHNICAL FOUNDATION

CAR RENTAL E-COMMERCE"

HUZAIFA AYUB

GOVERVER SINDH IT INITIATIVE GOVERNOR HOUSE KARACHI

1. BUSINESS GOALS OVERVIEW

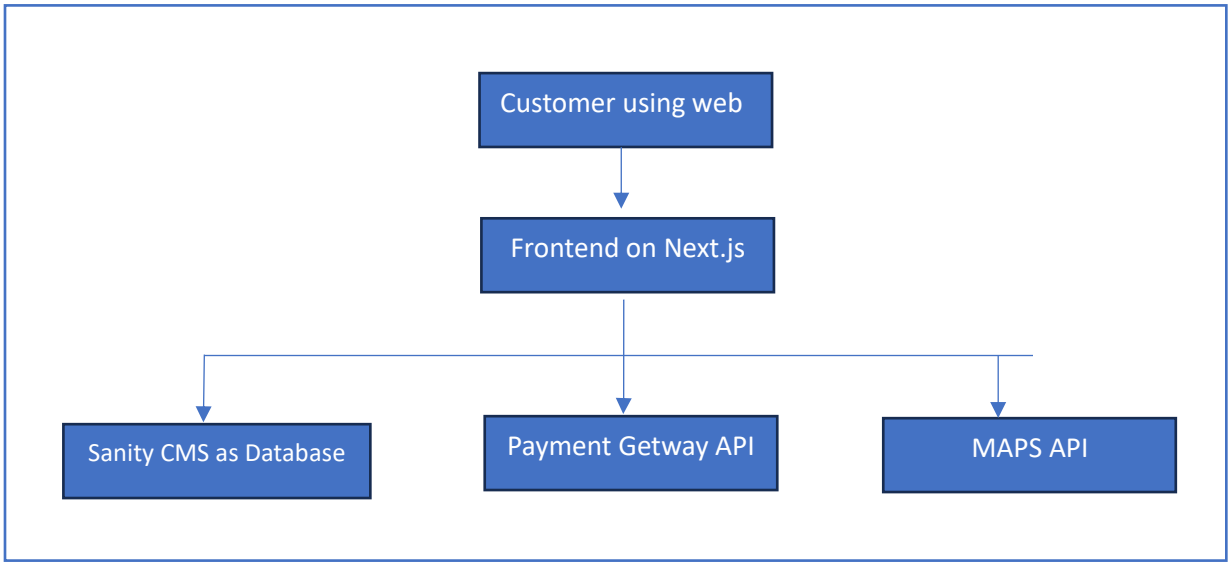
- Provide an easy-to-use platform for renting cars
- Target audience: Tourists, business travelers, and locals needing temporary transportation
- Offer various car types and rental durations
- Set apart by ease of use, flexibility, and competitive pricing

2. TECHNICAL REQUIREMENTS

- **Frontend Requirements:**
    - User-friendly interface for browsing and renting cars
    - Responsive design for both mobile and desktop users
    - Essential pages: Home, Car Listings, Car Details, Booking, User Profile
- **Backend (Sanity CMS):**
    - Manage car data, customer details, and rental records
    - Design schemas in Sanity CMS for:
        - Cars
        - Customers
        - Rentals
        - Locations
- **Third-Party APIs:**
    - Payment gateway (e.g., Stripe) for processing rental payments
    - Maps API (e.g., Google Maps) for showing car pickup locations

3. SYSTEM ARCHITECTURE

Here's a high-level diagram of how the system components interact:



4. MAIN WORKFLOW

- **Car Rental Process:**
    - User browses available cars on the website
    - User selects a car and rental dates
    - Frontend checks car availability with Sanity CMS
    - User provides personal and payment information
    - Payment is processed through the payment gateway
    - Rental record is created in Sanity CMS
    - User receives booking confirmation

- **Car Return Process:**
    - User returns the car to the designated location
    - Staff updates the rental status in Sanity CMS
    - System calculates final cost (including any extra charges)
    - User receives final receipt

5. API REQUIREMENTS

Based on our data schema, here are the key API endpoints we'll need:

- **Get Available Cars**
    - Endpoint: `/api/cars/available`
    - Method: GET
    - Description: Fetch all available cars for rent
    - Response Example:

```
{
 "cars": [
  {
     "car123",
   "model": "Toyota Corolla",
   "type": "Sedan",
   "pricePerDay": 50,
   "location": "Airport"
  },
  {
   "id": "car456",
   "model": "Ford Explorer",
   "type": "SUV",
   "pricePerDay": 75,
   "location": "Downtown"
  }
 ]
}
```

- **Create New Rental**
  - Endpoint: `/api/rentals`
  - Method: POST
  - Description: Create a new car rental
  - Payload Example:

```
{
  "customerId": "cust789",
  "carId": "car123",
  "startDate": "2023-06-01",
  "endDate": "2023-06-05"
}
```

  - Response Example:

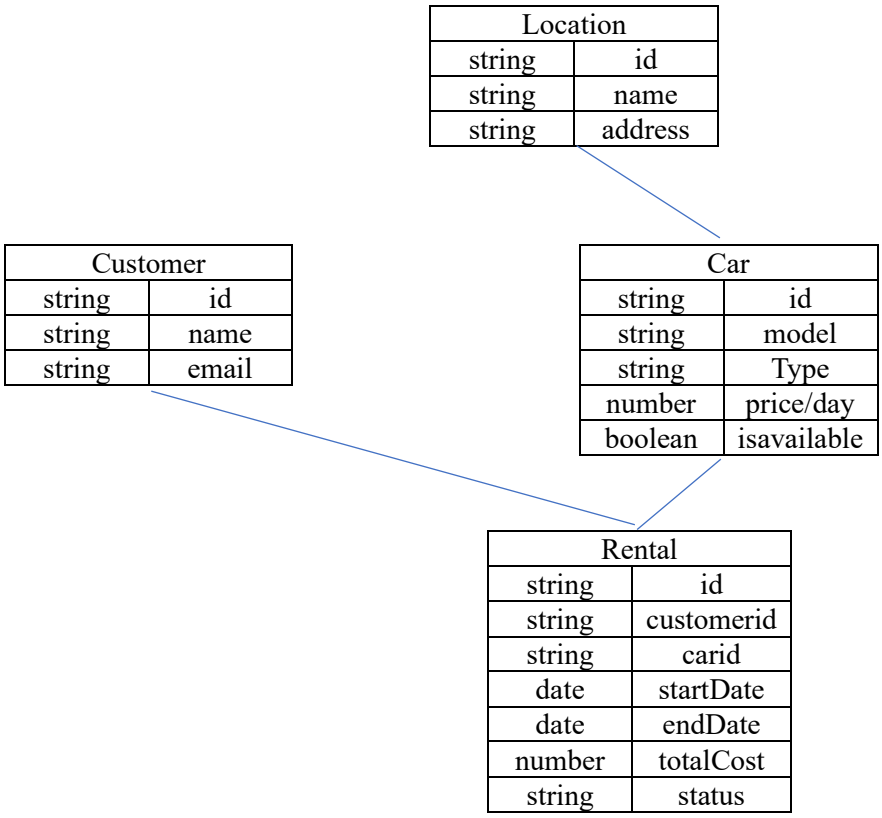
```
{
  "rentalId": "rent001",
  "status": "confirmed",
  "totalCost": 200
}
```

- **Get Rental Details**
  - Endpoint: `/api/rentals/{rentalId}`
  - Method: GET
  - Description: Get details of a specific rental
  - Response Example:

```
{
  "rentalId": "rent001",
  "car": {
    "model": "Toyota Corolla",
    "type": "Sedan"
  },
  "startDate": "2023-06-01",
  "endDate": "2023-06-05",
  "status": "active",
  "totalCost": 200
}
```

6. DATA SCHEMA DESIGN

Here's how we'll organize our data in Sanity CMS:



7. TECHNICAL STACK

- Frontend: Next.js (React framework)
- Backend & Database: Sanity CMS
- Payment Processing: Stripe
- Maps: Google Maps API
- Hosting: Vercel

8. TECHNICAL ROADMAP

- Set up Next.js project and Sanity CMS
- Design and implement frontend pages
- Create Sanity schemas and API endpoints
- Integrate payment gateway and maps API
- Implement user authentication
- Develop car browsing and rental booking functionality
- Create admin panel for managing cars and rentals
- Perform thorough testing
- Deploy to production