Name: Tanzeela Arashad

Roll No: 00431067

Student at GIAIC

Lead Developer, Morent Application

Email: tanzeelaarshad320@gmail.com

Morent Rental E-commerce Platform: Technical Documentation

1. Introduction

Morent is a platform that allows its Users to Rent Cars from Economy to Luxury to SUV. It bridges the gap between mobility and accessibility, eliminating the burden of cost of expensive cars and it helps you stay free from the stress of maintenance of cars.

1. Project Overview

- **Frontend**: Next.js (React-based framework for server-side rendering, static site generation, and API routes).
- **Backend**: Sanity CMS (headless CMS for managing content like car details, pricing, and user data).
- **Third-Party APIs**: For payment processing, geolocation, car availability, and other integrations.
- **Database**: Sanity CMS will handle structured content.

2. Technical Requirements

Frontend (Next.js)

1. **Pages and Routing**:
   - Homepage: Showcase featured cars, search bar, and promotions.
   - Car Listing Page: Filterable and sortable list of cars.
   - Car Detail Page: Detailed information about a specific car.
   - Booking Page: Form for user details, pickup/drop-off dates, and payment.
   - User Dashboard: Manage bookings, profile, and payment history.

   - Admin Dashboard: Manage cars listing, bookings, and content.
2. **Components**:
   - Reusable UI components (e.g., car cards, search bar, date picker).
   - Responsive design for mobile, tablet, and desktop.
3. **State Management**:
   - Use React Context (e.g., user authentication, booking details).
4. **API Integration**:
   - Fetch data from Sanity CMS (car details, pricing, etc.).
   - Integrate third-party APIs for payments (e.g., Stripe), maps (e.g., Google Maps), and car availability.
5. **SEO and Performance**:
   - Use Next.js features like getStaticProps and getServerSideProps for optimized performance.
   - Implement Meta tags and structured data for SEO.
6. **Authentication**:
   - Use NextAuth.js or Firebase for user authentication (sign-up, login, and password reset).
7. **Payment Integration**:
   - Integrate a payment gateway like Stripe or PayPal for secure transactions.
8. **Search and Filters**:
   - Implement a robust search and filter system for car listings (e.g., by price, location, car type).

Backend (Sanity CMS)

1. **Content Modeling**:
   - Define schemas for car details (e.g., make, model, year, price, image, availability).
   - Create schemas for user reviews, FAQs, and blog posts (if needed).
2. **API Endpoints**:
   - Use Sanity's GROQ (Graph-Relational Object Queries) to fetch data.
   - Create custom API routes in Next.js for server-side logic (e.g., booking confirmation).
3. **Media Management**:
   - Use Sanity's built-in media library for storing car images and videos.

4. **Webhooks**: Set up webhooks to trigger actions (e.g., send confirmation emails when a booking is made)

Third-Party APIs

1. **Payment Gateway**:
   - Stripe, PayPal for handling payments.
2. **Geolocation and Maps**:
   - Google Maps API for location-based services (e.g., pickup/drop-off points).
3. **Email and Notifications**:
   - Nodemailer for sending booking confirmations and notifications.
4. **Car Availability and Pricing**:
   - Integrate with a third-party API (if applicable) to fetch real-time car availability and pricing.
5. **Analytics**:
   - Google Analytics for tracking user behavior.

Diagram Rental E-Commerce Technical Documentation

```
+-------------------+     +-------------------+     +-------------------+

|                   |     |                   |     |                   |

|  Frontend         |     |  Backend          |     |  Third-Party      |

|  (Next.js)        |     |  (Sanity CMS)     |     |  APIs             |

|                   |     |                   |     |                   |

| - Homepage        |     | - Car Data        |     | - Payment Gateway |

| - Car Listings    |<----->| - Pricing       |<----->|   (Stripe, PayPal)|

| - Car Details     |     | - Availability    |     |                   |

| - Booking Form    |     | - User Reviews    |     | - Geolocation     |

| - User Dashboard  |     |                   |     |   (Google Maps)   |

|                   |     |                   |     |                   |

|                   |     |                   |     | - Email Service   |
|                   |      |                   |      |   (SendGrid)      |
|                   |      |                   |      |                   |
+-------------------+      +-------------------+      +-------------------+
        ^                          ^                          ^
        |                          |                          |
        |                          |                          |
        v                          v                          v
+-------------------+      +-------------------+      +-------------------+
|                   |      |                   |      |                   |
|   User Browser    |      |   Database        |      |   External        |
|                   |      |   (Optional)      |      |   Services        |
| - Renders UI      |      | - User Accounts   |      | - Maps, Payments, |
| - Handles Input   |      | - Booking History |      |   Emails, etc.    |
|                   |      |                   |      |                   |
+-------------------+      +-------------------+      +-------------------+
```