

Market Place Hackathon Rental Car ECommerce Website

**DRIVE YOUR DREAMS
RENT YOUR RIDE!**

Our Goal:

our main business objective is to built karachis best online car rental market place to give our customers an affordable ,fast and reliable means of hiring cars. It is our goal to provide a seamless rental experience by offering personalized and customizable services while keeping the benefits of affordability and convenience at the forefront. By donning so ,we aim to improve urban mobility, increase customer happiness ,and create a lasting bond with our consumers, allowing us to become the rents of choice for everyone in karachi.

objective:

The aim of this plan is to build a feature-rich, scalable, and user-friendly car rental platform tailored for a seamless rental experience. The core aspects of this project include:

- 1. Sanity CMS:** To manage car listings and rental orders efficiently.
- 2. Next Auth:** Log in and log Out
- 3. Custom APIs:** To handle car rentals, user orders, and order tracking.
- 4. Stripe:** For reliable and secure payment processing.
- 5. Upstash :**Manage data
- 6.Next.js:** Lightning-fast performance with server-side rendering and static site generation.
- 7.Tailwind CSS:** Clean, responsive, and modern design.

This platform is designed to provide both administrators and end-users with an intuitive, reliable system that aligns with modern web development standards.

System Architecture:

Workflow Overview:

The system will operate in the following sequence

- 1. User Authentication:** Secure login and signup powered by Next Auth
- 2. Car Browsing:** Real-time data fetched from Sanity CMS ensures up-to-date car availability.
- 3. Rental Process:** Users can select a car, specify rental durations, and proceed to checkout.

4. Payment Processing: Transactions are handled securely through Stripe, ensuring data protection. 5. **Order Tracking:** Rental order statuses are updated dynamically via custom APIs.

Architecture Diagram:

[User] --> search Car --> Filter Cars --> Choose Car--> Add to Cart -->|Signin| Auth [User] -->|Browses Cars| Frontend (Next.js) [Frontend] -->|Fetches Cars| Sanity CMS [Frontend] -->|Processes Payment| Stripe API [Frontend] -->|Tracks Orders| Custom APIs

Features & Workflows

Frontend Features:

1. Authentication: Use Clerk to handle user login/signup.

- Session management is handled by Next Auth without involving Sanity CMS.

2. Car Browsing:

- Fetch and display car data from Sanity CMS using GROQ queries. o Implement filters (e.g., price range, car type) and sorting functionality.

3. Checkout Process:

- Collect user rental details and payment information via Stripe-hosted checkout. o Display an order confirmation and receipt after successful payment.

4. Order Tracking:

- Retrieve rental duration and order status via custom APIs.

Backend Features:

1. Sanity CMS: Manage car listings and orders using Sanity Studio.

- Customize schemas for car details and order tracking.

2. Custom APIs:

- `/product/cars`: Fetch available cars.
- `/order/rent`: Create and save rental orders.
- `/product/track-order`: Retrieve the status of rental orders.

3. Admin Panel:

- Enable the admin to add, update, or delete cars via Sanity Studio

End point	Method	Description
/product/cars	GET	Fetch all available Cars
/order/rent	POST	Create a rental order and save details.
/product/track-orde	GET	Retrieve the status of a rental order by its ID
/payment/checkou	POST	Process payment using Stripe

Sanity CMS Cars Schema

```
1  import { defineType, defineField, defineArrayMember } from "sanity";
2
3  export const carRecommendation = defineType({
4    name: "carRecommendation",
5    type: "document",
6    title: "Car Recommendation",
7    fields: [
8      defineField({
9        name: "cars",
10       type: "array",
11       title: "Cars",
12       of: [
13         defineArrayMember({
14           type: "object",
15           fields: [
16             defineField({
17               name: "name",
18               type: "string",
19               title: "Name",
20               description: "Name of the car",
21             }

```

```

22     defineField({
23         name: "type",
24         type: "string",
25         title: "Type",
26         description: "Type of the car (e.g., SUV, Hatchback)",
27     }),
28     defineField({
29         name: "price",
30         type: "number",
31         title: "Price",
32         description: "Rental price per day",
33     }),
34     defineField({
35         name: "originalPrice",
36         type: "number",
37         title: "Original Price",
38         description: "Original price per day (if applicable)",
39         validation: (Rule) => Rule.required(),
40     }),
41     defineField({
42         name: "fuel",
43         type: "string",
44         title: "Fuel Capacity",
45         description: "Fuel capacity of the car (e.g., 70L)",
46     }),

```

Deliverables:

1. System Architecture Diagram:

A detailed visual representation of how components interact, including user authentication, car data fetching, payment processing, and order tracking.

2. Sanity Schemas: Custom schemas for cars and rental orders, designed for easy management and scalability.

3. API Endpoints: Fully functional endpoints for retrieving cars, creating orders, tracking rentals, and processing payments.

4. Frontend Pages:

- Home: Highlights featured cars and promotional offers.
- Car Listings: Displays all available cars with filter options.
- Car Details: Provides detailed information for each car.
- Checkout: Allows users to complete the rental process.
- Order Tracking: Enables users to monitor the status of their rentals