**Marketplace Builder Hackathon 2025 (Day-2)**

**Customized Car Rental Marketplace Plan**

**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. **Clerk**: For secure and easy user authentication.

3. **Custom APIs**: To handle car rentals, user orders, and order tracking.

4. **Stripe**: For reliable and secure payment processing.

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 Clerk
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] -->|Signin| Clerk**

**[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 Clerk without involving Sanity CMS.**
2. **Car Browsing:**
   * **Fetch and display car data from Sanity CMS using GROQ queries.**
   * **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.**
   * **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.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Endpoint** | |  | | --- | |  |   **Method** | **Description** |
| **/product/cars** | **GET** | **Fetch all available cars.** |
| **/order/rent** | **POST** | **Create a rental order and save details.** |
| **/product/track-order** | **GET** | **Retrieve the status of a rental order by its ID.** |
| **/payment/checkout** | **POST** | **Process payment using Stripe.** |

**Sanity CMS Schemas**

**Car Schema:**

****

**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.**