Marketplace Technical Foundation

Author:
Ali Shahid

Roll no:
00230027

Day 2 of marketplace builder hackathon

Marketplace Technical Foundation - **AliCarGo**

Day 2 Goal

The aim of Day 2 is to transform the business planning into a technical plan for AliCarGo, a car rental marketplace. The focus is to define technical requirements, create a system architecture, plan APIs, and draft essential documentation.

1. Technical Requirements

Frontend

- Framework: **Next.js** for building a dynamic and responsive user interface.
- Features to Include:
  - **Homepage**: Overview of available cars and offers.
  - **Product Listing**: Display detailed car information, availability, and filters (e.g., type, location).
  - **Cart**: Users can select and view cars they intend to rent.
  - **Checkout**: A seamless rental process with payment integration.
  - **User Authentication**: Login and registration functionality for renters.

Sanity CMS

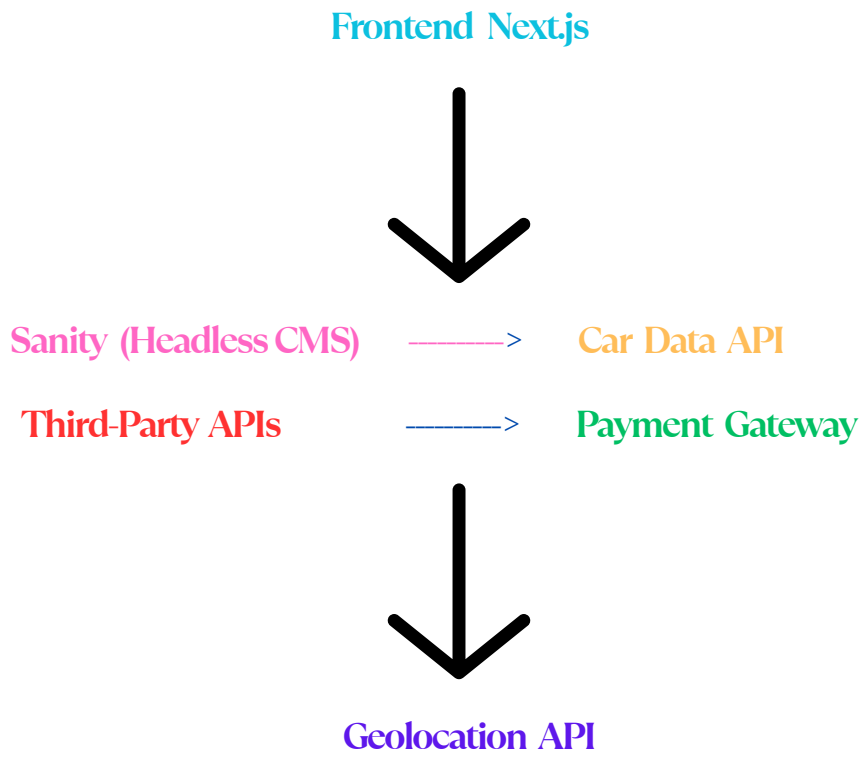
- **Purpose**: To manage product (car) data and orders efficiently.
- **Features**:
  - Manage car inventory, including name, price, images, and availability.
  - Monitor customer bookings and order history.

Third-Party APIs

- Payment Gateway: Integrate Stripe or PayPal for secure transactions.
- Shipment/Tracking: Not applicable, but could include email or SMS notifications for booking confirmation.
- Maps Integration: Google Maps API to display car pickup/drop-off locations.

2. System Architecture Plan

Architecture Diagram:



Flow Description

1. **Frontend**: Users interact with the website to browse, filter, and book cars.
2. **Sanity CMS**: Backend CMS to manage car inventory and bookings.
3. **APIs**:
   - **Product Data API**: Fetch car details and availability from Sanity CMS.
   - **Payment Gateway API**: Handle payment processing.
   - **Google Maps API**: Display location-based car rental services.

3. API Requirements Plan

Example API Structure

Product Endpoint

- **Endpoint Name**: /api/products
- **Method**: GET
- **Purpose**: Fetch all available cars.
- **Response**:

[{

"id": "car123",

"name": "Toyota Corolla",

"price": 50,

"availability": true,

"location": "Lahore",

"image": "car-image-url"

}]

Booking Endpoint

- **Endpoint Name**: /api/bookings
- **Method**: POST
- **Purpose**: Book a car for rental.
- **Request Body**:

{

"userId": "user123",

"carId": "car123",

"startDate": "2025-01-20",

"endDate": "2025-01-25"

}

API Schema for Sanity

```
export default

{

name: "product",

type: "document",

title: "Product",
```

```
fields: [
{
name: "name",
type: "string",
title: "Car Name",
},
{
name: "price",
type: "number",
title: "Rental Price per Day",
},
{
name: "availability",
type: "boolean",
title: "Available",
},
{
name: "location",
type: "string",
title: "Rental Location",
}
,
{
name: "image",
type: "image",
title: "Car Image",
```

},

],

}

Conclusion

The technical foundation of AliCarGo establishes a strong and structured plan for building an efficient and user-friendly car rental marketplace. By leveraging Next.js for a dynamic frontend, Sanity CMS for backend management, and third-party APIs for enhanced functionality like payments and location services, the platform ensures seamless user interactions and streamlined backend operations.

The system architecture provides a clear roadmap of how different components interact, ensuring scalability and maintainability. The detailed API requirements and Sanity schema draft lay the groundwork for efficient data management and integration. Additionally, collaborative feedback from peers and mentors ensures the plan is robust and aligns with best practices.

With this foundation in place, AliCarGo is ready to progress to the development phase, transforming this vision into a fully operational car rental marketplace.