Name :Shazia Bashir

Roll # : 00379887

Hackathon Day 2: Planning the Technical Foundation

Customized Car Rental Marketplace Plan :

System Architecture for Rental Car Service Platform

Frontend

- **Framework:** Next.js
- **Styling:** Tailwind CSS
- **Component Structure:**
    - Hero Section (Pixel-perfect design)
    - Sidebar (Vehicle type, capacity, price filter)
    - Pick and Drop Form (Pickup and Drop-off date/time selection)
    - About Page (Full-page layout with icons)
    - Slider Components (Image sliders where required)

Backend :

- **CMS:** Sanity CMS (for dynamic content management , for storing vehicle data, user data, and booking details)

- **APIs:**
    - Custom REST API for booking and user management (Node.js/Express.js)
    - 3rd Party APIs (Payment gateways, Google Maps API for location services)

Vehicle Management :

- **Tracking System :**

    - Live Vehicle Tracking (Integration with GPS APIs)
    - Real-time Vehicle Handover Process (QR Code or NFC-based verified

System Flow

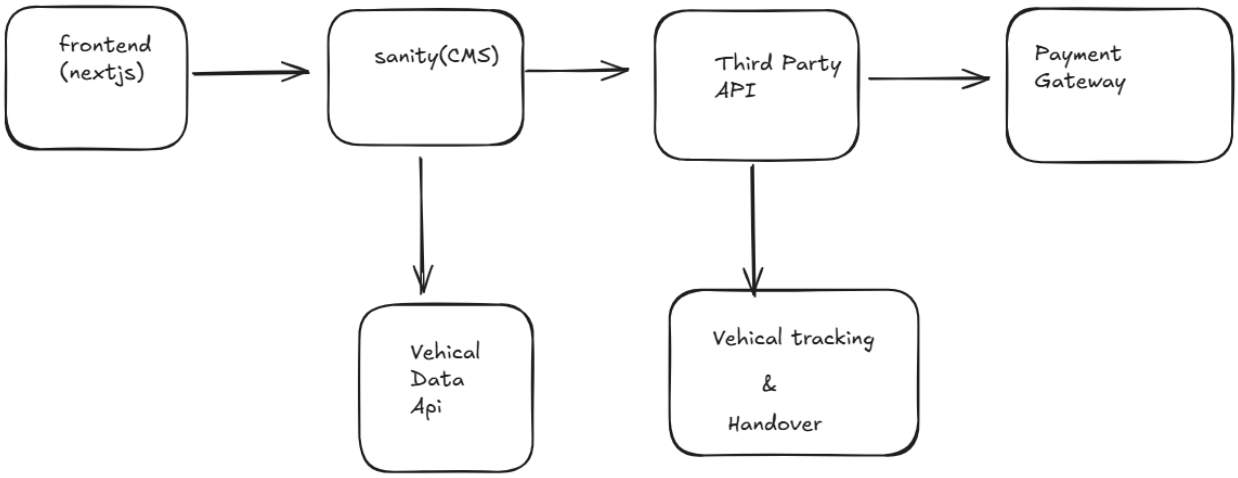
1. **User Interaction**
    a. Users select vehicle type, capacity, and price range using the Sidebar.
    b. Users choose pickup and drop-off details through the Pick and DropForm.

2. **Data Handling**
    a. Vehicle data is fetched from Sanity CMS .
    b. Booking details are processed via custom APIs.

3. **Tracking & Handover**

    a. Vehicles are tracked using GPS APIs.
    b. Vehicle handover is managed through secure verification systems.

Architecture Diagram



Key Workflows for Rental Car Service Platform

1. **User Registration & Authentication:**
    a. User signs up or logs in.
    b. User data is stored in the database (e.g., Firebase/Auth service).
    c. Confirmation email or SMS is sent to the user.

2. **Vehicle Browsing & Filtering:**
    a. User navigates the categories via the Sidebar (Type, Capacity, Price).
    b. Next.js fetches vehicle data from Sanity CMS.
    c. Filtered vehicle listings are displayed dynamically on the frontend.

3. **Pick-Up & Drop-Off Scheduling:**
    a. User selects pick-up/drop-off locations, dates, and times using the Pick and Drop Form.
    b. Selections are stored in state and sent to the backend when confirmed.

4. **Vehicle Booking:**
    a. User selects a vehicle and clicks on the 'Book Now' button.
    b. Booking details (vehicle, user, pick-up/drop-off info) are sent to the backend.
    c. Booking confirmation is stored in the database and displayed to the user.

5. **Payment Processing:**
    a. User proceeds to payment after booking.

b. Payment request is sent to a 3rd-party payment gateway API (e.g., Stripe).
c. On successful payment, booking status is updated in the database.

6. **Vehicle Handover & Live Tracking:**
    a. Vehicle delivery status is updated in the database.
    b. Live vehicle tracking is handled via a 3rd-party GPS API and displayed to the user.

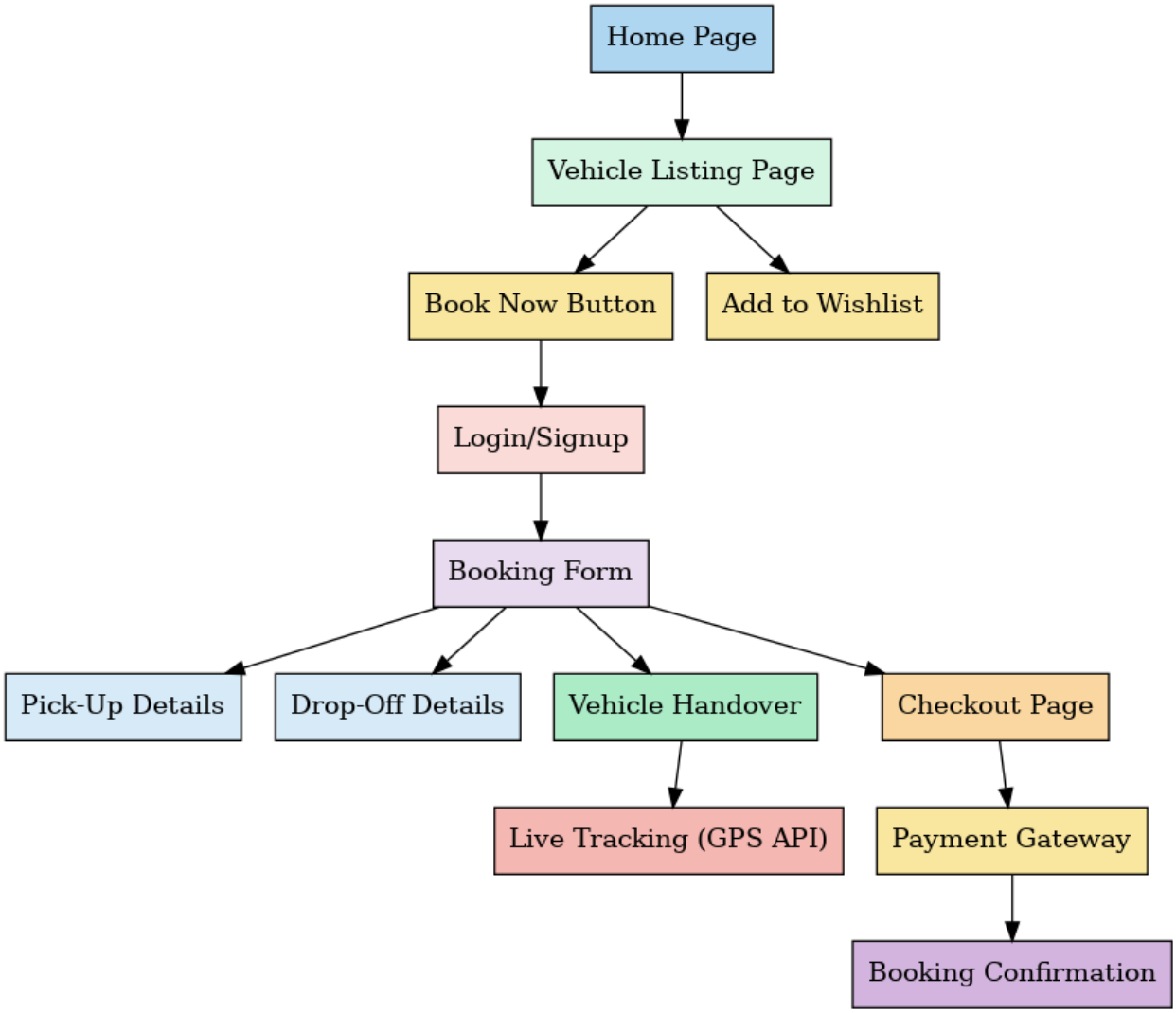
7. **Booking Management:**
    a. User views current and past bookings from the dashboard.
    b. Booking data is fetched from the database and rendered in the frontend.

8. **Admin Management:**
    a. Admin uploads or updates vehicle data via Sanity CMS.
    b. Admin manages bookings and user data from a secure dashboard.

9. **Notifications:**
    a. Email/SMS notifications are sent for booking confirmation, reminders, and updates using a 3rd-party service (e.g., Twilio, SendGrid).

10. **User Feedback & Reviews:**
    a. Users submit feedback after completing a booking.
    b. Reviews are stored in the database and displayed on vehicle pages.

workflow

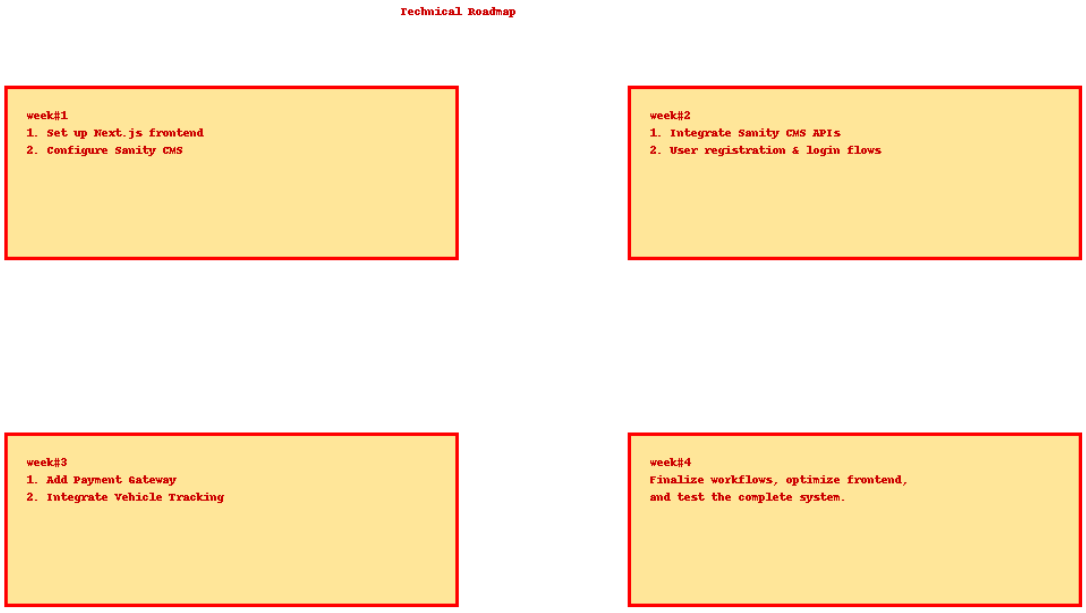


Step_3_API Requirements for Rental Car Service Platform

#NAME?	Method	Description	Payload	Response Example

/register-user	POST	Registers a new user in the system.	{"username": "JohnDoe", "email": "john.doe@example.com", "password": "password123"}	{"userId": 123, "status": "Success"}
/get-products	GET	Fetches a list of available products or services.	N/A	{"products": [{"id": 1, "name": "Sedan", "price": 50}]}
/create-order	POST	Creates a new order for selected items.	{"userId": 123, "productId": 456, "quantity": 2}	{"orderId": 567, "status": "Confirmed"}
/cancel-order	DELETE	Cancels an existing order.	{"orderId": 567}	{"status": "Cancelled"}
/update-condition	PUT	Updates the condition of a product or service after use.	{"orderId": 567, "condition": "Good"}	{"status": "Updated"}
/order-history	GET	Retrieves the history of all past orders for a user.	N/A	{"orders": [{"orderId": 567, "product": "Sedan", "date": "2024-12-20"}]}
/payment	POST	Processes payment for an order.	{"orderId": 567, "paymentMethod": "Credit Card", "amount": 100}	{"paymentId": 789, "status": "Success"}
/track-shipment	GET	Tracks the status of a shipment or return.	{"orderId": 567}	{"status": "In Transit", "expectedDelivery": "2024-12-25"}

Technical Road Map



Sanity CMS Schema

Car Schema :

```
// schemas/car.js
export default {
  name: 'car',
```

```
type: 'document',
title: 'Car',
fields: [
 {
  name: 'name',
  type: 'string',
  title: 'Car Name',
 },
 {
  name: 'type',
  type: 'string',
  title: 'Car Type',
  options: {
   list: ['Sedan', 'SUV', 'Hatchback', 'Convertible', 'Sport'],
  },
 },
 {
  name: 'pricePerDay',
  type: 'number',
  title: 'Price Per Day ($)',
 },
```

```
{
  name: 'availability',
  type: 'boolean',
  title: 'Available for Rent',
  initialValue: true,
},
{
  name: 'fuelCapacity',
  type: 'string',
  title: 'Fuel Capacity (Liters)',
  description: 'Example: 90L',
},
{
  name: 'transmission',
  type: 'string',
```

xxxxxxxxxxxxxxxxxx