Planning the Technical Foundation for a Car Rental Website

1. Frontend Requirements

Objective

Design a responsive and user-friendly interface for the car rental website, ensuring accessibility and seamless navigation across devices.

Key Pages and Features

1. Home Page

- **Eye-catching hero section with featured offers.**
- **Search functionality for cars (by location, date, car type, etc.).**
- **Testimonials and trust signals (e.g., secure booking, customer reviews).**

2. Product Listing Page

- **Display available cars based on user's search criteria.**
- **Filters (price range, car type, brand, etc.) and sorting options.**
- **Pagination or infinite scroll for a smooth browsing experience.**

3. Product Details Page

- **Detailed information about the selected car, including images, pricing, specifications, and rental terms.**
- **Call-to-action (CTA) for booking or adding the car to the cart.**

4. Cart Page

- **Overview of selected cars with pricing breakdown.**
- **Option to modify selections or proceed to checkout.**

5. Checkout Page

- User registration or guest checkout options.

- Form for entering personal details, rental period, and payment information.

6. Order Confirmation Page

- Display order summary with rental details and payment confirmation.

- Options to download/print the receipt and view/manage orders.

Technology Stack

- Framework/Library: React or Vue.js for dynamic components.

- Styling: Tailwind CSS or Bootstrap for rapid, responsive design.

- State Management: Redux or Vuex for handling application state.

- Testing: Jest and React Testing Library for unit and integration tests.

---

2. Backend Requirements with Sanity CMS

Objective

Configure Sanity CMS to manage dynamic content, including car inventory, customer details, and order records.

Key Schemas

1. Car Schema

- Fields: Name, Description, Images, Pricing, Availability, Features, Location.

- Relations: Link to categories (e.g., SUV, Sedan, Hatchback).

2. Customer Schema

- Fields: Name, Email, Phone Number, Address, Rental History.

3. Order Schema

- Fields: Order ID, Customer Reference, Cars Rented, Total Cost, Rental Period, Payment Status.

- Relations: Link to customer and car data.

4. Category Schema (Optional)

- Fields: Name, Description, Parent Category.

Technology Stack

- CMS: Sanity for content and data management.

- Backend Framework: Node.js with Express for custom APIs if required.

- Database: MongoDB (used alongside Sanity's built-in GROQ queries).

- Hosting: Sanity Studio deployed on Vercel or Netlify.

---

3. Third-Party API Integration

Objective

Incorporate essential external services for payment processing, shipment tracking, and other backend functionalities.

Planned Integrations

1. Payment Gateway

- Providers: Stripe, PayPal, or Razorpay.

- Features: Secure transactions, support for multiple currencies, and recurring payments for long-term rentals.

2. Shipment Tracking API (if applicable)

- Providers: EasyPost or Shippo.

- Use Case: Track delivery of physical rental agreements or accessories.

3. Authentication Services

- Providers: Firebase Authentication or Auth0.

- Features: Email/password login, social sign-in, and password recovery.

4. Email Notifications

- Providers: SendGrid or Mailgun.

- Use Case: Send booking confirmations, reminders, and promotional offers.

5. Analytics and Monitoring

- Tools: Google Analytics for website metrics and Sentry for error tracking.

Summary

The proposed plan ensures a robust and scalable foundation for the car rental website. The responsive frontend paired with Sanity CMS and strategic API integrations will deliver a seamless experience for users while simplifying backend management for administrators.