DAY : 02

PLANNING THE TECHNICAL FOUNDATION

BLUEPRINT FOR “MORENT”

Step 1: Define Technical Requirements

Frontend Requirements:

- **Essential Pages:**
  - **Home Page:** Showcase the service, car categories, and offers.
  - **Car Listings:** Display available cars with filters for model, price, and availability.
  - **Car Details:** Provide detailed information about selected cars (features, pricing, images).
  - **Booking/ Payment Page:** Allow users to select billing info, rental info, payment method, and confirmation for booking.
  - **Admin Dashboard:** Display booking details, manage rental status, and oversee car availability.
- **Design Features:**
  - Fully responsive design for mobile and desktop users.
  - Intuitive user interface with easy navigation and clear calls-to-action.

Backend (Sanity CMS)

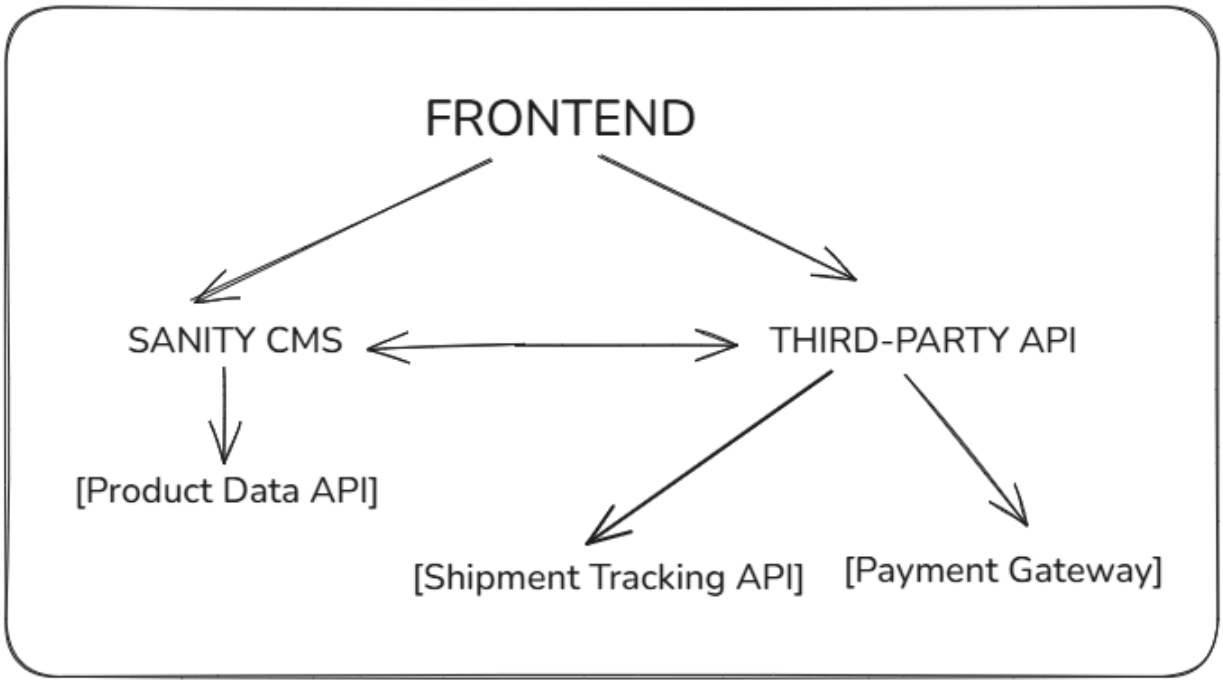
- **Purpose:**
  - Manage car data, customer details, and booking records efficiently.
- **Schemas:**
  - **Car Schema:** Fields for model, price, availability, features, and images.
  - **Booking Schema:** Fields for customer information, car ID, rental dates, and payment status.
  - **Customer Schema:** Fields for name, email, contact information, and rental history.

Third-Party APIs

- **Integrations:**
    - APIs for shipment tracking, payment gateways, and other required backend services.
    - API securely process frontend functionality

Step 2: Design System Architecture

Architecture Diagram



Key Workflows to Include:

User Registration:

- User creates an account via email/password or social login.
- User data is securely stored in **Sanity CMS**.
- A confirmation email is sent to verify the registration.

Car Browsing:

- User views available cars categorized by location, type, and price.
- Car data is fetched dynamically from **Sanity API**.
- Filter and search options allow users to refine their choices.

Booking Placement:

- User selects a car, rental dates, and any additional services.
- Booking details are added to the cart and proceed to checkout.
- Finalized booking details are saved in **Sanity CMS** for tracking.

Tracking and Updates:

- Booking status updates are fetched through a **Third-Party API**.
- Real-time booking or shipment details are displayed on the frontend for the user.

Step 3: Plan API Requirements

Endpoint	Method	Description	Example Response
/cars	GET	Fetch all available cars.	`{ "id": 1, "model": "SUV", "price": 100 }`
/availability	POST	Check car availability for dates.	`{ "carId": 1, "available": true }`
/booking	POST	Create a new booking.	`{ "bookingId": 123, "status": "Confirmed" }`
/payment	POST	Process payment securely.	`{ "paymentId": 456, "status": "Success" }`

API Endpoints:

The **MORENT** car rental marketplace relies on several key API endpoints to handle different user actions and ensure smooth operations:

- **/cars (GET):**
  This endpoint is used to fetch all available car details from the database. It provides information such as the car ID, model, price, and any other relevant details required to display cars to users.
- **/availability (POST):**
  This endpoint is designed to check car availability for specific dates. By providing the car ID and requested rental dates, the system verifies and returns whether the car is available or not.
- **/booking (POST):**
  This endpoint allows users to create a new booking. The booking details, including car ID, user information, and rental dates, are sent to this endpoint. The system processes the booking and stores it in the database, returning a success status and a unique booking ID.
- **/payment (POST):**
  This endpoint is used to process payments securely. Payment details, such as booking ID, payment method, and amount, are sent to this endpoint. The system processes the payment and returns a success status along with a unique payment ID.

CREATED BY: SOWAIBA NAZ
ROLL NO. 490877