Day 2:-

Planning The Technical Foundation

1. Technical Requirments:-

Frontend Requirements:-

- Develop a user-friendly, responsive interface optimized for mobile and desktop.
- Key pages to include:
  - Homepage.
  - Car listing / categories.
  - Car Detail.
  - Booking Cart / Payment.
  - Checkout / Admin
  - Order Confirmation.

Backend Requirments:

- Use Sanity CMS to efficiently manage car listing and details.

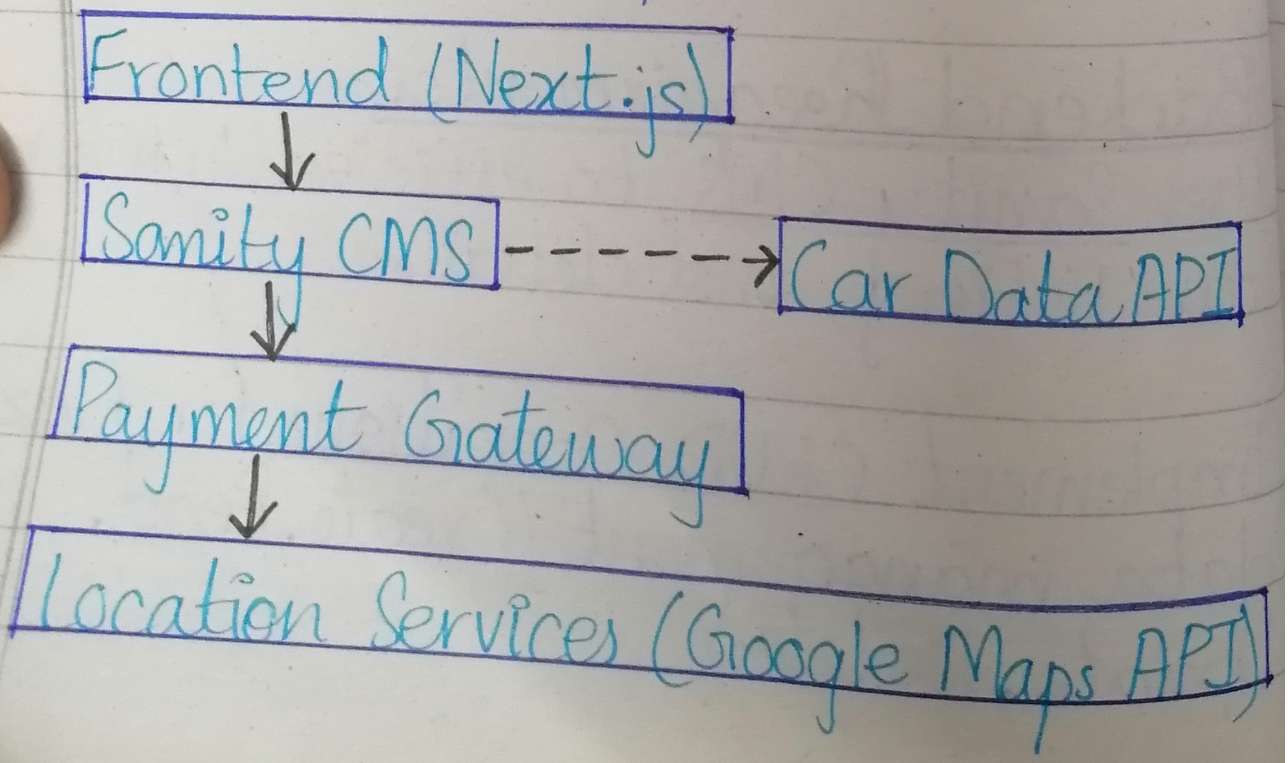
- Implement CRUD Operations for flexible data management (Create, Read, Update, Delete.)

2. Design System Architecture:

Utilize a modular, scalable architecture to support future growth.

Components:

- Frontend:- Built with react or Next.js for smooth user interaction
- Backend:- Sanity CMS for content management.
- Third-Party APIs:- Integrate payment gateways (e.g: Paypal, stripe etc) and location service (e.g: Google Maps).



Key Workflows:

- **Car Search:**

User enter search criteria

↓

Frontend queries Sanity API with filters

↓

Results Displayed.

- **Booking Process:**

User selects a car

↓

Enter retal dates and personal details

↓

Payment processed through the gateway

↓

Booking confirmed in Sanity

- **Rental Management:**

User view booking details, manages rental agreements, and recieves notifications.

3. Plan API Requirment:

Define essential API endpoints to ensure data handling.

- GET/cars: Retrieve a list of available cars.
- POST/book: Save booking details
- PUT/update-booking: Modify an existing booking.
- DELETE/cancel-booking: Remain a booking.

Authentication:
Use token to ensure secure access and protect user data.

4. Technical Documentation:-

Technical documentation should

include:

System Architecture:-

System components and their relation-ships.

API Usage:-

API integration and dataflow.

User Booking Journey:-

-User → Brauser Cars → View Car Details

→ Add to Cart → Proceed to checkout

→ Payment proceed via gateway →

order confirmation → deliver.