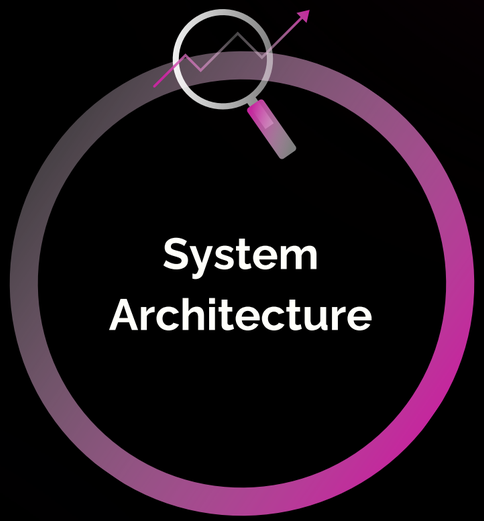
Marketplace Builder
Hackathon Day2

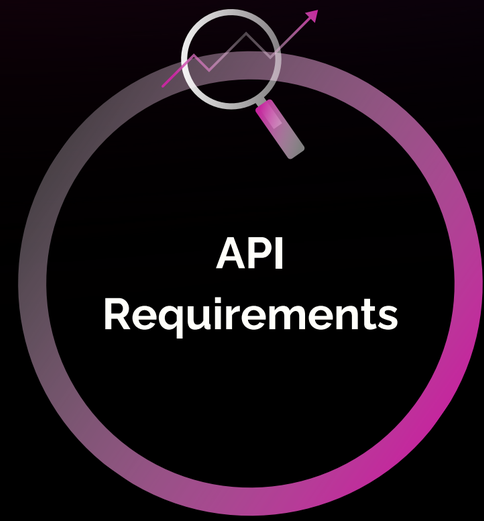
Planning

The Technical Foundation

Presented by
Areesha Kainat

Day 2 Activities: Transitioning to Technical Planning





Next Slide

Technical Requirement :

1.Frontend:

Next.js & Tailwind CSS: Responsive, modern design for all devices.
Key Pages: Home, Categories, Car Details, Dashboard, Payment, Cart, Checkout, Order Confirmation.
Quick Booking: Easy search filters by location, date, and car type.

2..Backend (Sanity CMS):

Schema Design: Manage car details, customer info, and orders.
Real Time Updates: Dynamic pricing and availability.

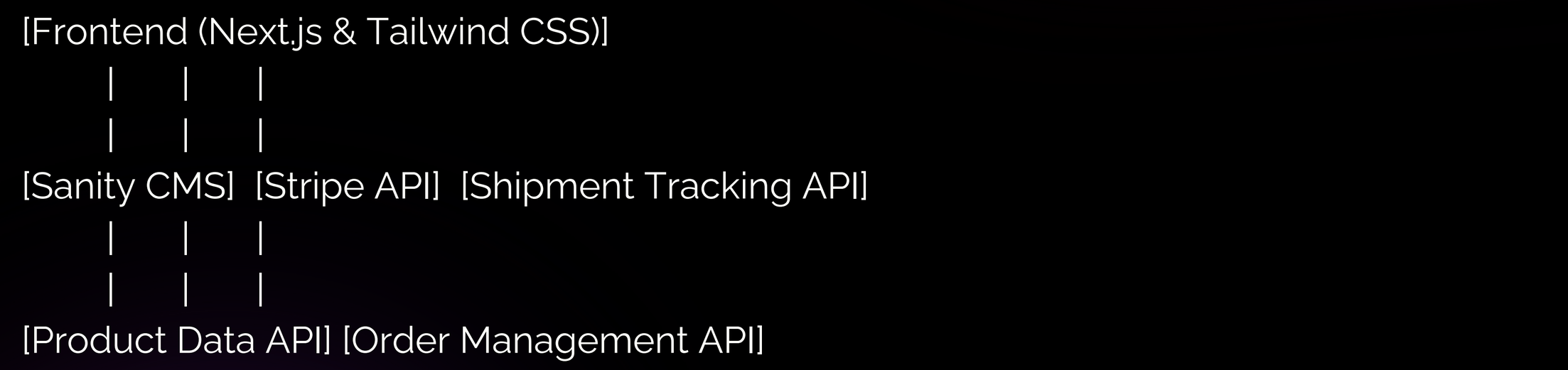
3.Third-Party APIs:

Stripe Payments: Safe and smooth payments in multiple currencies.
Shipment Tracking: Real-time updates for car delivery/pickup.

Next Slide

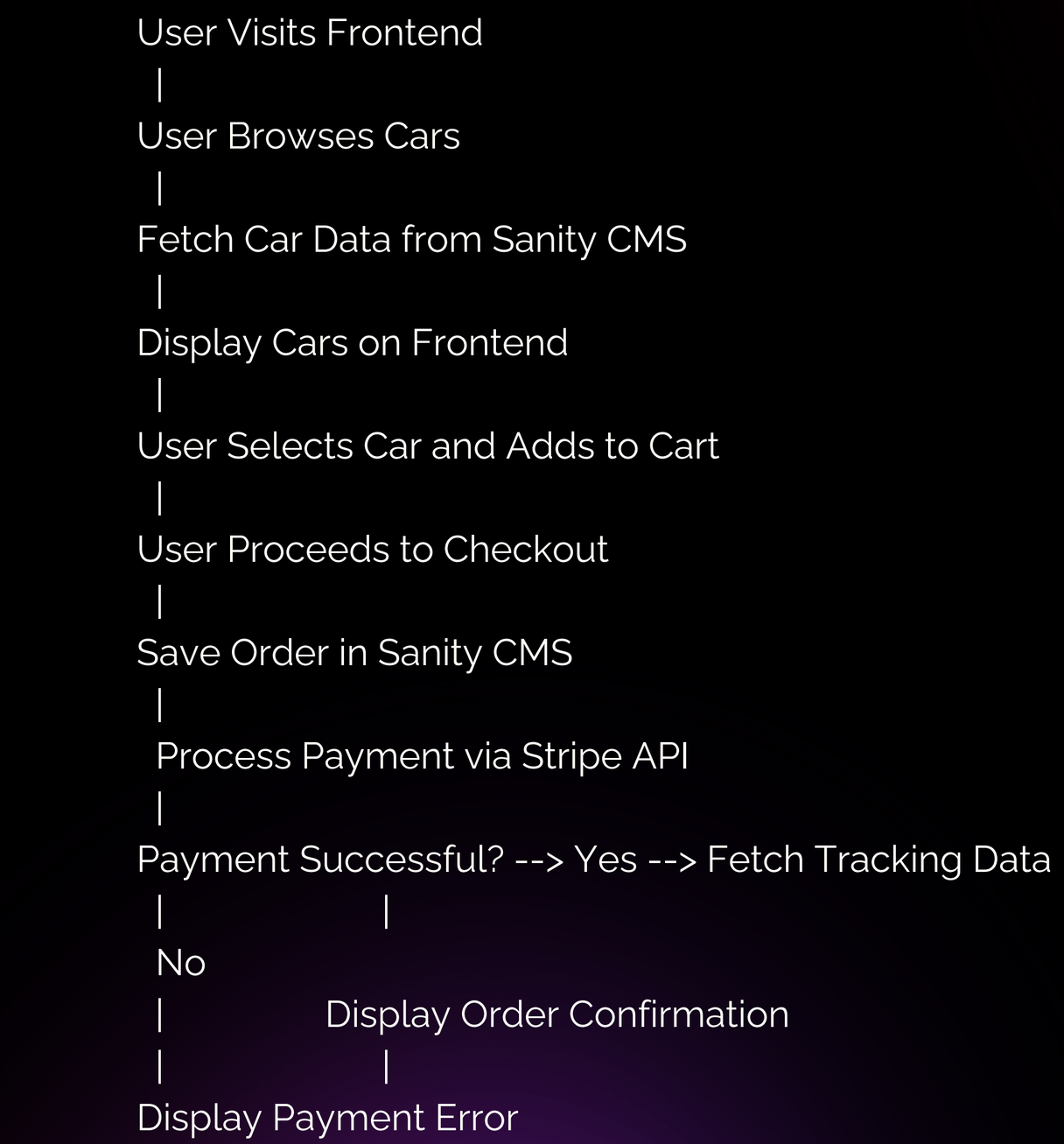
2.Design System Architecture:

Software Architecture:



Next Slide

Flowchart:



Next Slide

Algorithm:

Product Browsing:

Fetch car data from Sanity CMS and display dynamic listings.

Order Placement:

Add car to cart, proceed to checkout, and save order in Sanity CMS.

Payment Processing:

Process payment via Stripe API and update order status.

Shipment Tracking:

Fetch and display tracking details from the Tracking API.

Next Slide

3.Plan API Requirements

1. **Fetch Car** Categories
2. **Add** Rental Details
3. **Create** New Order
4. **Track** Shipment

Next Slide

1. Fetch Car Categories

- Endpoint: /category
- Method: GET
- What it does: Gets all car categories from Sanity CMS.
- Response: Car details (ID, name, price, image).

Next Slide

2. Add Rental Details

- Endpoint: /rental-duration
- Method: POST
- What it does: Saves rental details for a specific car.
- Payload: { "productId": 456, "duration": "7 days", "deposit": 500 }
- Response: { "confirmationId": 789, "status": "Success" }

Next Slide

3. Create New Order

- Endpoint: /orders
- Method: POST
- What it does: Saves a new order in Sanity CMS.
- Payload: Customer info, car details, payment status.

Next Slide

4. Track Shipment

- Endpoint: /shipment
- Method: GET
- What it does: Fetches real-time tracking info for car delivery/pickup.
- Response: Shipment ID, order ID, status, expected delivery date.

Next Slide

Marketplace Technical Foundation

1. System Architecture Overview

- **Frontend:** Next.js with Tailwind CSS for UI and user interactions.
- **Sanity CMS:** Manages car data, customer details, and orders.
- **Third-Party APIs:**
  Stripe: For payment processing.
  Shipment Tracking: For real-time delivery updates.

Next Slide

Diagram:

[Frontend] → [Sanity CMS] → [Stripe API]
↓
[Shipment Tracking API]

2. KEY WORKFLOWS

User Browsing:

- User visits the site.
- Frontend fetches car data from Sanity CMS.
- Displays cars with filters (type, price).

Order Placement:

- User selects car and adds to cart.
- Proceeds to checkout.
- Order details saved in Sanity CMS.

Payment & Tracking:

- Stripe processes payment.
- Shipment tracking updates provided.

Next Slide

Marketplace Technical Foundation

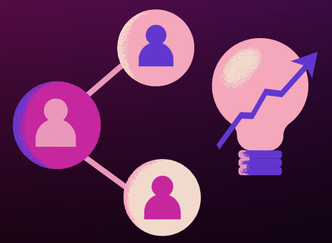
3. Category-Specific Instructions (Rental eCommerce)

Rental Workflows:

- Track rental duration, condition reports, and returns.
- Example Schema Fields: rentalDuration, depositAmount, conditionStatus.

Next Slide

4. API Endpoints



Endpoint	Method	Purpose	Response Example
/cars	GET	Fetches all car details	{ "id": 1, "name": "SUV X", "price": 100 }
/rental-duration	POST	Saves rental details	{ "confirmationId": 789, "status": "Success" }
/orders	POST	Creates a new order	{ "orderId": 123, "status": "Confirmed" }
/shipment	GET	Tracks car delivery	{ "shipmentId": 456, "status": "In Transit" }

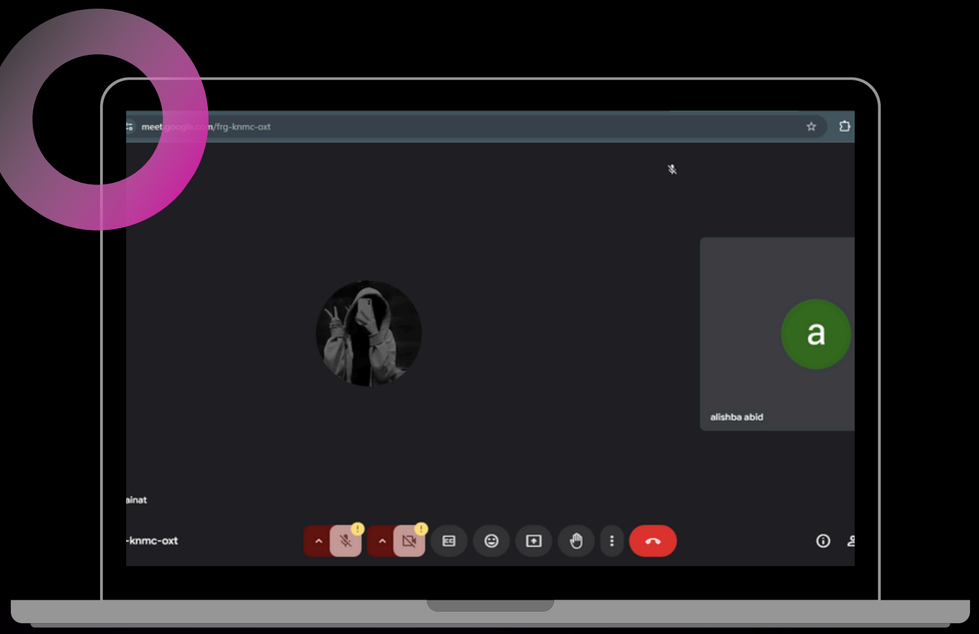
Next Slide

5. Sanity
Schema Example

```
export default {
  name: "car",
  type: "document",
  title: "Car",
  fields: [
    { name: "name", type: "string", title: "Name" },
    { name: "type", type: "string", title: "Type" },
    { name: "price", type: "number", title: "Price" },
    { name: "capacity", type: "string", title: "Fuel Capacity" },
    { name: "transmission", type: "string", title: "Transmission" },
    { name: "seats", type: "string", title: "Seats" },
    { name: "img", type: "image", title: "Image", options: {
hotspot: true } },
    { name: "favorite", type: "boolean", title: "Favorite" },
    { name: "slug", type: "slug", title: "Slug", options: { source:
"name", maxLength: 200 } },
  ],
};
```

Next Slide

Collaboration on Google Meet



Here’s a snapshot from my Google Meet session with my friend. We brainstormed ideas, discussed technical plans, and exchanged feedback to refine our project. Working together in real-time helped us solve challenges faster and share innovative ideas, making our project stronger.

Next Slide

Thank You!

Only 5 days to go! The journey is getting exciting, and we're pushing forward with new ideas and solutions! Let's keep the momentum going!