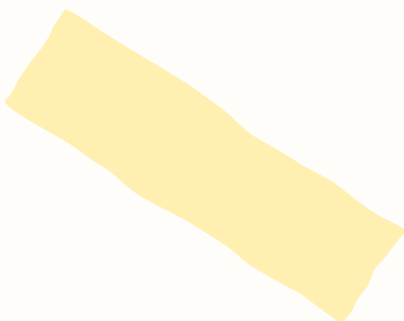# MARKETPLACE TECHNICAL FOUNDATION - MORENT CAR RENTAL WEBSITE

Rental E-Commerce Website

Planning the approch

User Steps

Sign In: User logs in using Google through NextAuth, with their data saved in Sanity.

Choose Car: User selects a car, reads specifications, and fills out a form powered by Zod, including user, rental, and payment data.

Submit Form: Form data is saved in Sanity, and a shipment label is generated through ShipEngine.

Confirmation: User sees a confirmation message and is redirected to the booking page.

Tracking: Tracking data is shown on the booking page.

Payment: Payment is processed through Stripe, and the status is displayed on the booking page.

Delivery: Car is delivered to the drop-off location on the specified date and time.

System Components

Frontend: NextJS Client, TailwindCSS, TypeScript, Zod.

Backend: Handles authentication, design, car search, favorites, data tables, shipment, and real-time tracking.

Database: Stores user, car, reservation, payment, and shipment data.

APIs: Sanity API, ShipEngine API, Stripe API, NextJS API.

Data Flow

Data is fetched, saved, and processed through various steps, including tracking, payment details, car data, and user data.

Entities

User: Name, email, image, ID, phone.

Car: Name, image, description, ID, availability, rental price per day.

Reservation: Reservation ID, car ID, user ID, locations, dates, total days, shipment costs.

Payment: Payment ID, reservation ID, status, method, amount.

Shipment: Shipment ID, car ID, user ID, locations, dates.

API Endpoints

Authentication

- Endpoint: /api/auth/signin/google
- Method: GET
- Purpose:
  - Initiates the OAuth flow for Google Sign-In.
- Flow:
  - Redirects the user to Google's OAuth authorization endpoint.
  - After successful authentication, Google redirects the user back to the NextAuth callback/redirect URL.

Fetching

- Endpoint: /api/cars
- Method: GET
- Purpose:
  - Fetches all cars from Sanity.
- Flow:
  - When the page loads, cars will be fetched and displayed on the pages.

Booking

- Endpoint: /api/new-booking
- Method: POST
- Purpose:
    - Takes UserData, CarData, PaymentData, and RentalData inside the body to initiate a booking for that specific car and user.
    - Saves the data in Sanity.
- Flow:
   - After the form is submitted, this endpoint is executed to initiate the booking and save data in Sanity.

Payment

- Endpoint: /api/payment
- Method: POST
- Purpose:
    - To make a payment for the rent.
- Flow:
    - Takes BookingData.
    - Processes the payment using Stripe.
    - Saves payment data in Sanity.
    - Returns the payment status.

Shipment

- Endpoint: /api/shipment
- Method: POST
- Purpose:
  - To initiate a shipment and return tracking data in the response.
- Flow:
  - Takes rental details and starts the shipment of the car to the user.

Flow Diagram

Car Rental Process Diagram
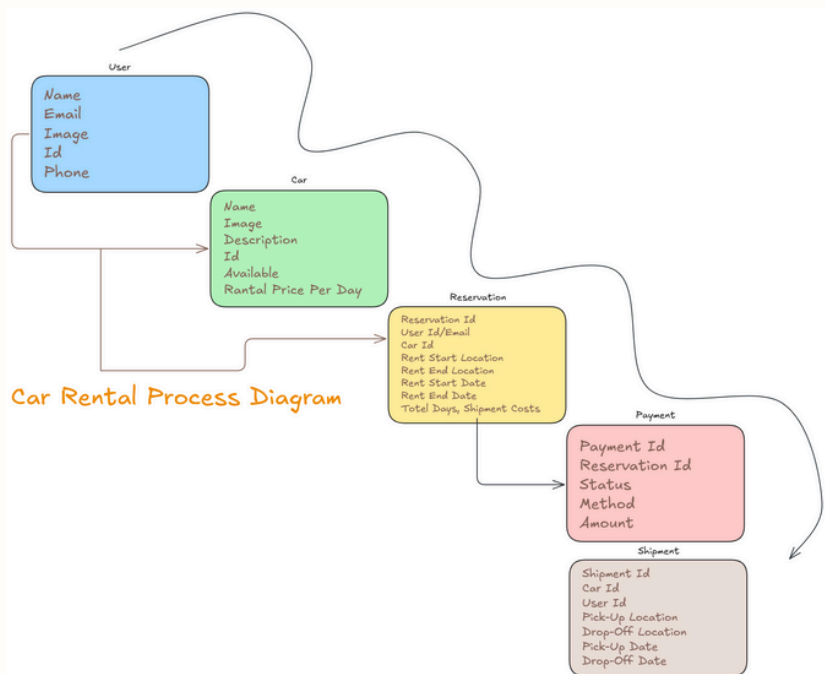
First user signin/login, using google provider through nextauth, their data will be saved inside sanity

/
/category
/details/car
/payment
/bookings
/hearted
/admin

User chooses their favorite car for rent, read car specifications, and fills the details form which is powered by zod,
this form contains inputs for user data, rental data and payment data

User submits the form, data is saved inside sanity, shipment label will be generated through shipengine
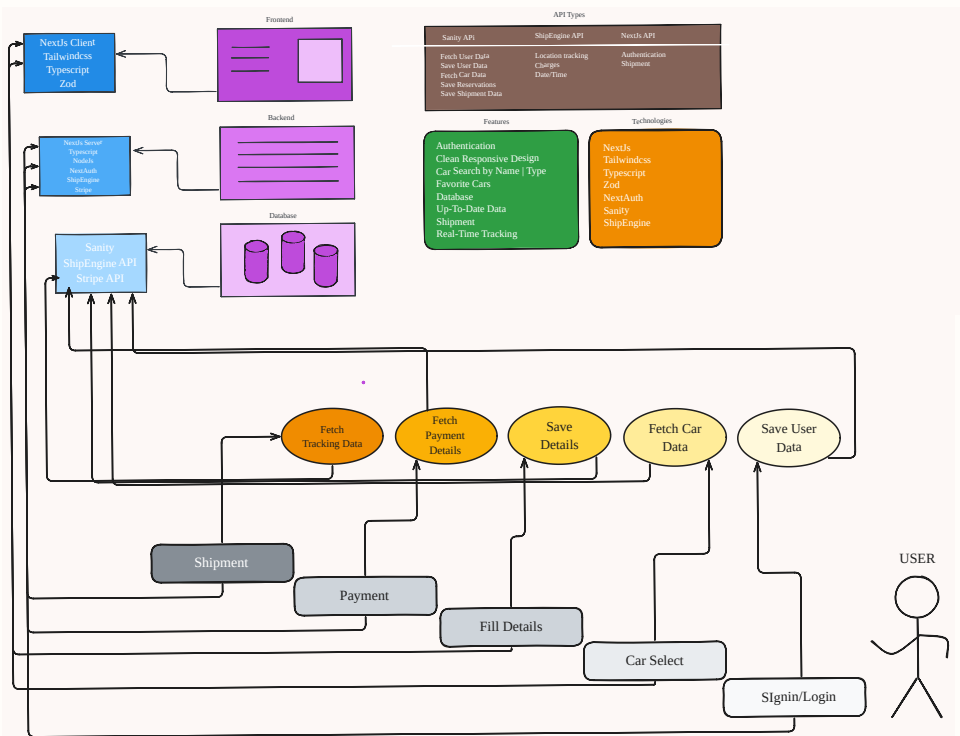
User will see a conformation message and will be redirected to the booking page

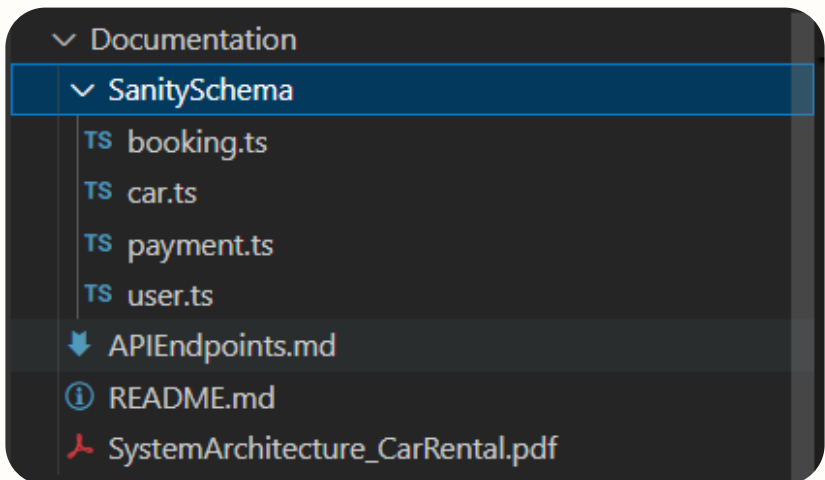Tracking data will be shown to the user inside the booking page

Payment has been sent to the platform, through stripe, user can see the payment status inside the booking page

The car will be delivered at the drop-off location at the drop-off datetime

Brief System Design Plan

Documents File Structure

User Schema

```javascript
import { defineField, defineType } from 'sanity';

export default defineType({
  name: 'user',
  title: 'User',
  type: 'document',
  fields: [
    defineField({
      name: 'name',
      title: 'Name',
      type: 'string',
    }),
    defineField({
      name: 'email',
      title: 'Email',
      type: 'string',
    }),
    defineField({
      name: 'image',
      title: 'Image',
      type: 'url',
    }),
    defineField({
      name: 'phone',
      title: 'Phone',
      type: 'string',
    }),
  ],
});
```

Car Schema

```javascript
import { defineField, defineType } from 'sanity';

export default defineType({
  name: 'car',
  title: 'Car',
  type: 'document',
  fields: [
    defineField({
      name: 'name',
      title: 'Name',
      type: 'string',
    }),
    defineField({
      name: 'current_price',
      title: 'Current Price',
      type: 'string',
    }),
    defineField({
      name: 'Available',
      title: 'Available',
      type: 'number',
    }),
    defineField({
      name: 'old_price',
      title: 'Old Price',
      type: 'string',
    }),
    defineField({
      name: 'heart',
      title: 'Heart',
      type: 'boolean',
    }),
```

Car Schema

```
defineField({
name: 'car_type',
title: 'Car Type',
type: 'string',
}),
defineField({
name: 'icons',
title: 'Icons',
type: 'boolean',
}),
defineField({
name: 'card_type',
title: 'Card Type',
type: 'string',
}),
defineField({
name: 'slug',
title: 'Slug',
type: 'slug',
options: {
source: 'name',
maxLength: 96,
},
}),

// Detail Page Specific Fields
defineField({
name: 'desc',
title: 'Description',
type: 'text',
}),
```

Car Schema

```
defineField({
name: 'image',
title: 'Image',
type: 'image',
options: {
hotspot: true,
}
}),
defineField({
name: 'capacity',
title: 'Capacity',
type: 'string',
}),
defineField({
name: 'steering',
title: 'Steering',
type: 'string',
}),
defineField({
name: 'gasoline',
title: 'Gasoline',
type: 'string',
}),
defineField({
name: 'reviews',
title: 'Reviews',
type: 'number',
}),
```

Car Schema

```
// Images as an array for the detail page
defineField({
name: 'images',
title: 'Images',
type: 'array',
of: [
{
type: 'image',
options: {
hotspot: true,
},
},
],
}),
],
});
```

Booking Schema

```
export default {
  name: 'booking',
  type: 'document',
  title: 'Booking',
  fields: [
    {
      name: 'user_name',
      type: 'string',
      title: 'User Name',
    },
    {
      name: 'user_address',
      type: 'string',
      title: 'User Address',
    },
    {
      name: 'user_phone',
      type: 'string',
      title: 'User Phone',
    },
    {
      name: 'user_email',
      type: 'string',
      title: 'User Email',
    },
    {
      name: 'car',
      type: 'reference',
      title: 'Car',
      to: [{ type: 'car' }],
    },
```

Booking Schema

```
{
name: 'drop-off-location',
type: 'string',
title: 'Drop-Off Location',
},
{
name: 'rent_start_date',
type: 'string',
title: 'Rent Start Date',
},
{
name: 'rent_end_date',
type: 'string',
title: 'Rent End Date',
},
{
name: 'total_price',
type: 'number',
title: 'Total Price',
},
],
};
```

Payment Schema

```
export default {
  name: 'payment',
  type: 'document',
  title: 'Payment',
  fields: [
    {
      name: 'payment_id',
      type: 'string',
      title: 'Payment ID',
    },
    {
      name: 'booking',
      type: 'reference',
      title: 'Booking',
      to: [{ type: 'booking' }],
    },
    {
      name: 'user',
      type: 'reference',
      title: 'User',
      to: [{ type: 'user' }],
    },
    {
      name: 'amount',
      type: 'number',
      title: 'Amount',
    },
    {
      name: 'status',
      type: 'string',
      title: 'Payment Status',
    },
```

Payment Schema

```
{
name: 'method',
type: 'string',
title: 'Payment Method',
},
{
name: 'payment_date',
type: 'datetime',
title: 'Payment Date',
},
],
};
```