Day 3 – API Integration Report for

“Morent Car Rentals”

Objective:

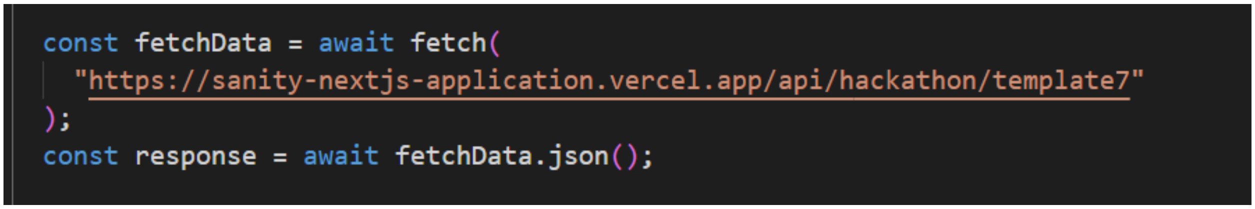
The objective of this report is to document the process of integrating APIs and migrating data into Sanity CMS for the "Morent Car Rentals" project. This includes schema adjustments, data migration steps, and API integration into the frontend using Next.js.

API Integration Process:

The API provided the following endpoints:

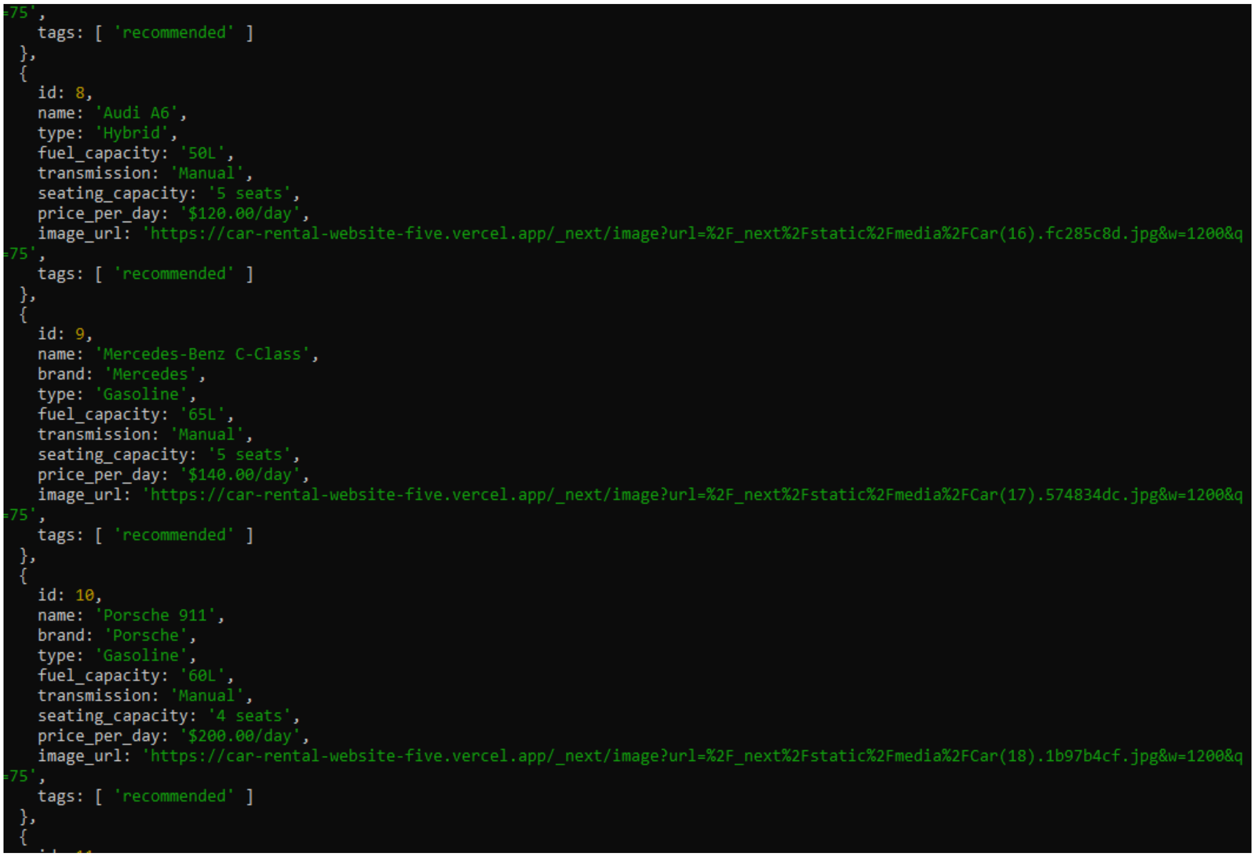
API URL:

https://sanity-nextjs-application.vercel.app/api/hackathon/template7



```
const fetchData = await fetch(
  "https://sanity-nextjs-application.vercel.app/api/hackathon/template7"
);
const response = await fetchData.json();
```

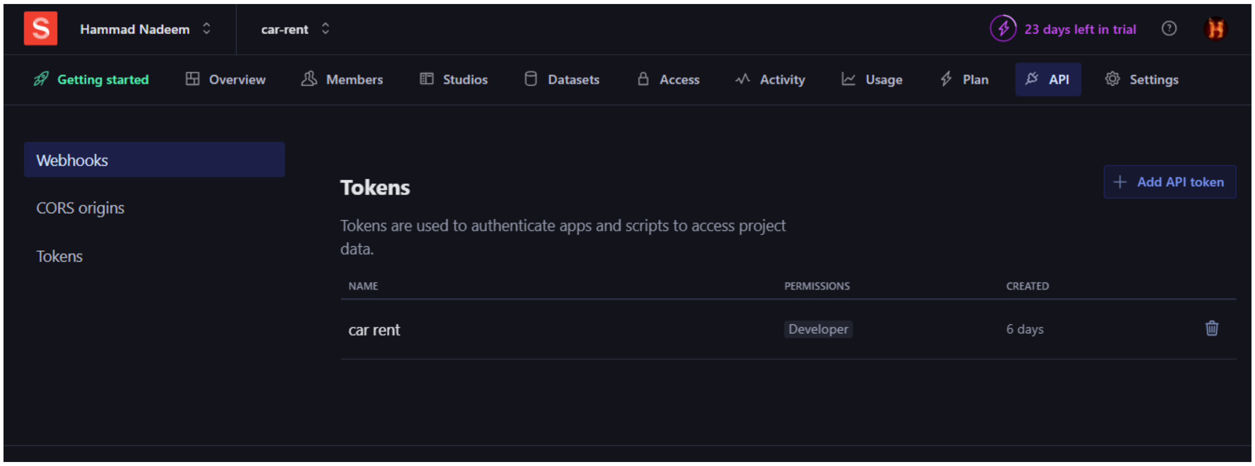
Console Output



```
=75',
    tags: [ 'recommended' ]
  },
  {
    id: 8,
    name: 'Audi A6',
    type: 'Hybrid',
    fuel_capacity: '50L',
    transmission: 'Manual',
    seating_capacity: '5 seats',
    price_per_day: '$120.00/day',
    image_url: 'https://car-rental-website-five.vercel.app/_next/image?url=%2F_next%2Fstatic%2Fmedia%2FCar(16).fc285c8d.jpg&w=1200&q
=75',
    tags: [ 'recommended' ]
  },
  {
    id: 9,
    name: 'Mercedes-Benz C-Class',
    brand: 'Mercedes',
    type: 'Gasoline',
    fuel_capacity: '65L',
    transmission: 'Manual',
    seating_capacity: '5 seats',
    price_per_day: '$140.00/day',
    image_url: 'https://car-rental-website-five.vercel.app/_next/image?url=%2F_next%2Fstatic%2Fmedia%2FCar(17).574834dc.jpg&w=1200&q
=75',
    tags: [ 'recommended' ]
  },
  {
    id: 10,
    name: 'Porsche 911',
    brand: 'Porsche',
    type: 'Gasoline',
    fuel_capacity: '60L',
    transmission: 'Manual',
    seating_capacity: '4 seats',
    price_per_day: '$200.00/day',
    image_url: 'https://car-rental-website-five.vercel.app/_next/image?url=%2F_next%2Fstatic%2Fmedia%2FCar(18).1b97b4cf.jpg&w=1200&q
=75',
    tags: [ 'recommended' ]
  },
  {
```

Migrating Data to Sanity CMS:

Step 1: Generate Sanity API Token.

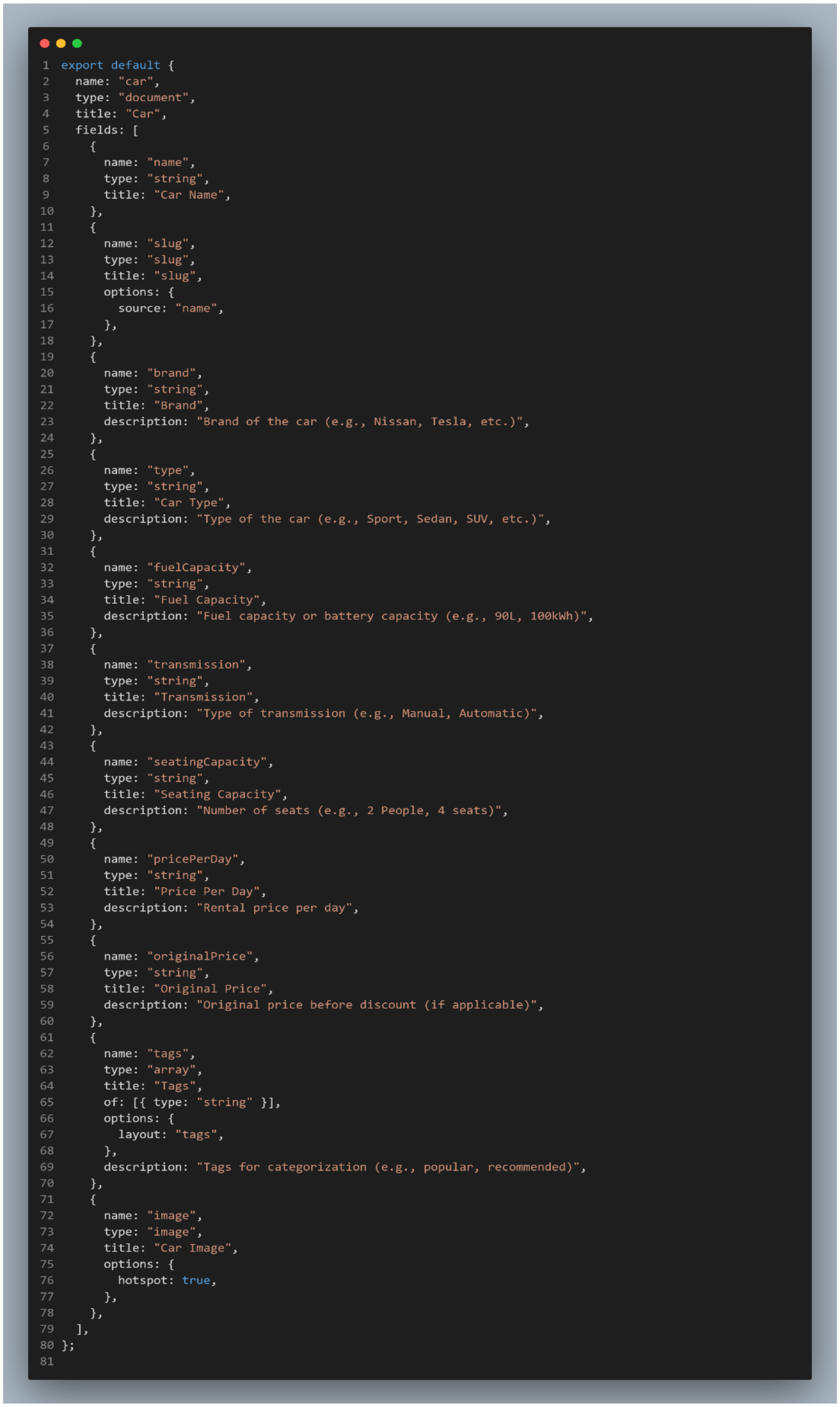


Tokens

Tokens are used to authenticate apps and scripts to access project data.

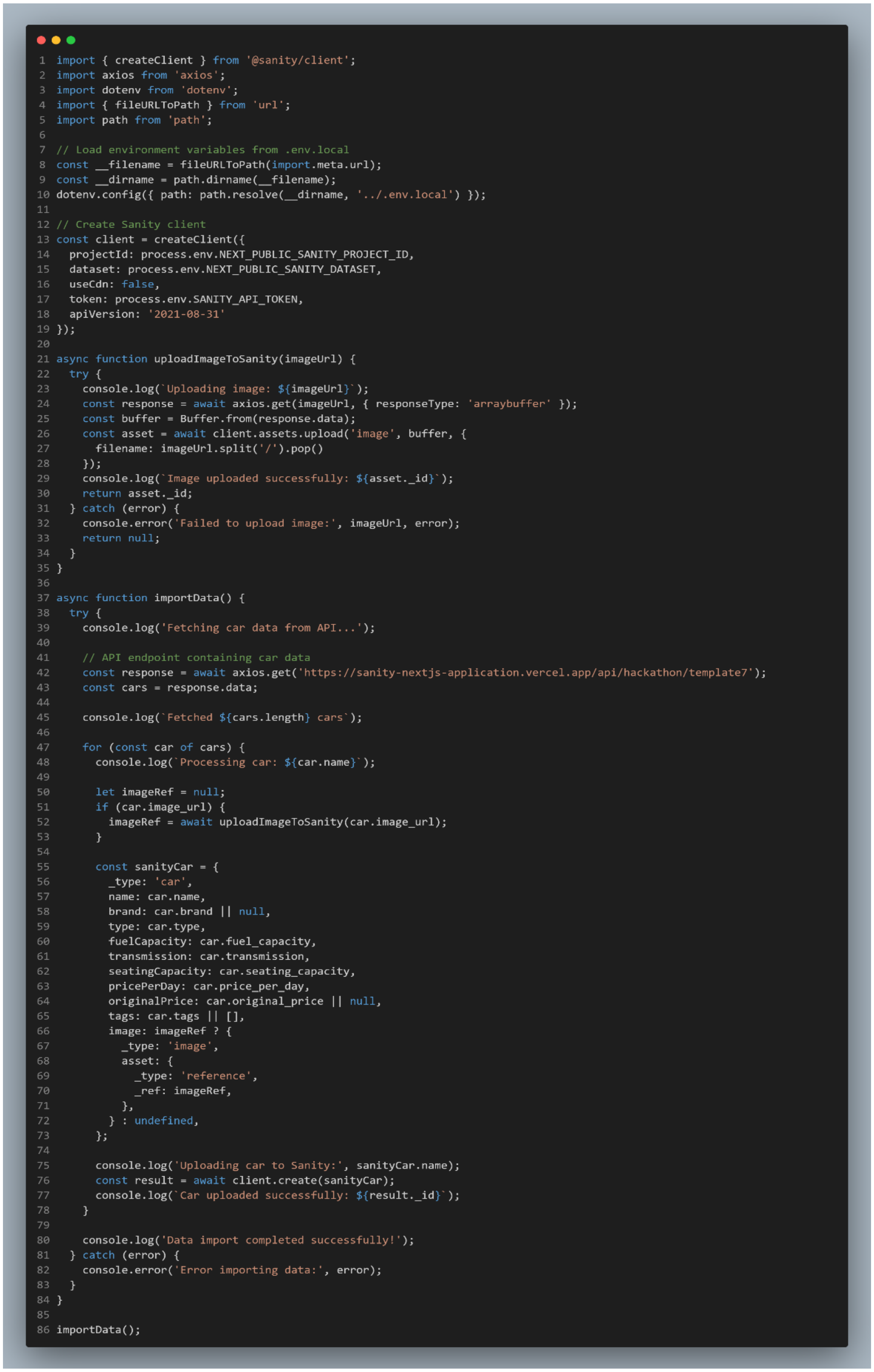
NAME	PERMISSIONS	CREATED
car rent	Developer	6 days

Step 2: Create Car Schema.



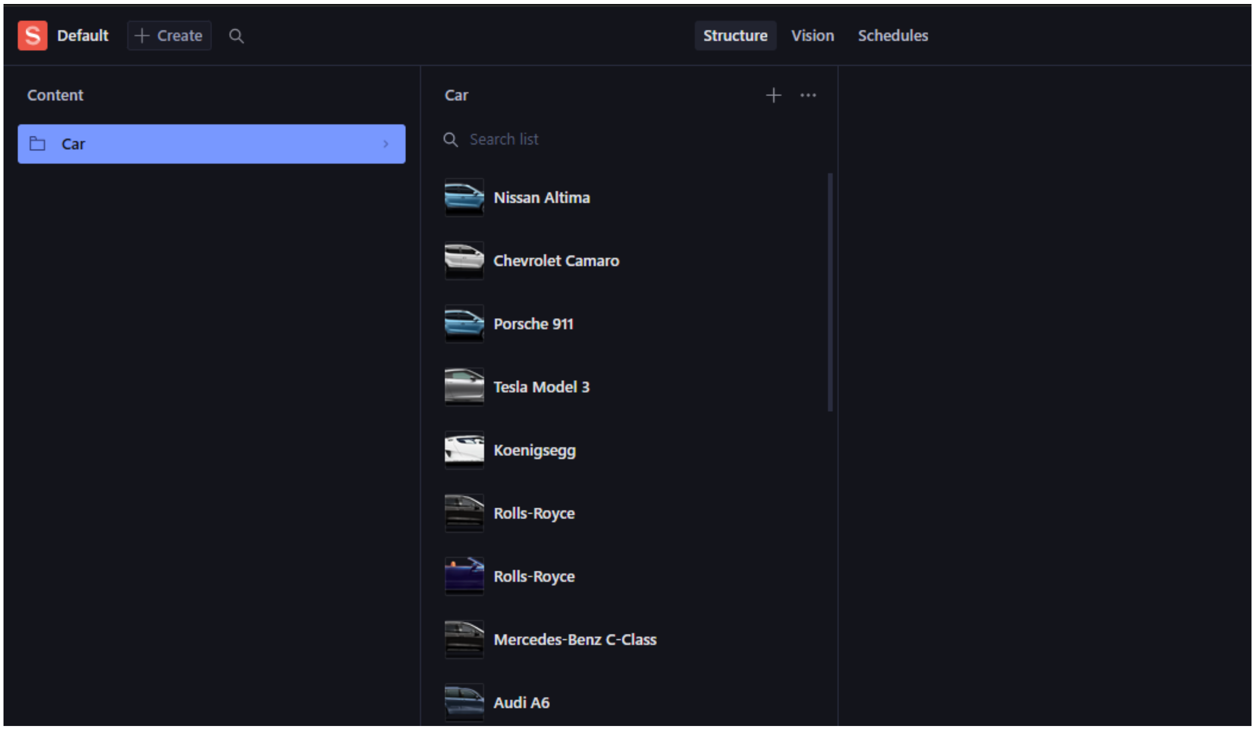
```
export default {
  name: "car",
  type: "document",
  title: "Car",
  fields: [
    {
      name: "name",
      type: "string",
      title: "Car Name",
    },
    {
      name: "slug",
      type: "slug",
      title: "slug",
      options: {
        source: "name",
      },
    },
    {
      name: "brand",
      type: "string",
      title: "Brand",
      description: "Brand of the car (e.g., Nissan, Tesla, etc.)",
    },
    {
      name: "type",
      type: "string",
      title: "Car Type",
      description: "Type of the car (e.g., Sport, Sedan, SUV, etc.)",
    },
    {
      name: "fuelCapacity",
      type: "string",
      title: "Fuel Capacity",
      description: "Fuel capacity or battery capacity (e.g., 90L, 100kWh)",
    },
    {
      name: "transmission",
      type: "string",
      title: "Transmission",
      description: "Type of transmission (e.g., Manual, Automatic)",
    },
    {
      name: "seatingCapacity",
      type: "string",
      title: "Seating Capacity",
      description: "Number of seats (e.g., 2 People, 4 seats)",
    },
    {
      name: "pricePerDay",
      type: "string",
      title: "Price Per Day",
      description: "Rental price per day",
    },
    {
      name: "originalPrice",
      type: "string",
      title: "Original Price",
      description: "Original price before discount (if applicable)",
    },
    {
      name: "tags",
      type: "array",
      title: "Tags",
      of: [{ type: "string" }],
      options: {
        layout: "tags",
      },
      description: "Tags for categorization (e.g., popular, recommended)",
    },
    {
      name: "image",
      type: "image",
      title: "Car Image",
      options: {
        hotspot: true,
      },
    },
  ],
};
```

Step 3: This code script used to fetch data from an external API and upload into sanity.



```
import { createClient } from '@sanity/client';
import axios from 'axios';
import dotenv from 'dotenv';
import { fileURLToPath } from 'url';
import path from 'path';

// Load environment variables from .env.local
const __filename = fileURLToPath(import.meta.url);
const __dirname = path.dirname(__filename);
dotenv.config({ path: path.resolve(__dirname, '../.env.local') });

// Create Sanity client
const client = createClient({
  projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
  dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
  useCdn: false,
  token: process.env.SANITY_API_TOKEN,
  apiVersion: '2021-08-31'
});

async function uploadImageToSanity(imageUrl) {
  try {
    console.log(`Uploading image: ${imageUrl}`);
    const response = await axios.get(imageUrl, { responseType: 'arraybuffer' });
    const buffer = Buffer.from(response.data);
    const asset = await client.assets.upload('image', buffer, {
      filename: imageUrl.split('/').pop()
    });
    console.log(`Image uploaded successfully: ${asset._id}`);
    return asset._id;
  } catch (error) {
    console.error('Failed to upload image:', imageUrl, error);
    return null;
  }
}

async function importData() {
  try {
    console.log('Fetching car data from API...');

    // API endpoint containing car data
    const response = await axios.get('https://sanity-nextjs-application.vercel.app/api/hackathon/template7');
    const cars = response.data;

    console.log(`Fetched ${cars.length} cars`);

    for (const car of cars) {
      console.log(`Processing car: ${car.name}`);

      let imageRef = null;
      if (car.image_url) {
        imageRef = await uploadImageToSanity(car.image_url);
      }

      const sanityCar = {
        _type: 'car',
        name: car.name,
        brand: car.brand || null,
        type: car.type,
        fuelCapacity: car.fuel_capacity,
        transmission: car.transmission,
        seatingCapacity: car.seating_capacity,
        pricePerDay: car.price_per_day,
        originalPrice: car.original_price || null,
        tags: car.tags || [],
        image: imageRef ? {
          _type: 'image',
          asset: {
            _type: 'reference',
            _ref: imageRef,
          },
        } : undefined,
      };

      console.log('Uploading car to Sanity:', sanityCar.name);
      const result = await client.create(sanityCar);
      console.log(`Car uploaded successfully: ${result._id}`);
    }

    console.log('Data import completed successfully!');
  } catch (error) {
    console.error('Error importing data:', error);
  }
}

importData();
```

Sanity Dashboard Screenshot



Nissan Altima

Chevrolet Camaro

Porsche 911

Tesla Model 3

Koenigsegg

Rolls-Royce

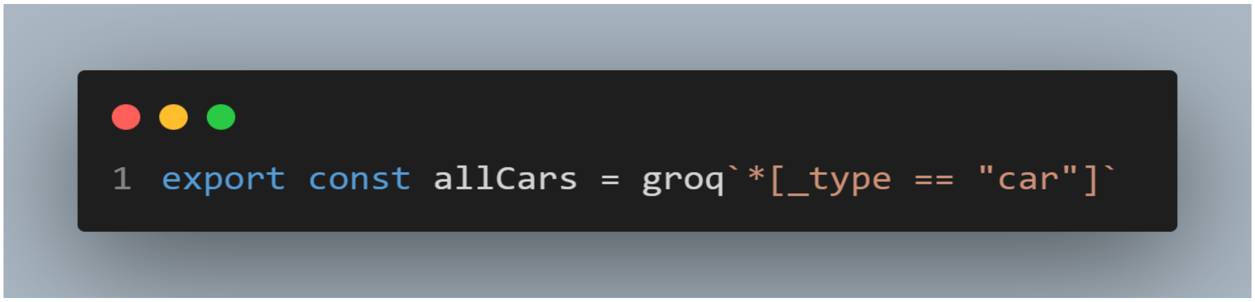
Rolls-Royce

Mercedes-Benz C-Class

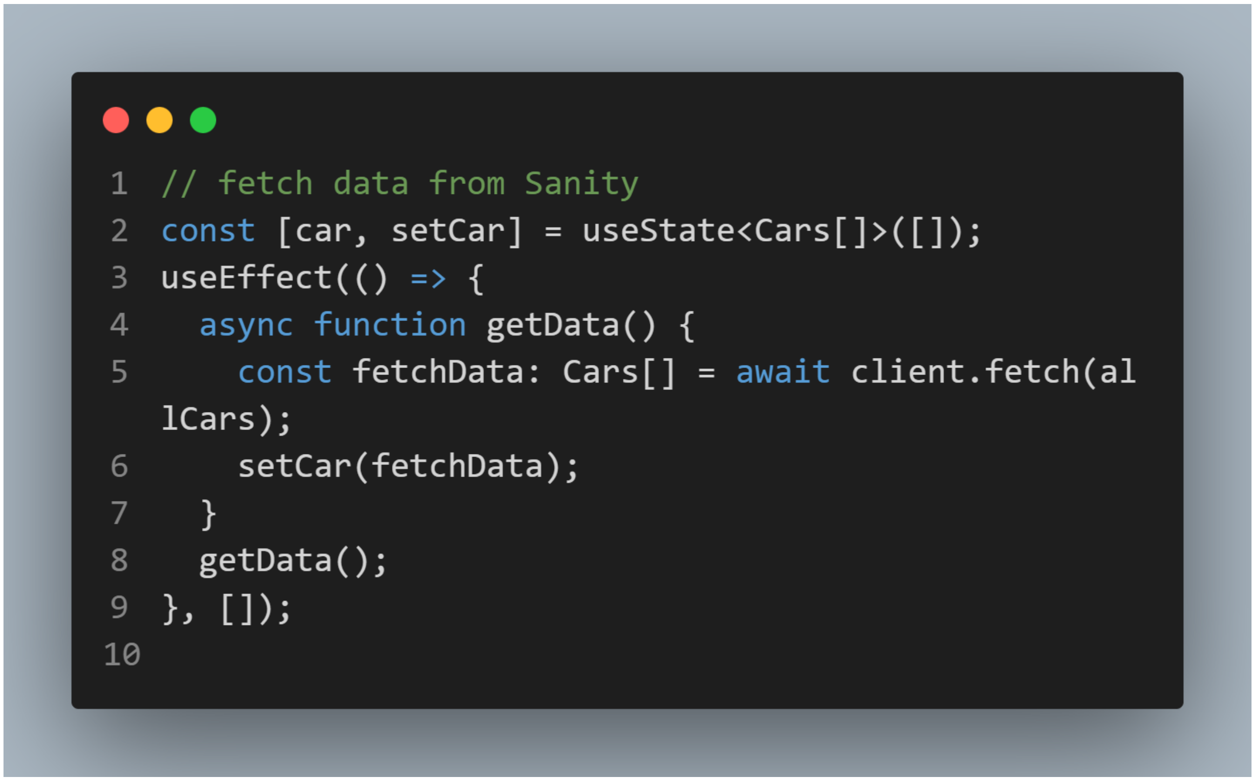
Audi A6

Frontend Integration

Rendering Data on Frontend

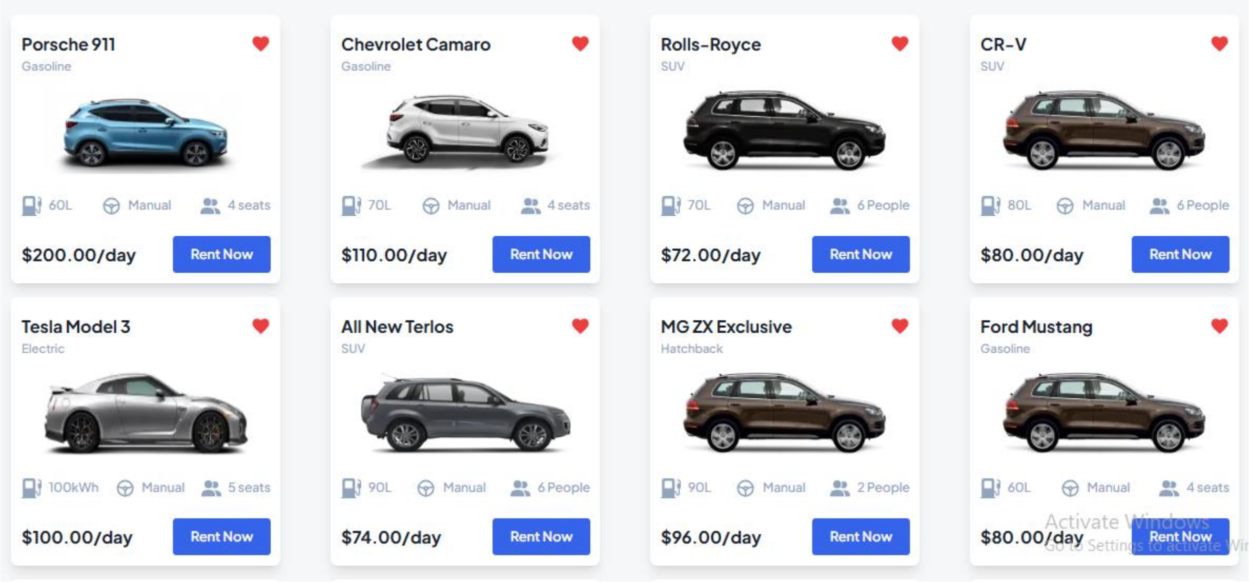


```
export const allCars = groq`*[_type == "car"]`
```



```
// fetch data from Sanity
const [car, setCar] = useState<Cars[]>([]);
useEffect(() => {
  async function getData() {
    const fetchData: Cars[] = await client.fetch(allCars);
    setCar(fetchData);
  }
  getData();
}, []);
```

Screenshot of the data listing page.



Porsche 911
Gasoline
60L · Manual · 4 seats
$200.00/day

Chevrolet Camaro
Gasoline
70L · Manual · 4 seats
$110.00/day

Rolls-Royce
SUV
70L · Manual · 6 People
$72.00/day

CR-V
SUV
80L · Manual · 6 People
$80.00/day

Tesla Model 3
Electric
100kWh · Manual · 5 seats
$100.00/day

All New Terlos
SUV
90L · Manual · 6 People
$74.00/day

MG ZX Exclusive
Hatchback
90L · Manual · 2 People
$96.00/day

Ford Mustang
Gasoline
60L · Manual · 4 seats
$80.00/day