Morent Car Rental

Abdu Subhan

—

Day 3

API INTEGRATION PROCESS

Step 1: Fetch Car Data

The car data is retrieved from an external API endpoint:

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

Each car object contains fields such as id, name, brand, type, fuel_capacity, transmission, seating_capacity, price_per_day, original_price, and image_url.

Step 2: Upload Car Images

The image_url field from the API is used to upload images to Sanity.

Using the Sanity client, the image is downloaded as a buffer using axios and then uploaded as an asset to Sanity.

Once uploaded, the image asset's _id is stored and referenced in the image field of the car object.

Step 3: Prepare Data for Sanity

After processing the image, a new car object is created with the required fields and mapped to the Sanity schema.

Optional fields (like original_price and tags) are set to null or default values if not present in the API response.

Step 4: Upload Car Data to Sanity

The transformed car object is uploaded to Sanity using the create method of the Sanity client.

Each car document is assigned a unique _id and includes all necessary data for rendering on the front end.

SCHEMA ADJUTMENT

```
export default {
  name: 'car',
  type: 'document',
  title: 'Car',
  fields: [
    {
      name: 'name',
      type: 'string',
      title: 'Car Name',
    },
    {
      name: 'id',
      type: 'number',
      title: 'ID',
    },
    {
      name: 'brand',
      type: 'string',
      title: 'Brand',
```

```
20.       description: 'Brand of the car (e.g., Nissan, Tesla, etc.)',
21.     },
22.     {
23.       name: 'type',
24.       type: 'string',
25.       title: 'Car Type',
26.       description: 'Type of the car (e.g., Sport, Sedan, SUV, etc.)',
27.     },
28.     {
29.       name: 'fuelCapacity',
30.       type: 'string',
31.       title: 'Fuel Capacity',
32.       description: 'Fuel capacity or battery capacity (e.g., 90L, 100kWh)',
33.     },
34.     {
35.       name: 'transmission',
36.       type: 'string',
37.       title: 'Transmission',
38.       description: 'Type of transmission (e.g., Manual, Automatic)',
39.     },
40.     {
41.       name: 'seatingCapacity',
42.       type: 'string',
43.       title: 'Seating Capacity',
44.       description: 'Number of seats (e.g., 2 People, 4 seats)',
45.     },
46.     {
47.       name: 'pricePerDay',
48.       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
      }
    }
  ],
};
```

78.

MIGRATION STEPS

I have used a migration script provided in the docs.

CODE MIGRATION SCRIPTS

```
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.NEXT_PUBLIC_SANITY_TOKEN,
  apiVersion: "2021-08-31",
});

async function uploadImageToSanity(imageUrl: string) {
```

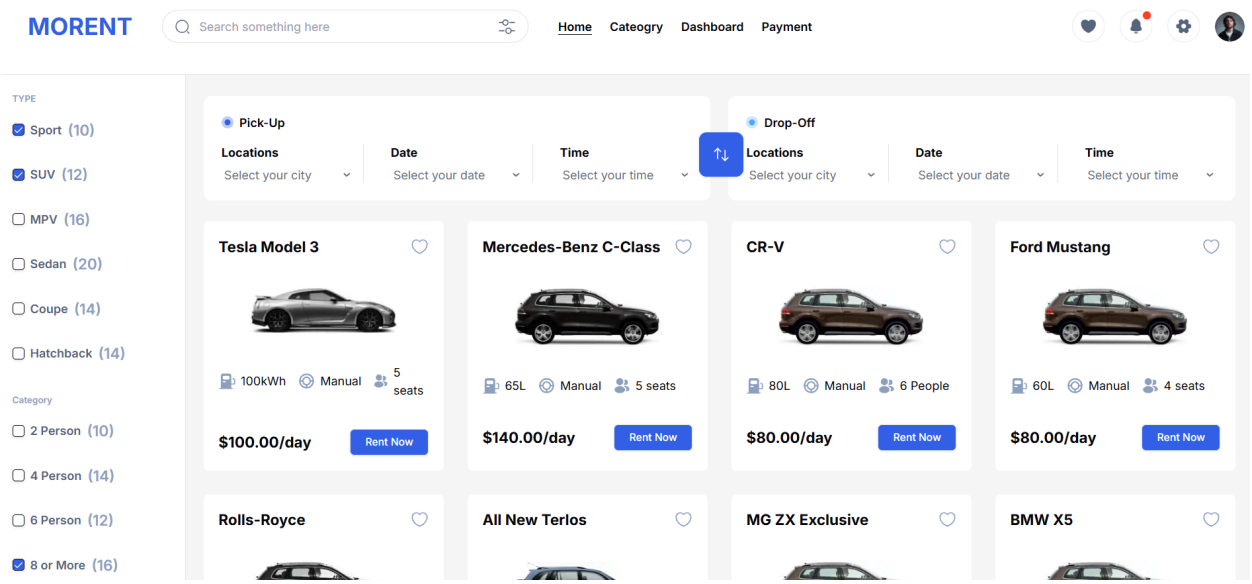
```
  try {
    console.log(`Uploading image: ${imageUrl}`);
    const response = await axios.get(imageUrl, { responseType: "arraybuffer" });
    const buffer = Buffer.from(response.data);
    const asset = await client.assets.upload("image", buffer, {
      filename: imageUrl.split("/").pop(),
    });
    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("1");
    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",
    id: car.id,
    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);
  console.log("2");
  const result = await client.create(sanityCar);
  console.log(`Car uploaded successfully: ${result._id}`);
  console.log("3");
}
```

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

export default importData;
```

SCREENSHOTS



Popular Car

View All

Rolls-Royce

70L Manual 4 People

$96.00/day Rent Now

Koenigsegg

90L Manual 2 People

$99.00/day Rent Now

Nissan GT-R

80L Manual 2 People

$80.00/day ~~$100.00~~ Rent Now

Nissan GT-R

80L Manual 2 People

$80.00/day ~~$100.00~~ Rent Now

Recomendation Car

Tesla Model 3

100kWh Manual 5 seats

Mercedes-Benz C-Class

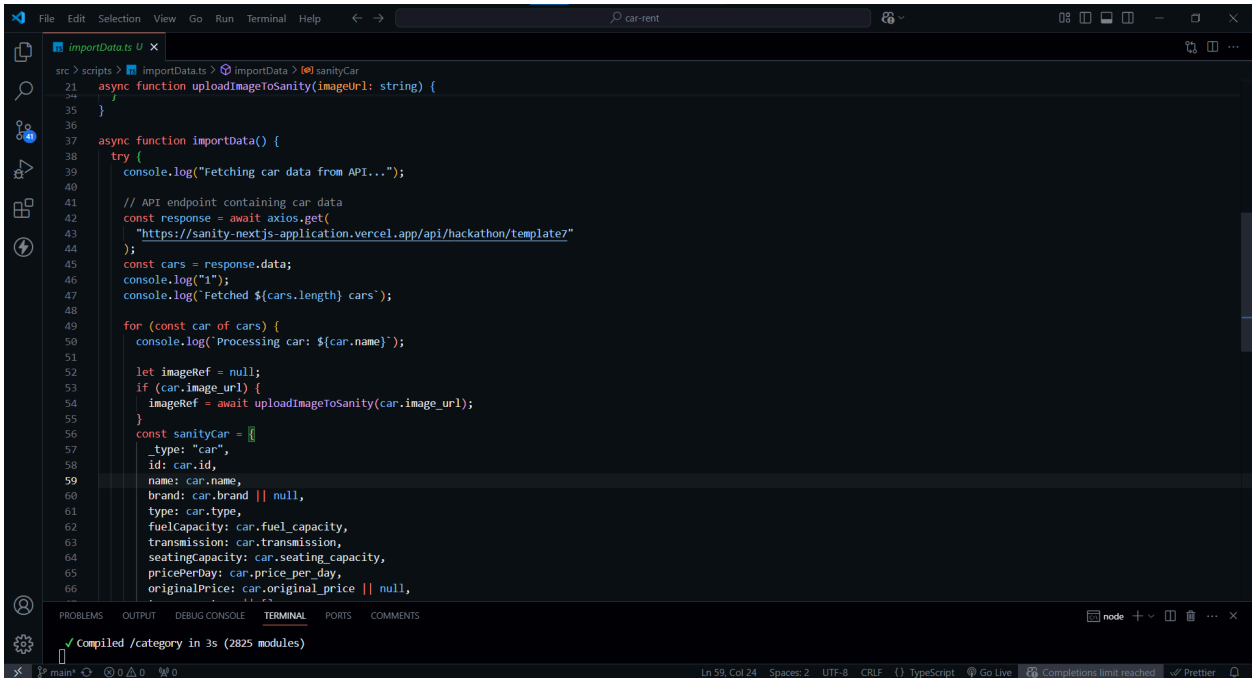
65L Manual 5 seats

CR-V

80L Manual 6 People

Ford Mustang

60L Manual 4 seats



```
async function uploadImageToSanity(imageUrl: string) {
  ...
}

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("1");
    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",
        id: car.id,
        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,
```

```
✓ Compiled /category in 3s (2825 modules)
```