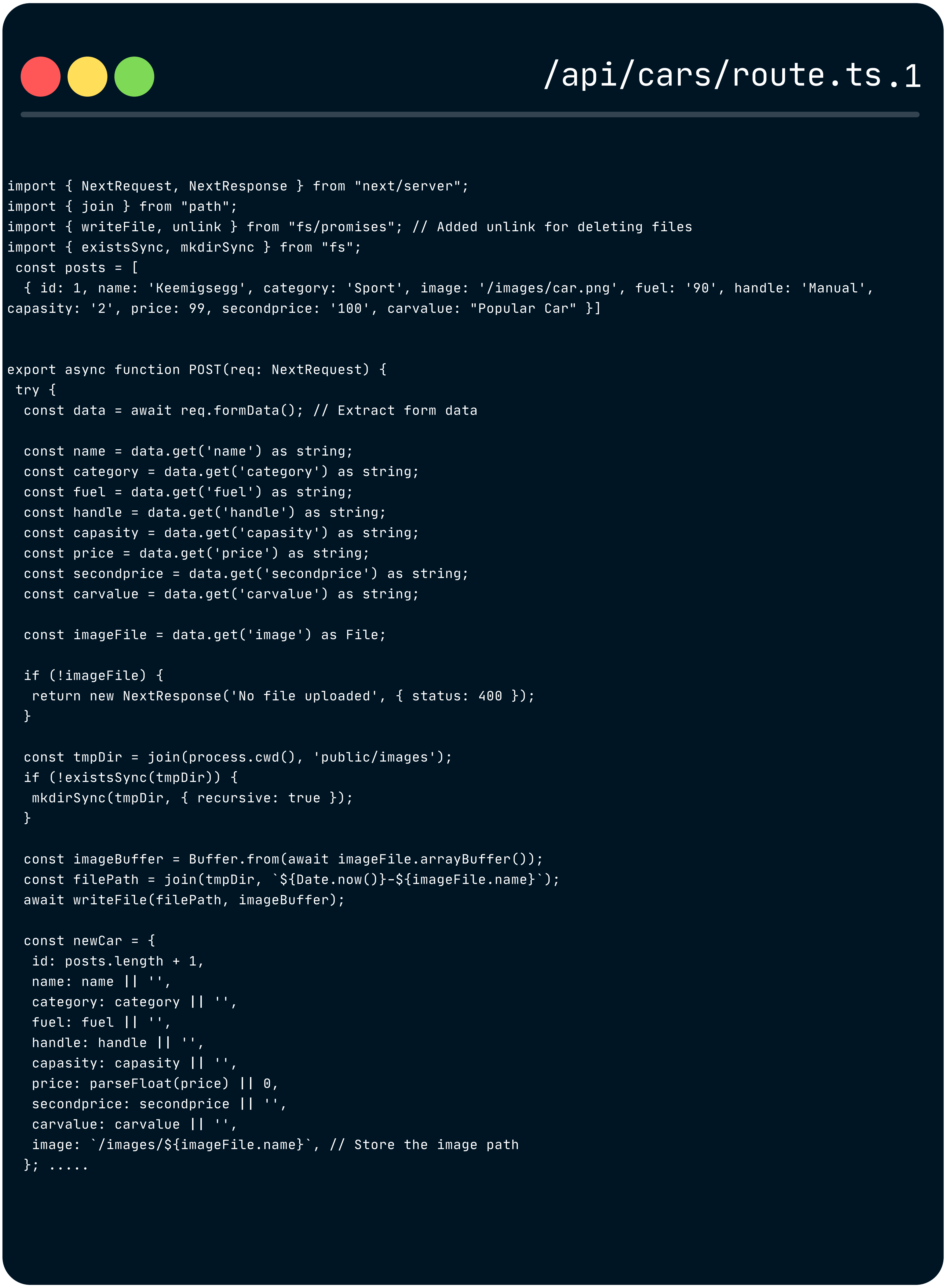
Abdul Manan Wighio

API Integration for MORENT Car Rental Web

- I carefully read the provided API documentation to understand the available endpoint (/products).
- I identified the structure of the data returned by the API, including field names and data types.
- I used Postman to test the API endpoint and ensure the data was returned correctly. I created the .mjs file, fetched the API data within it, and then uploaded the data to Sanity.
- There was a mistake in the provided repository for template 7, where we couldn't upload the data to sanity, i think many of you might be facing this issue, so here is the code :
- But first install these packages :

```
npm i dotenv axios
```

- Now Paste the following code in the scripts/importData.mjs:
- Make sure to create it outside the src folder.


/api/cars/route.ts.1

```
import { NextRequest, NextResponse } from "next/server";
import { join } from "path";
import { writeFile, unlink } from "fs/promises"; // Added unlink for deleting files
import { existsSync, mkdirSync } from "fs";
 const posts = [
  { id: 1, name: 'Keemigsegg', category: 'Sport', image: '/images/car.png', fuel: '90', handle: 'Manual',
capasity: '2', price: 99, secondprice: '100', carvalue: "Popular Car" }]


export async function POST(req: NextRequest) {
 try {
  const data = await req.formData(); // Extract form data

  const name = data.get('name') as string;
  const category = data.get('category') as string;
  const fuel = data.get('fuel') as string;
  const handle = data.get('handle') as string;
  const capasity = data.get('capasity') as string;
  const price = data.get('price') as string;
  const secondprice = data.get('secondprice') as string;
  const carvalue = data.get('carvalue') as string;

  const imageFile = data.get('image') as File;

  if (!imageFile) {
   return new NextResponse('No file uploaded', { status: 400 });
  }

  const tmpDir = join(process.cwd(), 'public/images');
  if (!existsSync(tmpDir)) {
   mkdirSync(tmpDir, { recursive: true });
  }

  const imageBuffer = Buffer.from(await imageFile.arrayBuffer());
  const filePath = join(tmpDir, `${Date.now()}-${imageFile.name}`);
  await writeFile(filePath, imageBuffer);

  const newCar = {
   id: posts.length + 1,
   name: name || '',
   category: category || '',
   fuel: fuel || '',
   handle: handle || '',
   capasity: capasity || '',
   price: parseFloat(price) || 0,
   secondprice: secondprice || '',
   carvalue: carvalue || '',
   image: `/images/${imageFile.name}`, // Store the image path
  }; .....
```

/addproducts.tsx .2

```
const pushCarsToSanity = async () => {
try {
 // API se car data fetch karna
 const response = await fetch('/api/cars');
 if (!response.ok) {
  throw new Error('Failed to fetch cars from API');
 }
 const cars = await response.json();

 // Loop through all cars
 for (const car of cars) {
  // Check if the car already exists in Sanity
  const existingCar = await client.fetch(`*[_type == "cars" && _id == "cars-${car.id}"]`);

  // Agar car already exists to skip the insertion
  if (existingCar.length > 0) {
   console.log(`Car with ID cars-${car.id} already exists in Sanity.`);
   continue; // Skip this car and move to the next one
  }

  let imageRef = '';

  // Agar car mein image hai to Sanity mein upload karenge
  if (car.image) {
   imageRef = await fetchAndUploadImage(car.image);
  }

  // Car document ka format Sanity ke liye
  const sanityProduct = {
   _id: `cars-${car.id}`, // Unique ID for car
   _type: 'cars', // Sanity document type
   name: car.name,
   price: car.price,
   secondprice: car.secondprice || '', // Optional field
   category: car.category,
   image: {
    _type: 'image',
    asset: {
     _type: 'reference',
     _ref: imageRef, // Reference to uploaded image
    },
   },
   handle: car.handle,
   carvalue: car.carvalue,
   capasity: car.capasity,
   fuel: car.fuel,
  };

  // Sanity mein car document create karna
  await client.createOrReplace(sanityProduct);
  console.log(`✅ Imported car: ${sanityProduct.name}`);
 }

 console.log('✅ All cars have been imported!');
} catch (error) {
 console.error('❌ Error pushing cars to Sanity:', error);
}
};
;
```

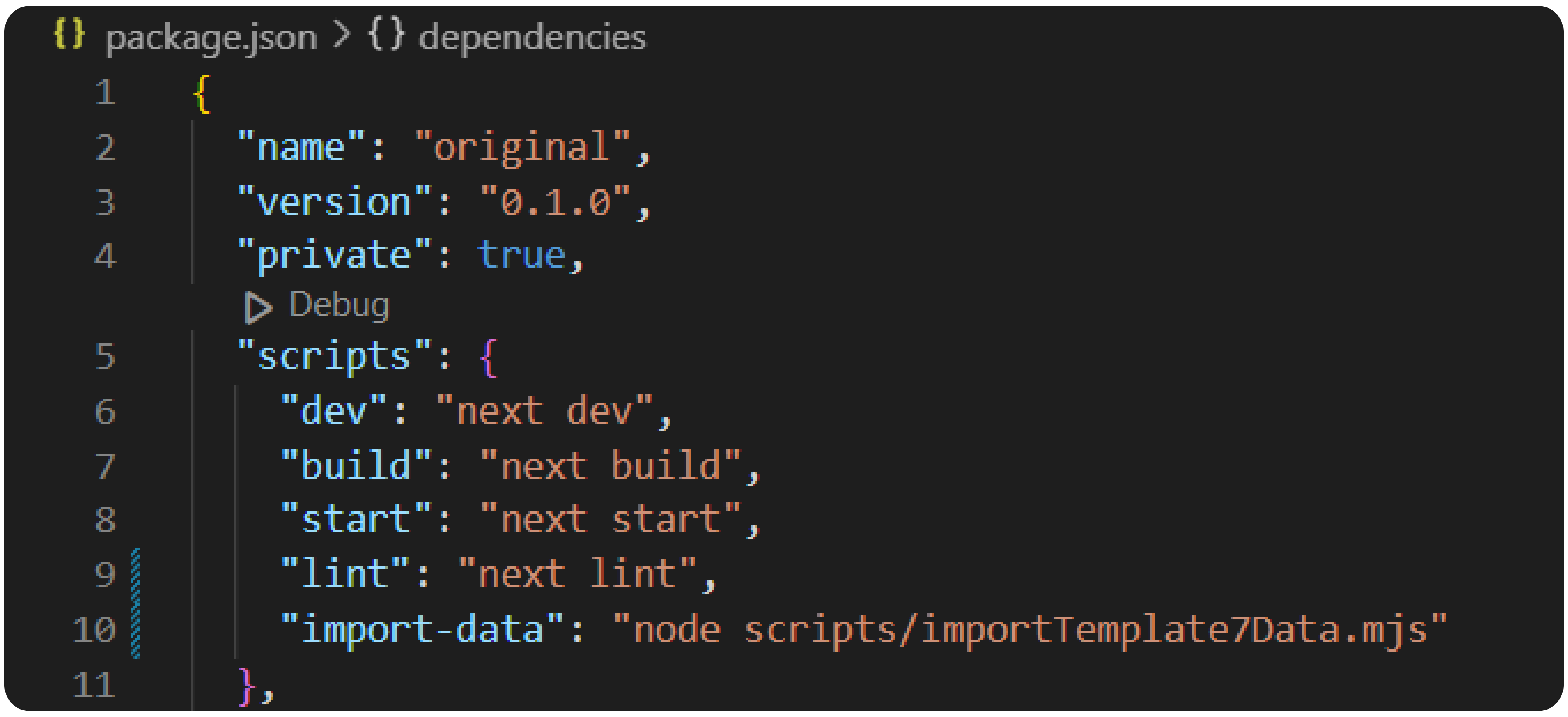
- Make sure to have sanity token, project-id, and datasets in the env files:
- create the file in src/sanity/schemaTypes/car.ts:

cars.ts

```
import { defineField, defineType } from "sanity";

export const Cars = defineType({
 name: "cars",
 title: "Cars",
 type: "document",
 fields: [{
   name: "name",
   title: "Name",
   type: "string",
   description: "Enter car name",
   },{
   name: "slug",
   title: "Slug",
   type: "slug",
   description: "Car Serial no.",
   options: {
   source: "name", // Slug will be generated based on the 'name' field
   maxLength: 96, // Max length for slug
   },
   },{
   name: "category",
   title: "Category",
   type: "string",
   description: "Car category",
   },{
   name: "image",
   title: "Image",
   type: "image",
   description: "Select image",
   }, ...
```

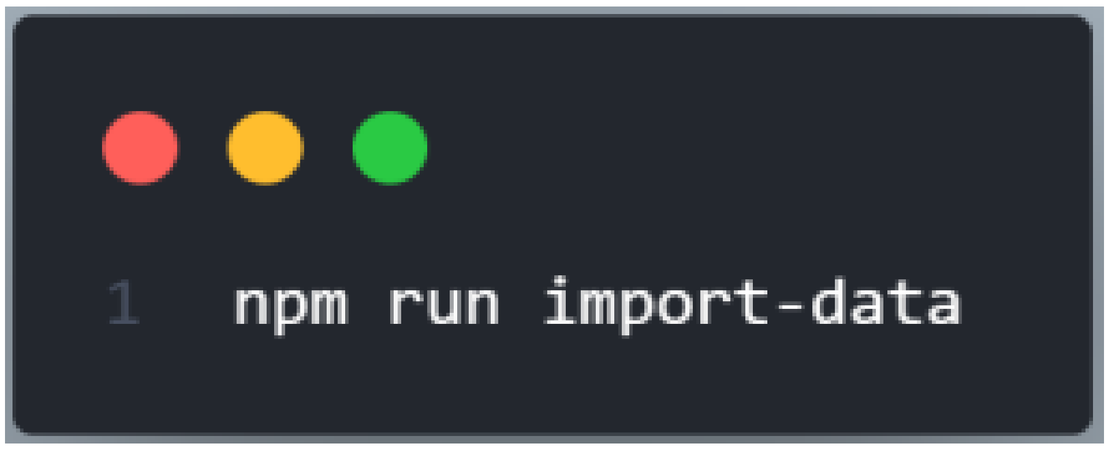
- Now update the package.json file :



package.json > {} dependencies

```
{
  "name": "original",
  "version": "0.1.0",
  "private": true,
  ▷ Debug
  "scripts": {
    "dev": "next dev",
    "build": "next build",
    "start": "next start",
    "lint": "next lint",
    "import-data": "node scripts/importTemplate7Data.mjs"
  },
```

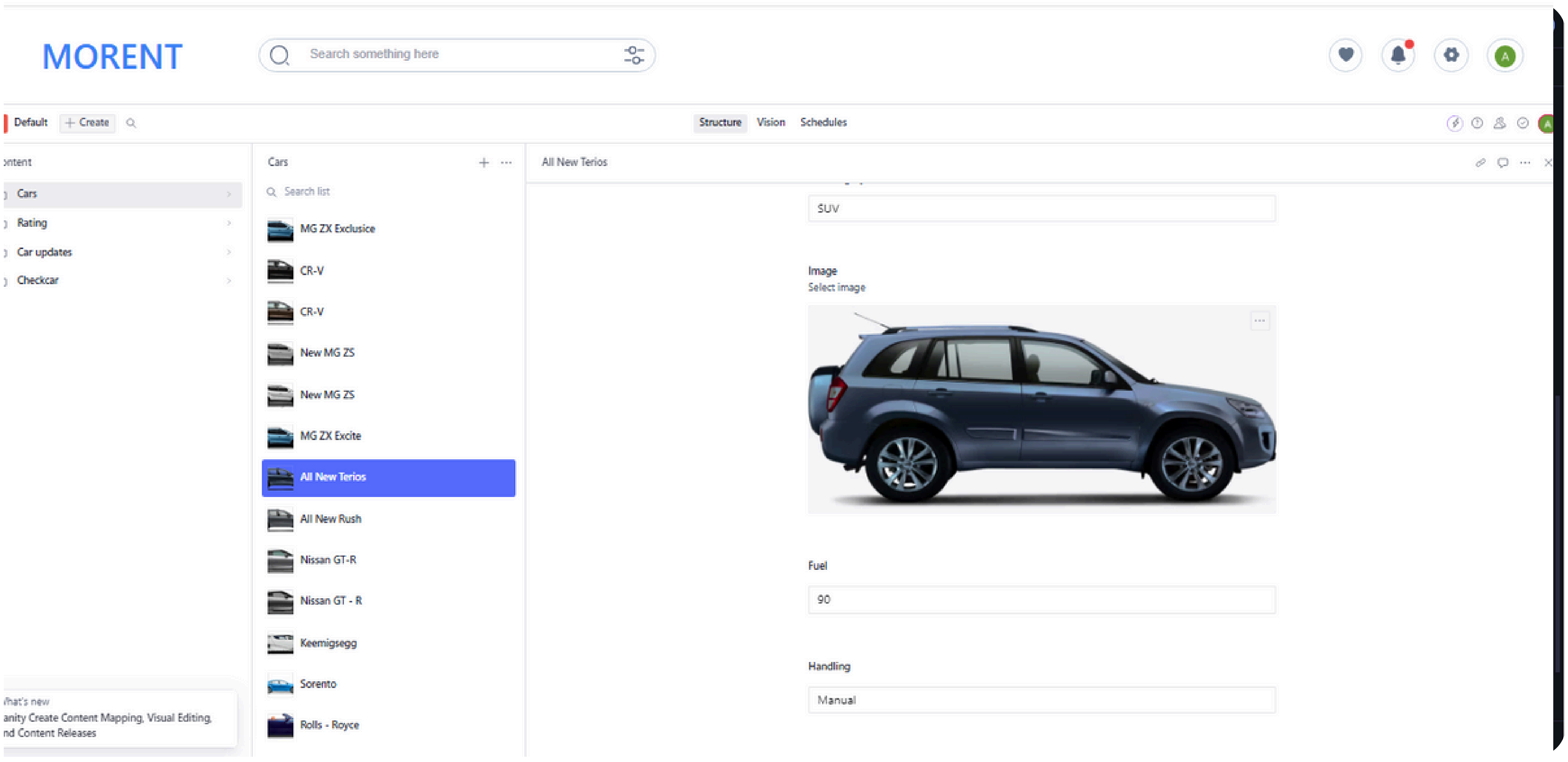
- Now run the following command:



```
npm run import-data
```

- Now the data should be inserted in the sanity.
- In this project, I successfully integrated the provided API into my Next.js frontend and migrated data into Sanity CMS. I adjusted the schema to match the API data structure and ensured the data was accurately displayed in the frontend. This exercise helped me gain practical experience in API integration, data migration, and schema validation, which are essential skills for building scalable marketplaces.

Result

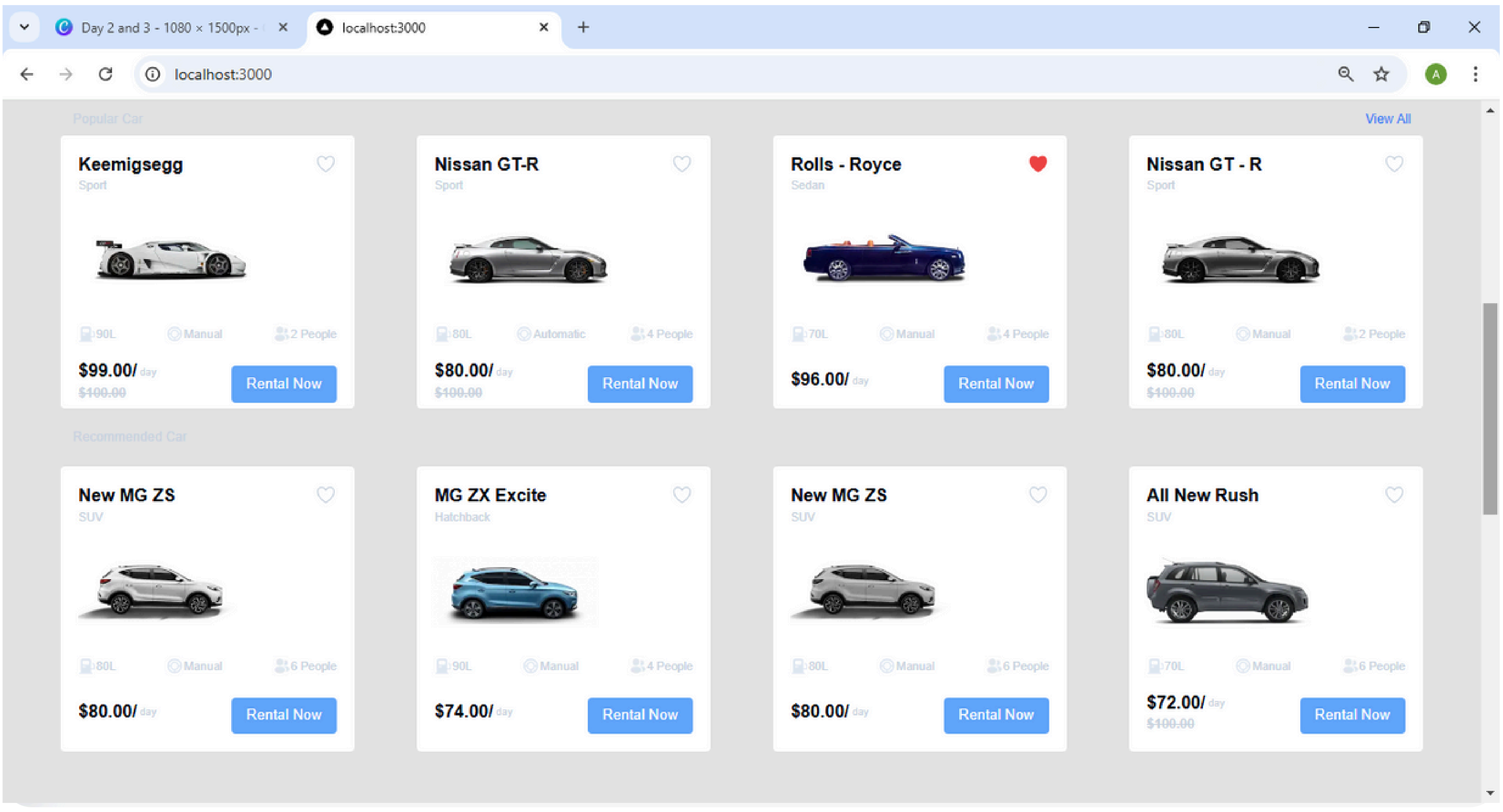


SUV

Image
Select image

Fuel
90

Handling
Manual



Keemigsegg
Sport
90L Manual 2 People
$99.00/ day
~~$100.00~~

Nissan GT-R
Sport
80L Automatic 4 People
$80.00/ day
~~$100.00~~

Rolls - Royce
Sedan
70L Manual 4 People
$96.00/ day

Nissan GT - R
Sport
80L Manual 2 People
$80.00/ day
~~$100.00~~

New MG ZS
SUV
80L Manual 6 People
$80.00/ day

MG ZX Excite
Hatchback
90L Manual 4 People
$74.00/ day

New MG ZS
SUV
80L Manual 6 People
$80.00/ day

All New Rush
SUV
70L Manual 6 People
$72.00/ day
~~$100.00~~

Tasks	Status
-API Understanding	✓
-Schema Validation	✓
-Data Migration	✓
-API Integration in Next.js	✓
-Submission Preparation	✓