Name : Shazia Bashir

Roll # : 00379887

API Integration Report - [Rental Car Website]

Reviewed API Documentation:

- I carefully read the provided API documentation for my assigned template to understand the available endpoint (/cars).
- I identified the structure of the data returned by the API, including field names and data types.

Set Up API Calls:

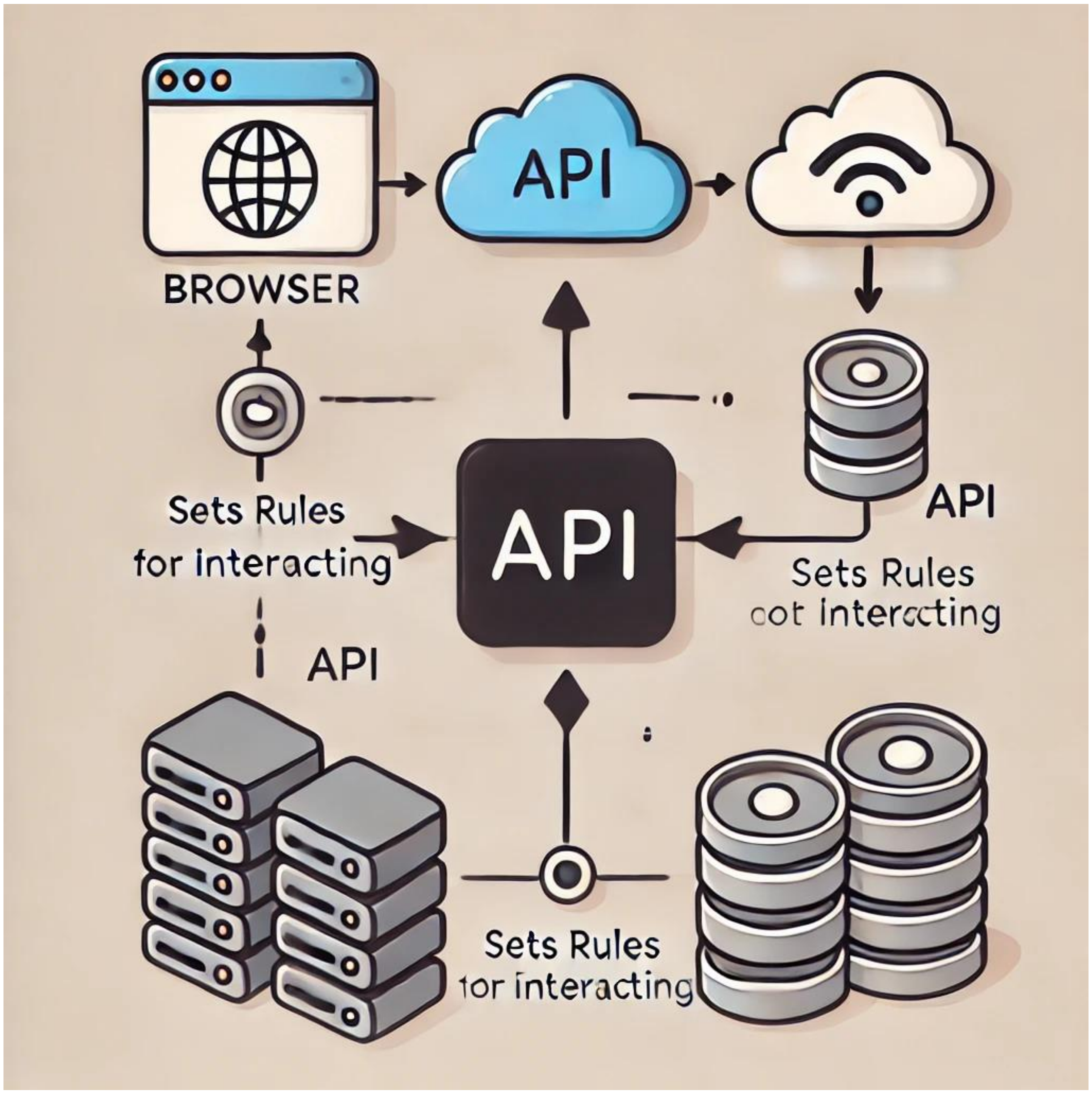
- I used Thunder client to test the API endpoint and ensure the data was being returned correctly. I created utility functions in my Next.js project to fetch data from the API.
- I used fetch to make GET requests to the API endpoints and stored the responses in variables. I logged the API responses in the console to verify the data structure.

Compared API Data with Sanity Schema:

- I reviewed the API data structure and compared it with the existing schema in Sanity CMS. I identified mismatches in field names and data types.
- I updated the Sanity schema to match the API data structure. For example:
- API Field: car_title → Sanity Field: name

- API Field: price → Sanity Field: price (with proper data type)

I added new fields in Sanity CMS to accommodate additional data from the API , because the API is not enough to complete my website products and their details .



To migrate data from the API to Sanity CMS,

I followed these steps:

- I decided to use the provided API to fetch data and write a script to import it into Sanity CMS. o I created a script Folder and then i created a migration (.mjs) file to fetch data from the API and transform it into the format required by Sanity CMS.

- I used the Sanity client library to upload the data to the CMS. I ran the migration script to import product data, categories, and other relevant information into Sanity CMS.
- I verified the imported data by checking the Sanity dashboard and ensuring all fields were correctly populated.

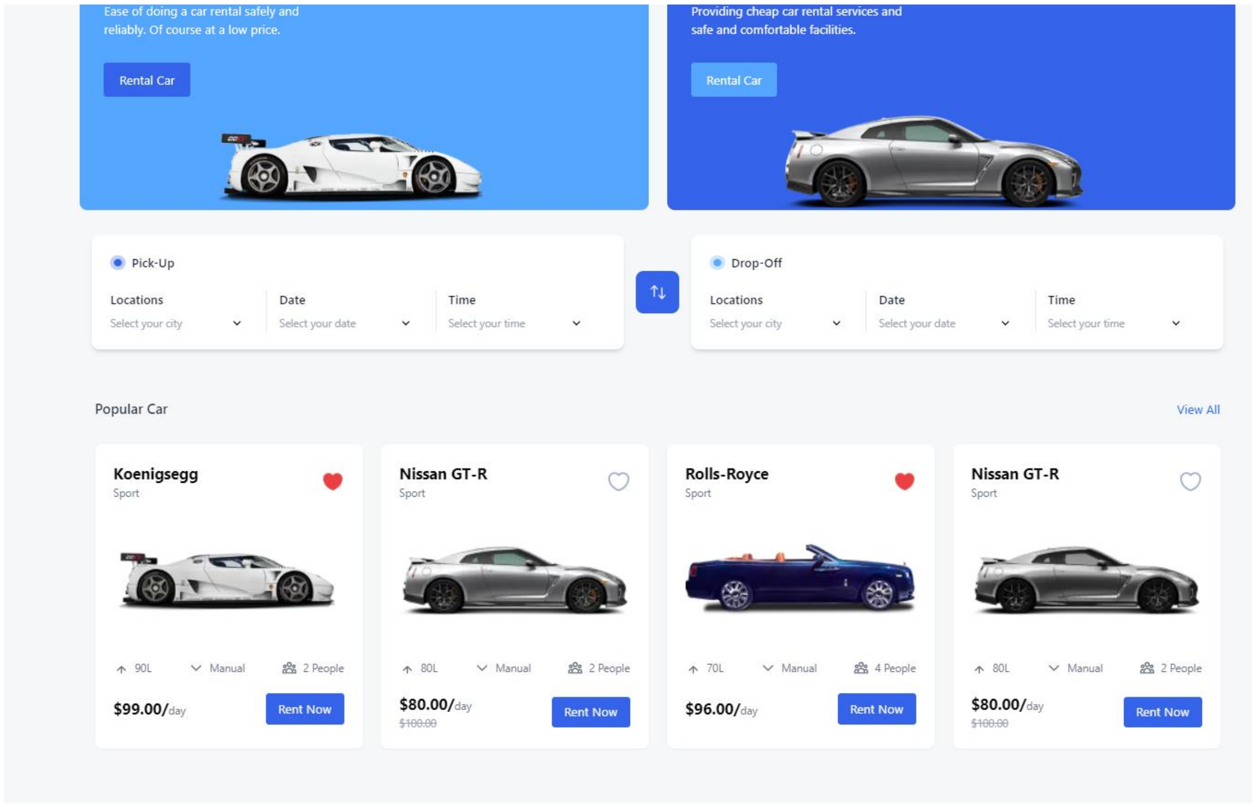
In this project, I successfully integrated the provided API into my Next.js frontend and migrated data into Sanity CMS. I adjusted the schema to match the API data structure and ensured the data was accurately displayed in the frontend. This exercise helped me gain practical experience in API integration,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

Api understanding ✅

SCHEMA Validation ✅

Data Migrated ✅

Api integrated in Next.Js ✅



Ease of doing a car rental safely and reliably. Of course at a low price.

Rental Car

Providing cheap car rental services and safe and comfortable facilities.

Rental Car

Pick-Up

Locations	Date	Time
Select your city	Select your date	Select your time

Drop-Off

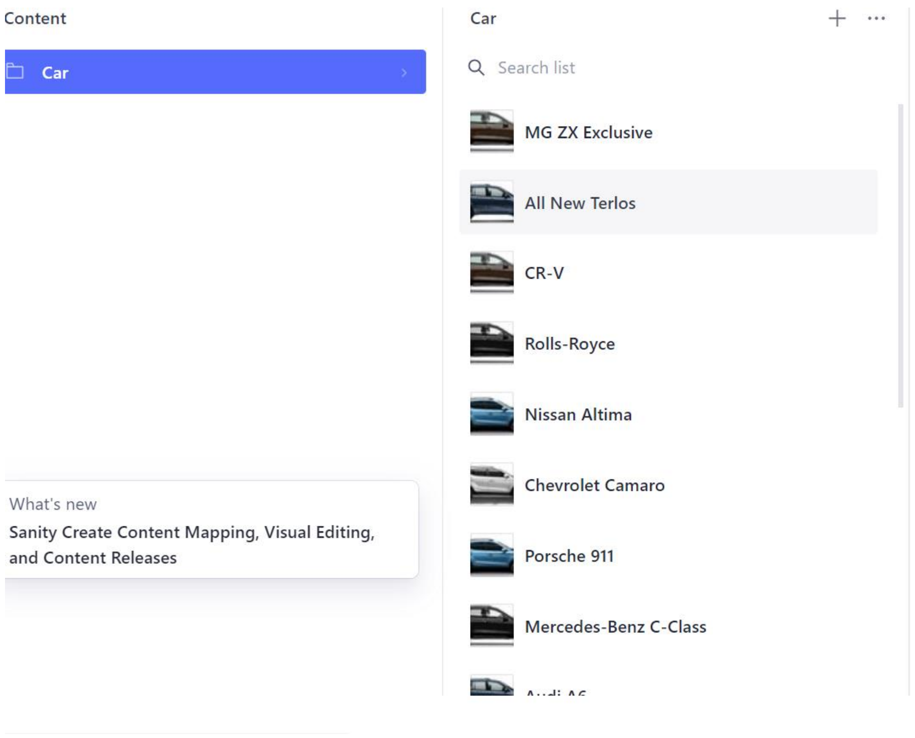
Locations	Date	Time
Select your city	Select your date	Select your time

Popular Car

View All

Car	Type	Fuel	Transmission	Capacity	Price	Old price
Koenigsegg	Sport	90L	Manual	2 People	$99.00/day	
Nissan GT-R	Sport	80L	Manual	2 People	$80.00/day	~~$100.00~~
Rolls-Royce	Sport	70L	Manual	4 People	$96.00/day	
Nissan GT-R	Sport	80L	Manual	2 People	$80.00/day	~~$100.00~~

Rent Now



Content

Car

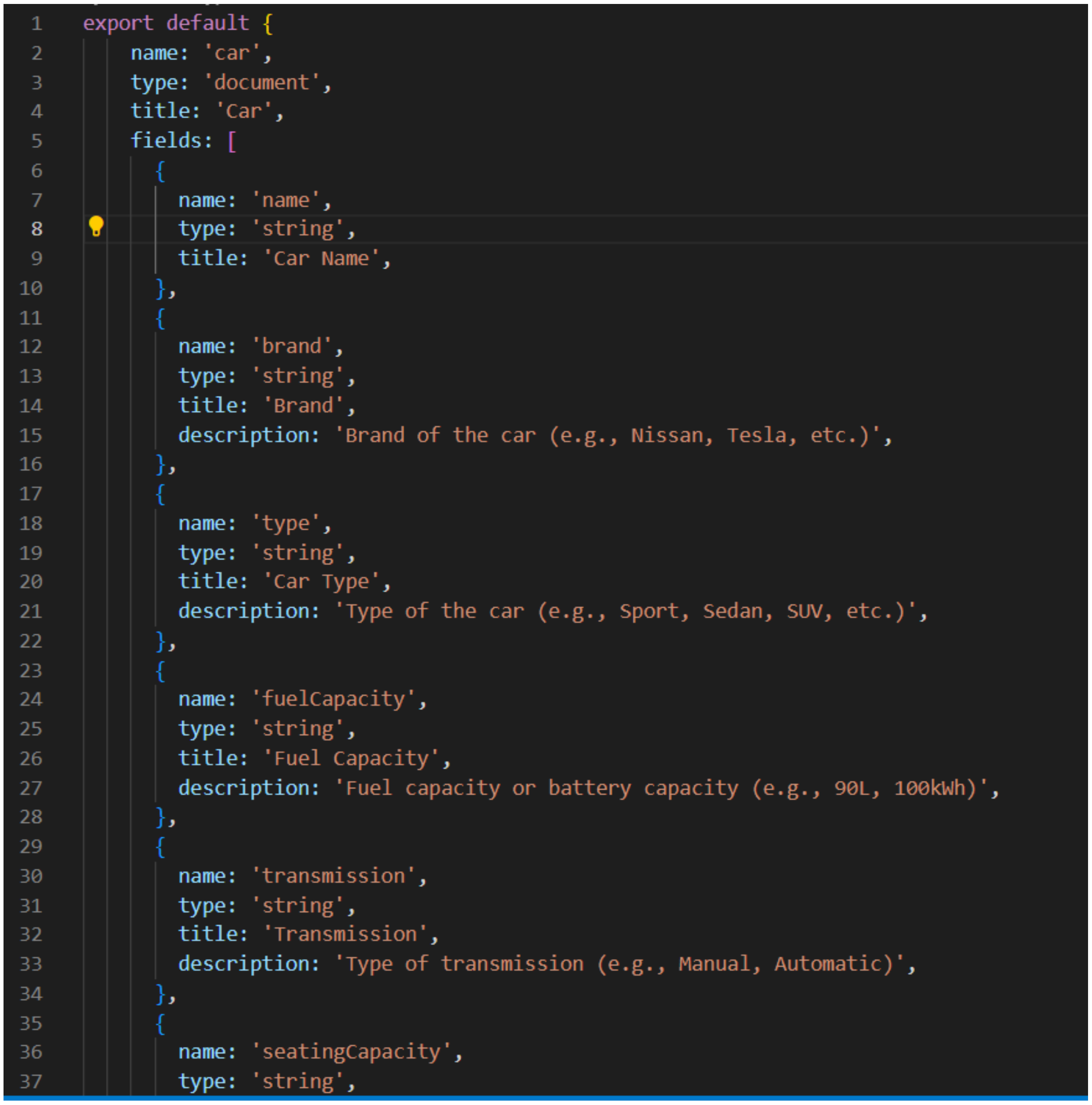
What's new

Sanity Create Content Mapping, Visual Editing, and Content Releases

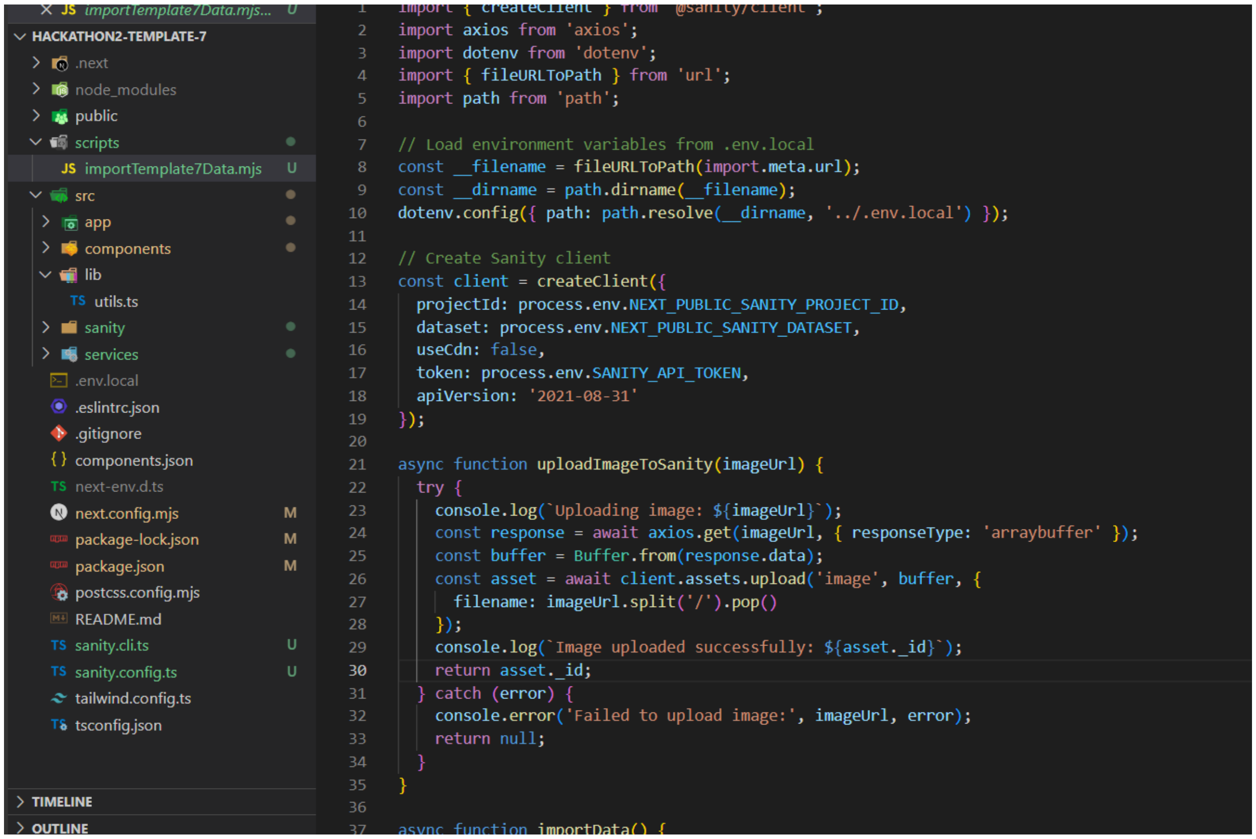
Car

Search list

- MG ZX Exclusive
- All New Terlos
- CR-V
- Rolls-Royce
- Nissan Altima
- Chevrolet Camaro
- Porsche 911
- Mercedes-Benz C-Class



```
export default {
    name: 'car',
    type: 'document',
    title: 'Car',
    fields: [
      {
        name: 'name',
        type: 'string',
        title: 'Car 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',
```



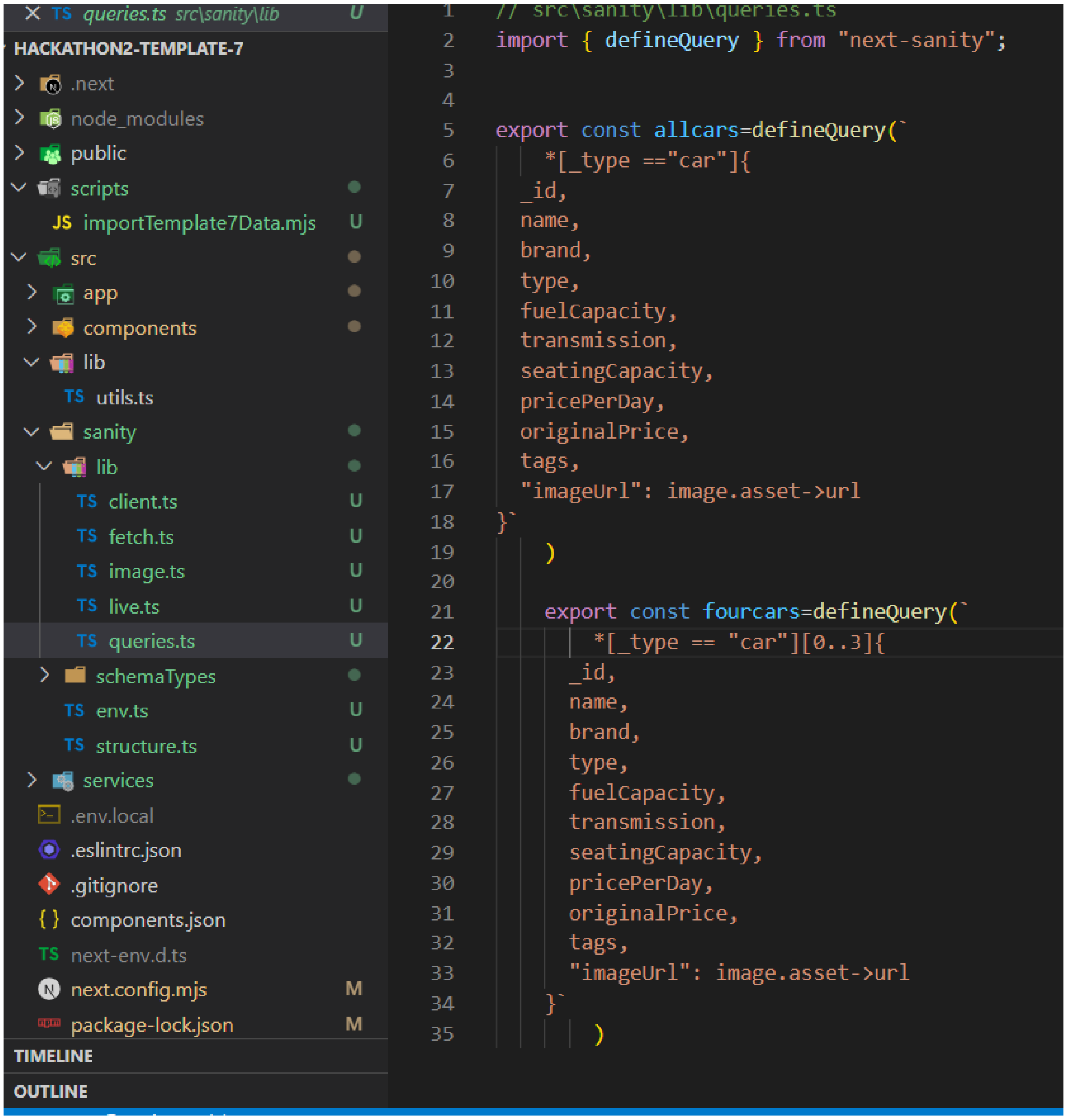
```
import { createClient } from '@sanity/client';
import axios from 'axios';
import dotenv from 'dotenv';
import { fileURLToPath } from 'url';
import path from 'path';

// 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() {
```



```
// src\sanity\lib\queries.ts
import { defineQuery } from "next-sanity";


export const allcars=defineQuery(`
    *[_type =="car"]{
  _id,
  name,
  brand,
  type,
  fuelCapacity,
  transmission,
  seatingCapacity,
  pricePerDay,
  originalPrice,
  tags,
  "imageUrl": image.asset->url
}`
    )

    export const fourcars=defineQuery(`
        *[_type == "car"][0..3]{
      _id,
      name,
      brand,
      type,
      fuelCapacity,
      transmission,
      seatingCapacity,
      pricePerDay,
      originalPrice,
      tags,
      "imageUrl": image.asset->url
    }`
        )
```