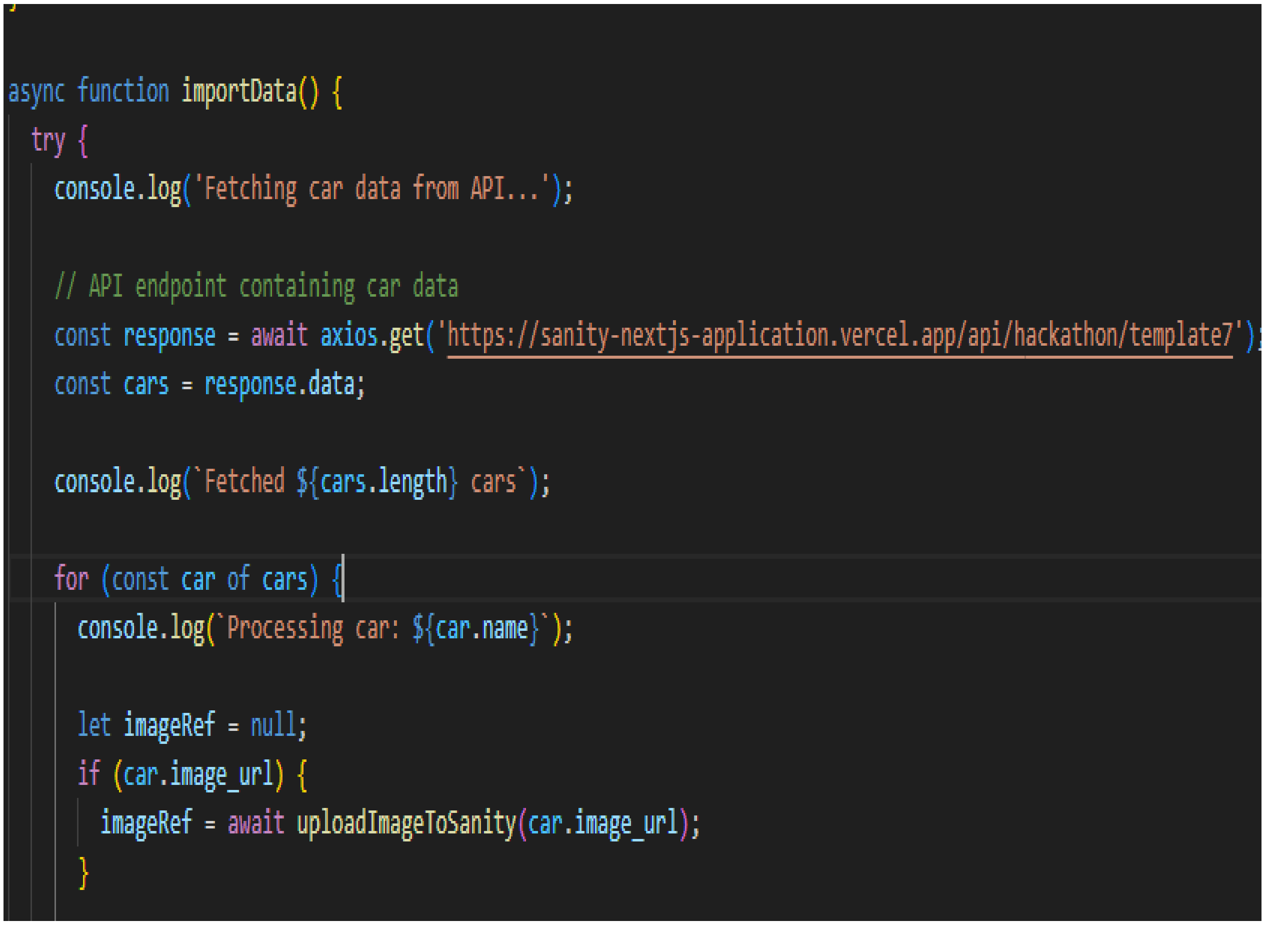
API integration process

Prepared by: Rizwan
Roll No: 00289287

1. API Selection and Setup

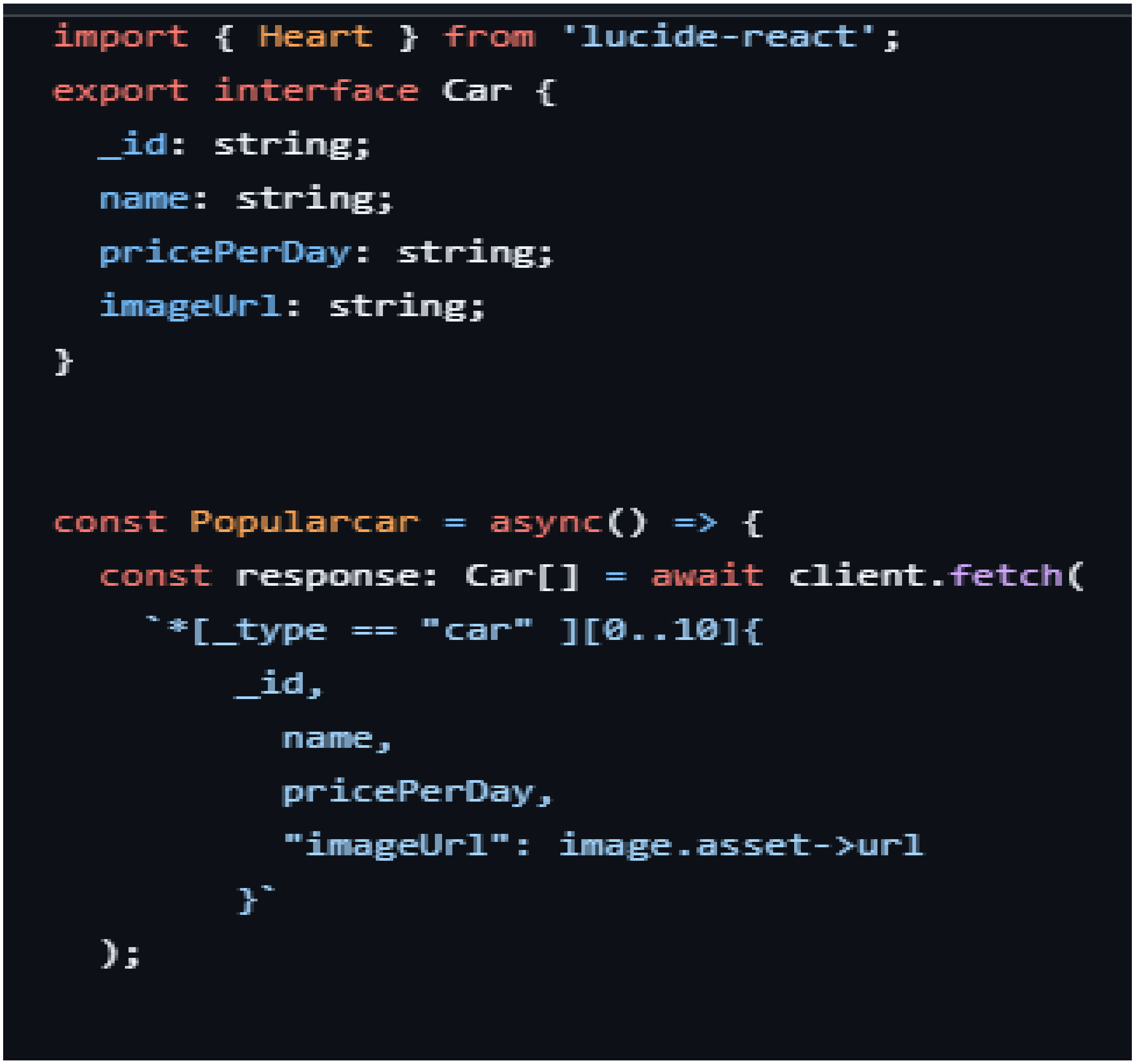
- API was provided to import product and category data for the project.
- Integrated the API using the fetch method in Next.js to streamline the data flow.
- Configured dynamic routes for fetching data based on category or slug parameters.
- Example API Endpoint:



```
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);
      }
```

2. Frontend Integration

- Implemented useEffect and useState hooks to fetch and render data dynamically.



```
import { Heart } from 'lucide-react';
export interface Car {
  _id: string;
  name: string;
  pricePerDay: string;
  imageUrl: string;
}


const Popularcar = async() => {
  const response: Car[] = await client.fetch(
    `*[_type == "car" ][0..10]{
        _id,
          name,
          pricePerDay,
          "imageUrl": image.asset->url
        }`
  );
```

```
    type: "document",
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',
      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
      }
    }
  ],
};
```

[redacted]

[redacted]

[redacted]

1. Product Schema Adjustments

- Modified the schema to align with API structure, including fields for description, and slug.
- Final Product Schema:

Migration Steps and Tools Used

1. Data Migration Steps

1. Received data through the provided API.
2. Exported data from the API in JSON format.
3. Transformed the data to match the new schema structure.
4. Imported the formatted data into Sanity CMS using their CLI tool.

2. Tools Used

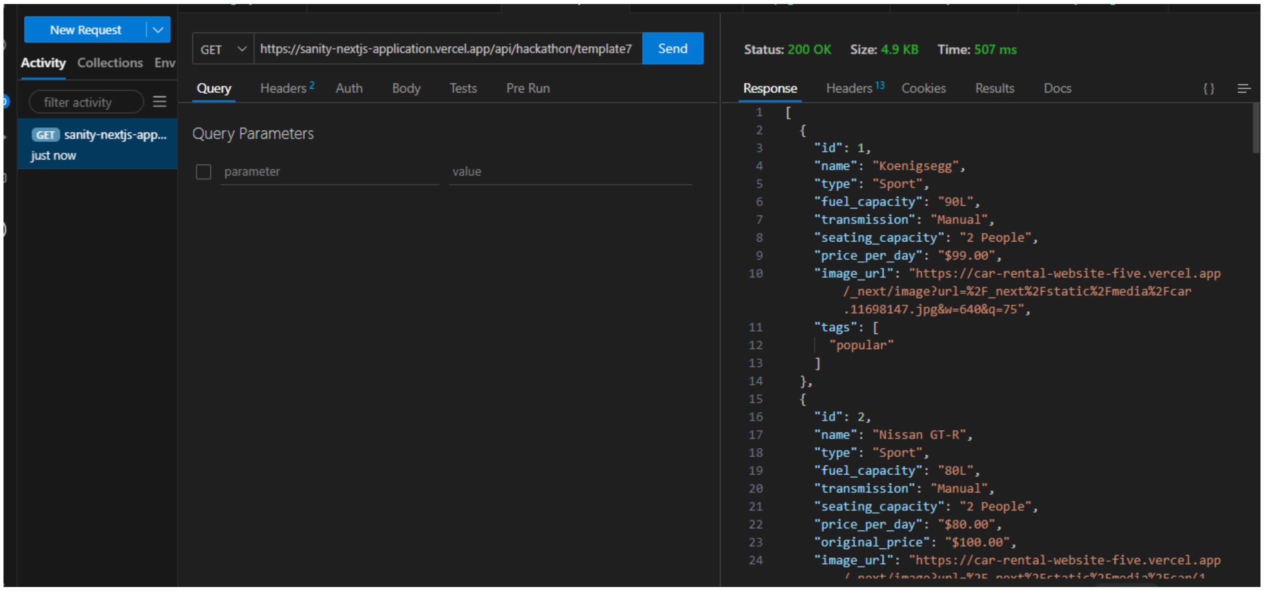
- **CLI**: For importing/exporting data.
- **ThunderClient**: To verify and test API endpoints.
- **Next.js**: For seamless frontend integration.

3. Migration Script Example

sanity dataset import exported-data.json production

---

1. API Calls



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

Query Parameters

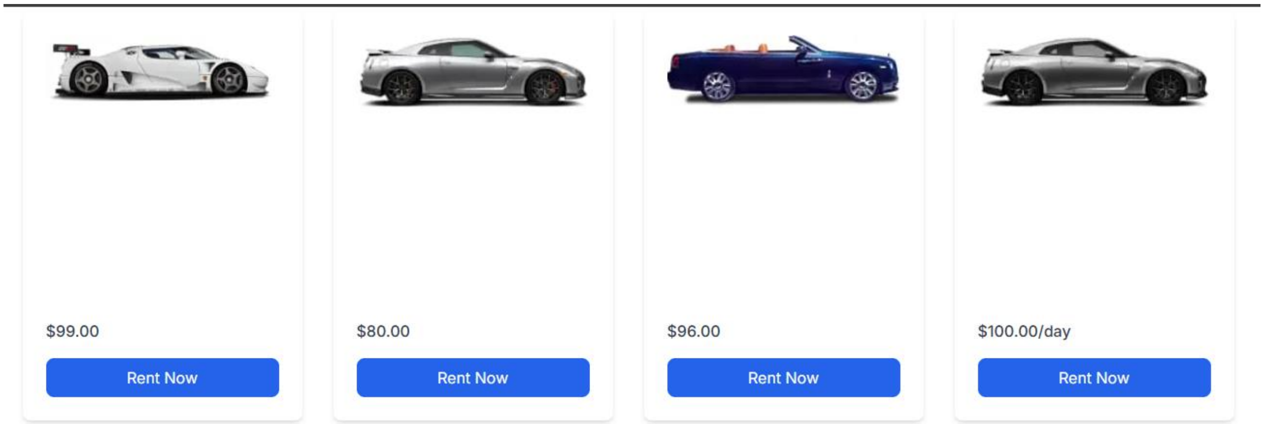
□ parameter value

Status: 200 OK Size: 4.9 KB Time: 507 ms

```
[
  {
    "id": 1,
    "name": "Koenigsegg",
    "type": "Sport",
    "fuel_capacity": "90L",
    "transmission": "Manual",
    "seating_capacity": "2 People",
    "price_per_day": "$99.00",
    "image_url": "https://car-rental-website-five.vercel.app
      /_next/image?url=%2F_next%2Fstatic%2Fmedia%2Fcar
      .11698147.jpg&w=640&q=75",
    "tags": [
      "popular"
    ]
  },
  {
    "id": 2,
    "name": "Nissan GT-R",
    "type": "Sport",
    "fuel_capacity": "80L",
    "transmission": "Manual",
    "seating_capacity": "2 People",
    "price_per_day": "$80.00",
    "original_price": "$100.00",
    "image_url": "https://car-rental-website-five.vercel.app
      /_next/image?url=%2F_next%2Fstatic%2Fmedia%2Fcar/1
```

2. Data Displayed on Frontend

Product Listing

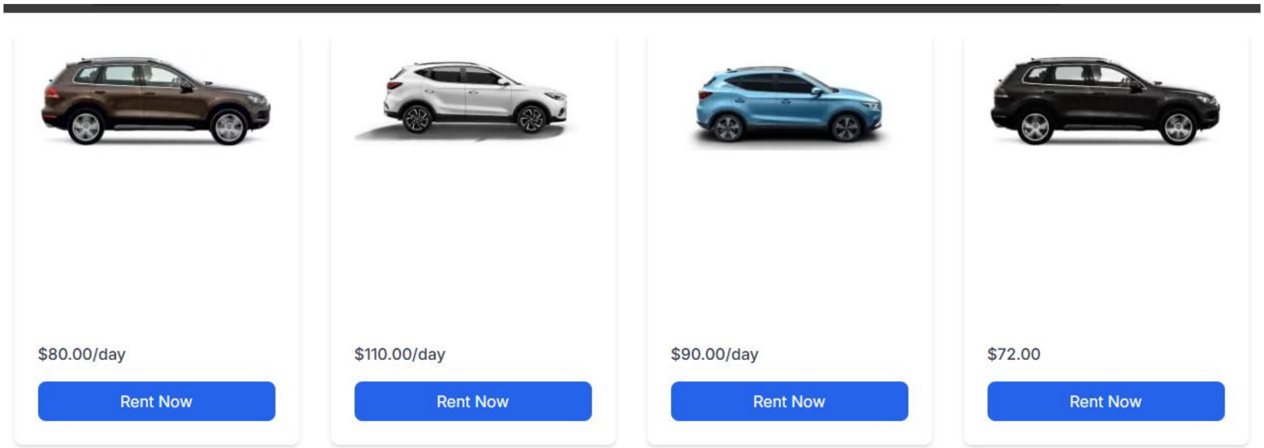


$99.00

$80.00

$96.00

$100.00/day



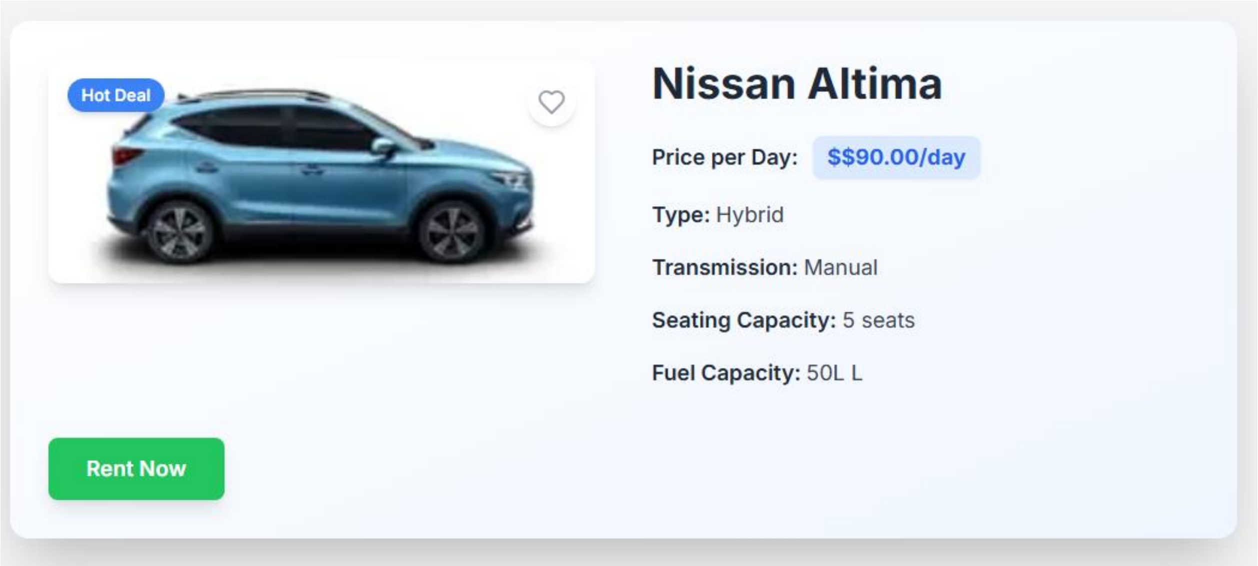
$80.00/day

$110.00/day

$90.00/day

$72.00

Product Details



Nissan Altima

Price per Day: $$90.00/day

Type: Hybrid

Transmission: Manual

Seating Capacity: 5 seats

Fuel Capacity: 50L L

3. Populated Sanity CMS Fields

Products Sanity CMS

Day 3 Checklist:

Task	Status
API Understanding	✔
Schema Validation	✔
Data Migration	✔
API Integration In Next.js	✔
Submission Preparation	✔