Day 3

API Integration Report – Car Rental Ecommerce Website

1. API Integration Process:

The process of integrating the external API involved fetching data and displaying it in our marketplace. The main goal was to connect the API with the frontend, allowing us to show car rental details on the website dynamically. Here's how it was done:

1. **Setting up the API**:
   - First, the API provided by our teacher was configured. This API endpoint contained car details like name, image, rating, rent, fuel capacity, seats, and transmission type.
2. **Fetching Data from the API**:
   - In the Next.js project, we set up API calls to fetch data from the provided API endpoint. This allowed us to gather all the car details and display them on the frontend.
3. **Displaying Data on the Frontend**:
   - The data fetched from the API was passed to the components on the frontend, ensuring that the car rental information appeared correctly for users.
4. **Sanity CMS Integration**:
   - After fetching the data, we used Sanity CMS to manage and store the car details.
   - We linked the API data with the appropriate fields in Sanity CMS so that the data could be updated dynamically on the site.

2. Adjustments Made to Schemas:

To ensure the API data was stored properly in Sanity CMS, the schemas (structure of data) in Sanity needed to be adjusted. Here are the changes made:

1. **Car Schema Changes**:
   - We created fields for:
     - **Car Name**
     - **Car Image**

Prepared By: Shumaila Aijaz
Roll Number: 00219697

- **Rating**
- **Fuel Capacity**
- **Seats**
- **Transmission Type**

  - These fields allowed us to store the car details that were fetched from the API in a structured way in Sanity CMS.

2. **Mapping API Data to CMS**:
   - The data from the API (such as the car name, image, rating, etc.) was mapped to the corresponding fields in Sanity. This step ensured that the data displayed correctly on the frontend.

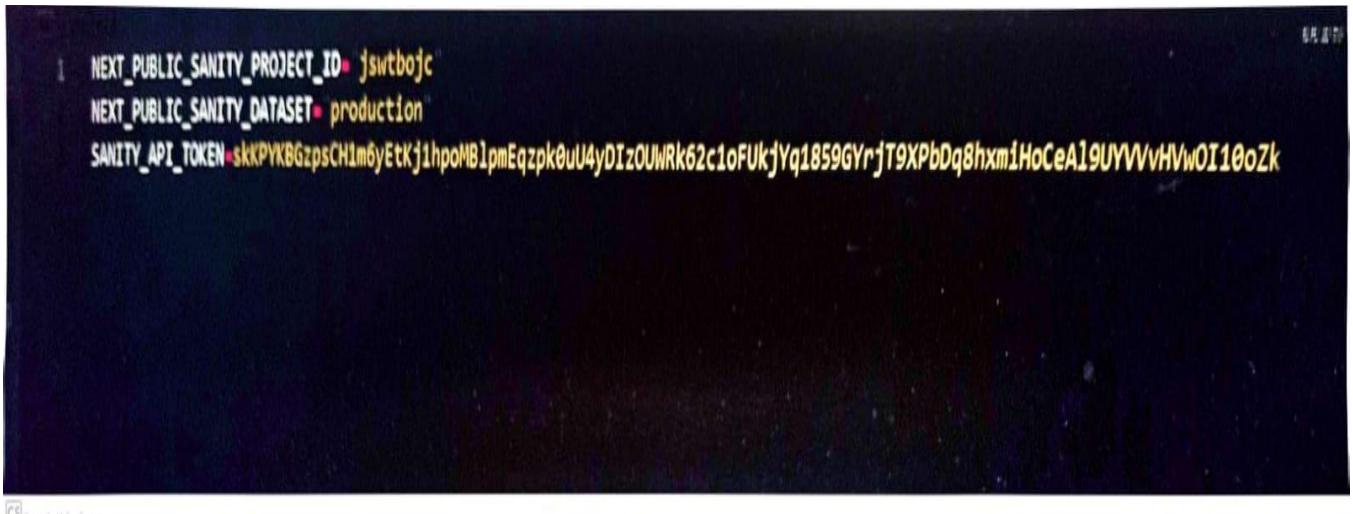
Prepared By: Shumaila Aijaz
Roll Number: 00219697

3. Migration Steps and Tools Used:

The migration process was about moving the car rental data from the API to Sanity CMS, so that it could be displayed on the website. Here's how the migration was carried out:

1. **Setting up Sanity CMS**:
   - Sanity was set up with the necessary schemas to store car details. We configured it to work with the frontend and API.
2. **Data Migration Process**:
   - We used the tools provided by the instructor to migrate data from the API to Sanity. The car details were fetched from the API and pushed into the Sanity CMS fields for storage.
3. **Tools Used**:
   - **Sanity CMS**: Used for content management and storing the car details.
   - **Next.js**: Used for fetching and displaying the data on the frontend.
4. **Automation of Data Migration**:
   The migration was automated so that every time new data was fetched, it was directly populated into Sanity CMS, ensuring the frontend always displayed the latest car rental information.

Environment Variables



Prepared By: Shumaila Aijaz
Roll Number: 00219697

hackathon-2 > src > sanity > schemaTypes > TS product.ts > default > fields

```ts
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)',
      },
```


Prepared By: Shumaila Aijaz
Roll Number: 00219697

```ts
import { type SchemaTypeDefinition } from 'sanity'
import car from './car'
import userOrder from './userOrder'

export const schema: { types: SchemaTypeDefinition[] } = {
  types: [car,userOrder],
}
```

```
hackathon-2 > scripts > importSanityData.mjs > client > apiVersion
```

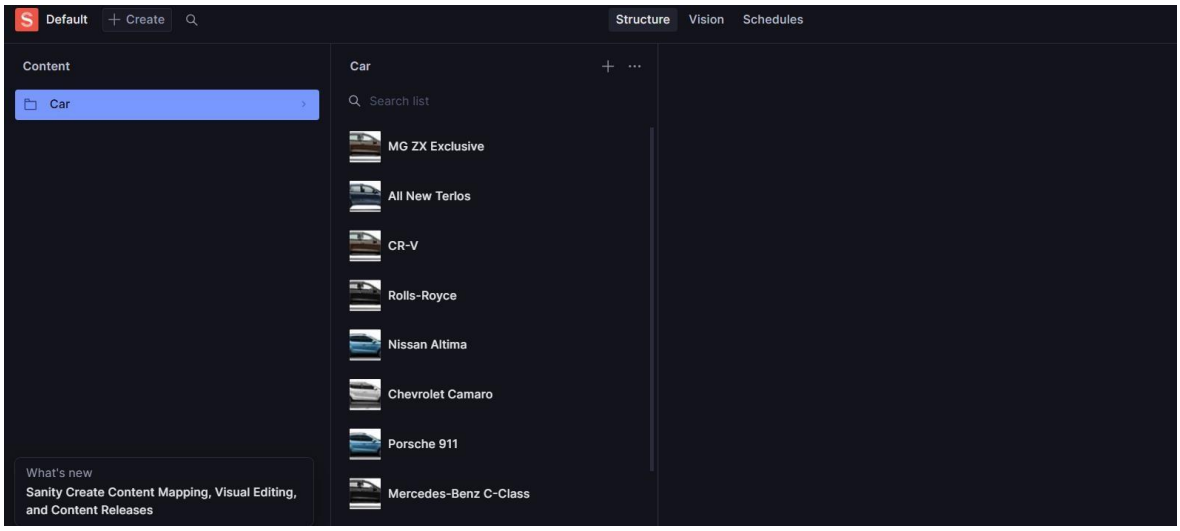
```js
import { createClient } from '@sanity/client';
import axios from 'axios';
import dotenv from 'dotenv';
import { fileURLToPath } from 'url';
import path from 'path';

// 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;
  }
}
```


Prepared By: Shumaila Aijaz
Roll Number: 00219697



Prepared By: Shumaila Aijaz
Roll Number: 00219697

Prepared By: Shumaila Aijaz
Roll Number: 00219697