Day 3

API INTEGRATION AND DATA MIGRATION

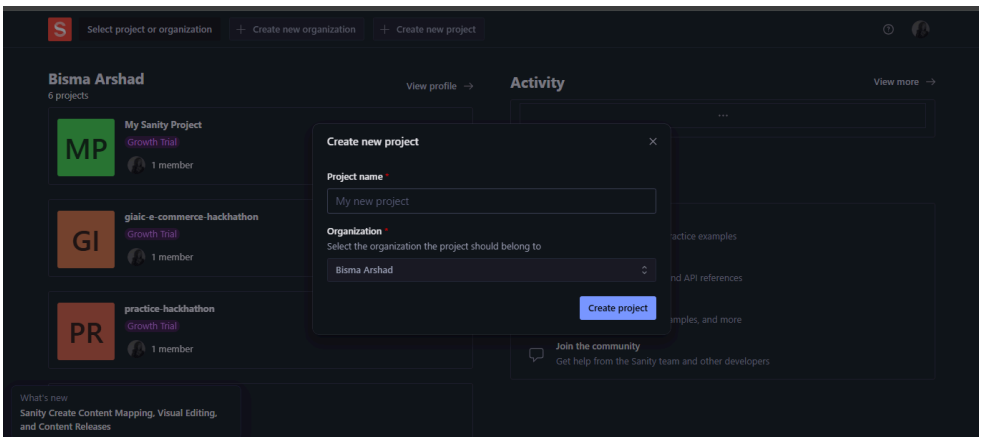
API Integration Report - [Bandage]

Sanity API Integration

I implemented Sanity API integration in my project. Here's the process:

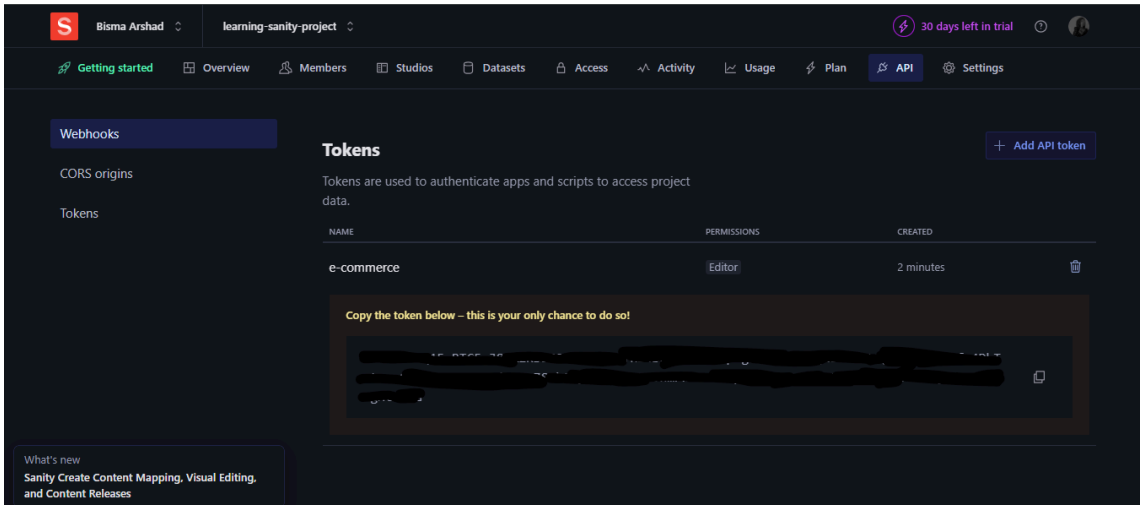
Step 1: Create a Sanity Project

I created a new project in Sanity and defined my data model. I created a schema and documents.



Step 2: Generate an API Token

I generated an API token in Sanity, which is necessary for connecting my application to Sanity's API. I generated the token and stored it in my application.



Step 3: Define API Endpoints

I defined API endpoints in Sanity's API, such as /api/vercel/docs, /api/vercel/assets, or /api/vercel/query. I defined the endpoints and used them in my application.

Step 4: Send API Requests

I sent API requests from my application, including the API token, API endpoint, and request body. I sent the request and received a response.

Step 5: Handle API Responses

I handled the API response, parsing the data and using it in my application. I handled the response and displayed the data in my application.

Step 6: Test API Integration

I tested the API integration, verifying the API request and response. I tested and ensured that the API integration works correctly.

Schema Adjustments

- Schema Identification
- Field Addition/Removal
- Field Modification
- Relationship Establishment
- Validation Rules Update
- Schema Testing

Schema Identification

I identified the schema that required adjustments. I analyzed the schema and understood what adjustments were needed.

Field Addition/Removal

I added new fields or removed existing fields from the schema. I updated the schema and

added new fields as required.

Field Modification

I modified the properties of existing fields in the schema, such as data type, default value, or validation rules. I updated the schema and modified the field properties.

Relationship Establishment

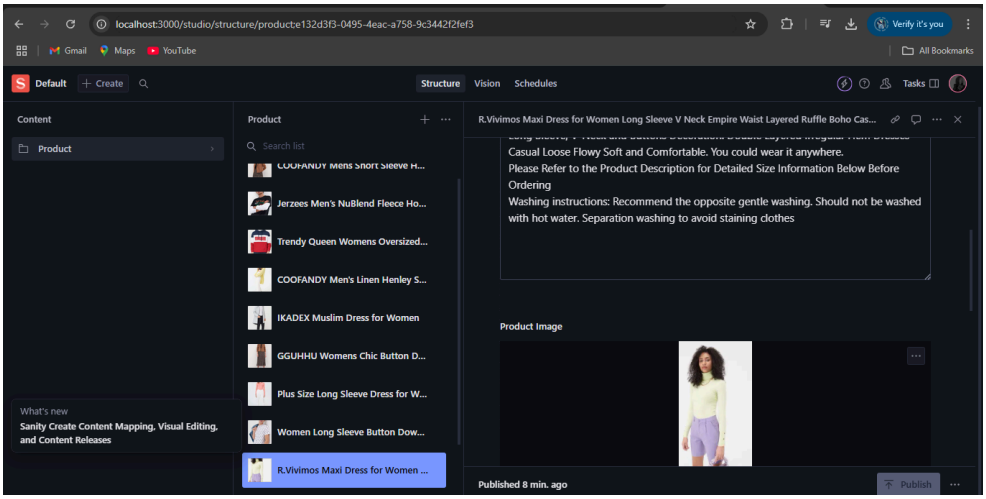
I established new relationships in the schema, such as one-to-one, one-to-many, or many-to-many. I updated the schema and established new relationships.

Validation Rules Update

I updated the validation rules in the schema, such as required fields, unique constraints, or regex patterns. I updated the schema and updated the validation rules.

Schema Testing

I tested the schema adjustments, such as data insertion, update, or deletion. I tested the schema and ensured it worked correctly.



API Integration Example

I implemented API integration in my project. Here's a JavaScript (Node.js) example:

```
// Import axios library
const axios = require('axios');

// API endpoint URL
const apiUrl = 'https://api.example.com/data';

// API request headers
const headers = {
'Authorization': 'Bearer YOUR_API_KEY',
'Content-Type': 'application/json'
};

// API request data
const data = {
'name': 'John Doe',
'email': 'john.doe@example.com'
};

// Send API request
axios.post(apiUrl, data, { headers })
.then(response => {
console.log(response.data);
})
.catch(error => {
console.error(error);
});
```

What Does This Code Do?

This code implements API integration. It sends a POST request to the API endpoint, including the API key and request data.

Data successfully displayed in the frontend.