**API Integration Report**

**Schema Definition**

As part of the project development, a schema for products was defined and categorized to allow efficient management of product data. The Category field was added to the schema to help organize products based on their types, ensuring easier filtering and categorization on the frontend.

**Category Addition**

The Category field was successfully integrated into the schema, enabling product categorization directly within the Sanity CMS. This step was essential for later stages of the project, where category-based filtering was a critical requirement for the product display.

**API Integration for Category-Based Listing**

On Day 3, the project progressed with API integration for fetching product data dynamically. Initially, mock API data was used to simulate the process. This mock data was designed to replicate real-world data that would be fetched from an actual API endpoint.

* **Mock API Usage**: Mock API data was created to simulate product information. This allowed for testing the functionality of the product listing and categorization features before connecting to the live data source.
* **Sanity Data Import**: Once the API integration was tested, the mock data was imported into Sanity CMS. This step ensured that the Sanity platform was configured to store and manage the necessary product data, including the newly added category field.
* **Frontend Implementation**: After successfully importing the data into Sanity, the frontend was updated to dynamically display products based on their categories. Products could now be filtered and shown according to their respective categories, enhancing the overall user experience. This feature was implemented by fetching the data through API calls and filtering the products on the frontend based on the selected category.