# DAY 3
## API AND DATA MIGRATION

```
+--------------------+      +-----------------+      +-------------------+      +---------------+
|  External API      | <---> | Next.js Frontend | <---> | Sanity CMS        | <---> | Frontend Render |
|  |(Fetch API data)  |   | (Store & Retrieve data) | (Display products, categories)|
+--------------------+      +-----------------+      +-------------------+      +---------------+
          |                      |                      |                      |
          v                      v                      v                      v
   Fetch Product Data      Use API Utility Functions    Schema Validation      Render Data
   (e.g., Product List)
    (e.g., /products)          & Transformation          (Field mapping)          on Next.js page
          |                                                  |
          v                                                  v
   Transform Data (Mapping to Schema)    Validate Data & Ensure Compatibility    Display
   Data in Components (HTML)
          |                                                  |
          v                                                  v
   Migrate Data to Sanity CMS: Manually or via API
```
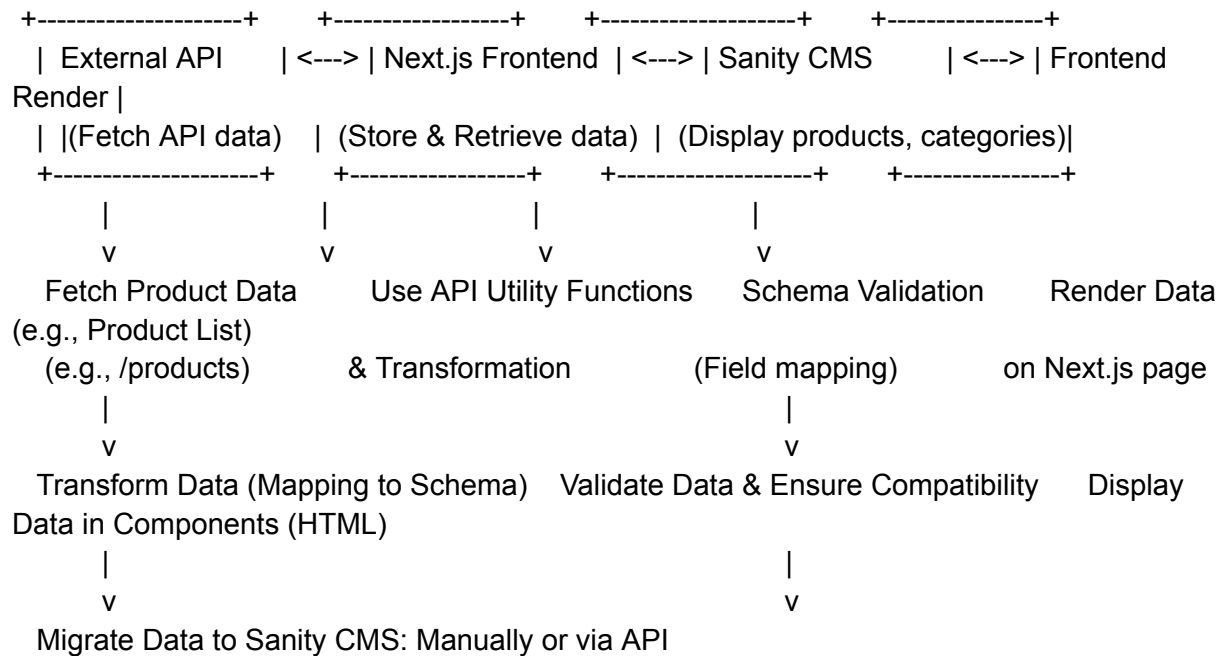
Steps Explained in the Diagram:
External API (Data Source):

This represents the external platforms. The Next.js frontend will make API calls to fetch product data, categories, etc.
Next.js Frontend:

The frontend will integrate the fetched data using utility functions.
You may need to map fields from the API to your Sanity schema (e.g., map product_name to name).
After transforming the data, you can store it in Sanity.
Sanity CMS:

Sanity CMS stores the data in a structured format, such as products, categories, and any other content related to the marketplace.
You can use API methods (like create, update, or delete) to insert, modify, and retrieve content from the CMS.
Frontend Rendering:

The final step involves rendering the data from Sanity CMS in the Next.js frontend (e.g., product listings, categories).
This is where the marketplace will be displayed to the user.

Conclusion:
This diagram visually demonstrates the process from fetching data from an external API, transforming and validating it, migrating it into Sanity CMS, and then rendering the data on your frontend.

This flow will ensure that your API integration and data migration are seamless, structured, and compatible with your Next.js and Sanity CMS project.