

AI Engineer Assessment: Asset Information Extraction System

Objective

Design and implement a **minimal version** of an AI-powered system that takes a product's **model number** and **asset classification name**, searches for relevant product information online, and extracts structured data using a Large Language Model (LLM).

The system should combine **web search**, **prompt engineering**, and **AI response parsing** to automatically generate structured product metadata. Think of it as a lightweight version of a retrieval-augmented generation (RAG) pipeline specialized for asset verification.

What You'll Build

You are expected to implement a simplified system that:

1. Accepts a **JSON input** with asset information.
2. Performs a **web search** based on the input.
3. Feeds relevant context from search results to a **language model** (like Gemini or GPT).
4. Parses and returns a **structured JSON output** with validated fields.
5. Implements basic **error handling and retry logic**.

Input Format

Your system should accept a JSON objects like this:

```
{
  "model_number": "MRN85HD",
  "asset_classification_name": "Generator (Marine)",
  "manufacturer": "", // optional
  "asset_classification_guid2": "" // optional
}
```

- *model_number* and *asset_classification_name* are **required inputs**.
- Other fields are optional but may help build better search queries

Output Format

Return a final JSON like:

```
{
  "asset_classification": "Marine Generator",
  "manufacturer": "Cummins",
  "model_number": "MRN85HD",
  "product_line": "Onan",
  "summary": "The Cummins MRN85HD is a Marine Generator categorized
under....."
}
```

Fallback Mechanism

- If the model returns incomplete fields, retry up to 5 times in interval
- If still incomplete, return a fallback response with:
 - asset_classification: set to "Generator Emissions/UREA/DPF Systems"
 - model_number: original input
 - Other fields: empty

Logging

- Print logs or debug messages indicating the input received, retries (if any), and reasons for fallback if triggered