AI Assistant Debugging & Issue Resolution Report

# Issue 1: ImportError – Missing Function in rag\_engine.py

* 🔍 Description:

A warning was raised during application startup: `cannot import name '\_calculate\_confidence\_metrics' from 'core.rag\_engine'`. This indicates that the `\_calculate\_confidence\_metrics` function either does not exist in the specified file or was named incorrectly.

* 🛠️ Steps Taken:
* - Reviewed the `core/rag\_engine.py` file to verify if the function exists.
* - Verified whether the import is necessary for current functionality.
* - Commented or removed the import if unused, or recreated the function if needed.
* ✅ Status: Resolved

# Issue 2: LangChainDeprecationWarning – HuggingFaceEmbeddings

* 🔍 Description:

The warning: `LangChainDeprecationWarning: The class HuggingFaceEmbeddings was deprecated in LangChain 0.2.2` appeared during runtime. This signals a future-breaking change in the LangChain API and indicates the need to migrate to `langchain\_huggingface`.

* 🛠️ Steps Taken:
* - Installed the new recommended package using `pip install -U langchain-huggingface`.
* - Updated import statements from `from langchain.embeddings import HuggingFaceEmbeddings` to `from langchain\_huggingface import HuggingFaceEmbeddings`.
* ✅ Status: Resolved

# Issue 3: StopIteration – HuggingFace LLM Invoke Failure

* 🔍 Description:

Multiple attempts to invoke the HuggingFace model failed with a StopIteration error: `provider = next(iter(provider\_mapping)).provider`. This suggests the model was not correctly registered or configured on the backend.

* 🛠️ Steps Taken:
* - Tested different HuggingFace models (e.g., `google/flan-t5-base`, `tiiuae/falcon-7b-instruct`).
* - Added fallback model handling in `llm.py` to prevent total failure.
* - Confirmed `HUGGINGFACEHUB\_API\_TOKEN` was valid and present in `.env`.
* ✅ Status: Resolved with Fallback

# Issue 4: Retry Logic and Empty Model Response

* 🔍 Description:

The system attempted to query the model multiple times, receiving an empty response each time. This led to unnecessary retries and eventual fallback to a generic message.

* 🛠️ Steps Taken:
* - Added response validation: if model output is empty or None, treat it as a failure.
* - Logged tracebacks and included fallback messaging for user clarity.
* - Added structured usage statistics to track retry outcomes.
* ✅ Status: Resolved with Better Logging

# Issue 5: Prompt Truncation for Large Inputs

* 🔍 Description:

Long prompts exceeded the token limit allowed for some free-tier models, causing inference errors.

* 🛠️ Steps Taken:
* - Implemented truncation: prompt length is checked and cut to a max of 1000 characters with ellipsis.
* - Logged any truncation for transparency during debugging.
* ✅ Status: Resolved