Intern Task Assignment: LLM + RAG-Based Function Execution API

Objective

Develop a Python-based API service that dynamically retrieves and executes automation functions using **LLM + RAG (Retrieval-Augmented Generation)**. The system should process user prompts, map them to predefined automation functions, and generate executable Python code for function invocation.

Task Requirements

1. Function Registry

Create a registry of common automation functions, such as:

- **Application Control:** Open Chrome, Calculator, Notepad, etc
- **System Monitoring:** Retrieve CPU/RAM usage etc
- **Command Execution:** Run shell commands

Example function(include some more that you):

```
import os
import webbrowser

def open_chrome():
    webbrowser.open("https://www.google.com")

def open_calculator():
    os.system("calc")
```

---

2. LLM + RAG for Function Retrieval

- Store function metadata in a **vector database** (e.g., **FAISS, ChromaDB**).
- Use open source model to convert user queries into embeddings.
- Retrieve the best-matching function dynamically.

---

3. Dynamic Code Generation for Function Invocation

- Generate structured and executable Python scripts based on the retrieved function.
- Ensure proper **imports, error handling, and modularity**.

Example Output for Query: "Launch Google Chrome" you can be creative for this part.

```
from automation_functions import open_chrome

def main():
    try:
        open_chrome()
        print("Chrome opened successfully.")
    except Exception as e:
        print(f"Error executing function: {e}")

if __name__ == "__main__":
    main()
```

---

4. Maintain Context

- **Enhance RAG with session-based memory:** Integrate chat history to provide context-aware responses, ensuring accumulated results from previous interactions are utilized for better function retrieval and execution.

5. API Service Implementation

Develop an API (using **FastAPI, Flask, or Django**) that exposes an endpoint.

POST `/execute`

Request:

```
{ "prompt": "Open calculator" }
```

Response:

```
{ "function": "open_calculator", "code": "<Generated Code Snippet>" }
```

Deliverables

✅ **Function Registry** (Python script with predefined automation functions)
✅ **RAG Model** (Vector search + LLM for function retrieval)
✅ **Dynamic Code Generation** (Structured function invocation scripts)
✅ **API Service** (Function execution via REST API)

Evaluation Criteria

🔹 Accuracy of **function retrieval using RAG**
🔹 **Quality and structure** of generated function invocation code
🔹 **API robustness and error handling**
🔹 Extendability for future **automation tasks**

Bonus (Optional Enhancements)

- Implement **logging and monitoring** for function execution.
- Extend the system to support **custom user-defined functions**.

This assignment will assess your **Python expertise, API development, LLM integration, and problem-solving skills**.

Submission Deadline:
Submission Format: GitHub repository with README and sample requests along with screenshots.