Task: AI Engineer Intern ():- Building a Basic Webpage Opener and PDF Question Answering Agent

Objective: As an AI Engineer intern, your task is to design and implement a simple conversational AI agent using a language model (preferably not a openai model) any open source model of your choice (hint: mistral MOE). The agent should be able to:

1. Open a specified webpage in the user's default browser upon request.
2. Close the opened tab.
3. (PRO-INTERN TASK 😎) Implement a basic Retrieve-and-Generate (RAG) feature to answer questions based on input PDF files based on any open source model.

Requirements:

1. **Webpage Opener and Closer:**
   - Write a bot (agent) Leveraging LLMs that takes user input (with a URL or website name in vary natural manner) and opens it in the default browser.
   - Also implement that the agent is able to close the opened tab.
   - Ensure the script is at least works for (Windows or macOS).
2. **Conversational Agent:**
   - Integrate a language model (Must not be from openai), can be any Open source model (hint: Mistral MOE or can be other of your choice) to handle user interactions in a conversational manner.
   - The agent should understand commands and execute them in very natural manner.
3. **RAG-based PDF Question Answering (**PRO-INTERN TASK 😎**):**
   - This problem statement is intended to showcase your advanced skills, implement a basic RAG model for question answering on PDF files.
   - The agent should take a PDF file and a question as input and return an answer based on the content of the PDF.
   - Use pure Python libraries for PDF processing and RAG (Must not use any libraries) techniques.

Guidelines:

1. Use pure Python for the entire core implementation, avoiding any external dependencies or frameworks (like Langchain or else) unless they are really required (Need explanation why).
2. Structure the code in a modular and clean manner, with appropriate comments and documentation.

3. You can use the Libraries like Langchain or else only as inspiration for understanding concepts but are not required to use it directly and implement the solution. The solution should be purely from scratch with your own implementation.

4. For the RAG feature, you can refer to Hugging Face's implementation as a starting point, but adapt it to work without their framework.

Deliverables:

1. Python scripts containing the conversational agent with the webpage opener and closer functionality.

2. Implemented, the RAG-based PDF question answering module.

3. README file explaining the implementation, dependencies, and how to run the script.

4. Any additional documentation or comments within the code to explain the logic and design choices.

5. All the inputs and outputs should be in very conversational manner with the agent.

6. ALL IMPLEMENTATION SHOULD BE PURELY WRITTEN FROM SCRACTCH WITHOUT ANY USE OF SIMILAR MODULES OR FRAMEWORK (Langchain, chanlit or else).

7. You are advised to apply your own concepts rather than relying on libraries like Langchain or any similar.

EVALUATION WILL BE BASED ON BOTH SOLUTIONS SUBMISSION

GOOOOD LUCK!!!