Low-Code Self-Hosting of RAG-Enabled Language Models Using Ollama and Open WebUI

Julius Sempio
Senior AI Engineer (DOST-NAIRA)
Advanced Science and Technology Institute
Quezon City, Philippines
juliusnoah.sempio@asti.dost.gov.ph
ORCID 0009-0009-8547-0602

Elmer Peramo
Project Leader (DOST-NAIRA)
Advanced Science and Technology Institute
Quezon City, Philippines
elmer@asti.dost.gov.ph

***Abstract*—Large language models (LLMs) are increasingly used for enterprise document processing, yet their adoption in many organizations is limited by (1) the perception that end users must first acquire programmer- or engineer-level skills, and (2) reliance on always-online, proprietary cloud services. This paper presents a practical low-code/no-code (LCNC) approach to self-hosting a retrieval-augmented generation (RAG)–enabled LLM by combining Ollama (to run freely available open-weight models) with Open WebUI (OWU), which offers a graphical interface for nontechnical users to configure and operate their setups. We describe the resulting architecture and evaluate it across multiple deployment scenarios, comparing local CPU-only, local GPU-assisted, and cloud-hosted configurations in terms of response latency and answer quality on a small corpus of domain-specific documents. We further demonstrate its use as a low-code application platform (LCAP) for document processing and information retrieval through the ACABAI-PH Information Resource Assistant (AIRA), a prototype chatbot for navigating government program documents. Our findings highlight the trade-offs between hardware resources, responsiveness, and deployment simplicity when bringing self-hosted RAG systems to resource-constrained, connectivity-challenged environments.**

***Index Terms*—large language model (LLM), self-hosting, low-code/no-code, retrieval augmented generation (RAG)**

I. Introduction

Many Internet users are now familiar with the convenience of language models (LMs) and large language models (LLMs) for tasks involving written text. When end users can effectively harness their capabilities for document processing, these models become powerful tools for streamlining operational procedures in the workplace.

For government agencies in the Philippines, retrieval-augmented generation (RAG) is of particular interest, as these offices seek effective technologies to help process diverse paperwork and produce outputs such as narrative technical reports, position papers, and policy formulation or interpretation documents [1]. However, most staff in these agencies do not have a programming background. LLM advocates often emphasize the engineering-level skills expected of end users [2], and many personnel lack the time or resources to learn how to customize software at the source-code level.

In this context, the paradigm of low-code/no-code (LCNC) solutions is becoming increasingly relevant, both for promoting the adoption of new technologies and for ensuring the inclusion of nontechnical personnel [3]. Accelerated by the rise of "citizen developers" during the COVID-19 pandemic and further enabled by code-capable AI and ML services, low-code application platforms (LCAPs) were projected to account for 65% of application development by 2024 and are expected to continue growing in the coming years, provided that security concerns, which remain the responsibility of IT professionals and experts, are adequately addressed [4].

Another critical factor is reliable access to these applications. Many areas in the Philippines would like to use such technologies but still have limited or no Internet connectivity. Their options are often restricted to paying for expensive commercial providers [5], such as satellite-based Starlink [6], or becoming beneficiaries of government science initiatives for remote Internet access, such as REIINN [7]. Most popular chatbot services, including ChatGPT, Claude, Copilot, and Gemini, require an Internet connection for basic access and a token-based paid subscription for additional features [8]. However, some providers, such as OpenAI, Meta, and DeepSeek, are also releasing open-weight versions of their core LLMs that can be accessed free of charge [9] and integrated into self-hosted LCNC solutions.

With the ability to self-host a RAG-enabled LCAP, nontechnical personnel can, in principle, reduce the time needed to retrieve important and relevant information from the hundreds or thousands of digital documents typically managed in government or corporate settings. When the RAG system performs retrieval as a background task, staff members can either work on other, more meaningful tasks in parallel or take a brief break while waiting for their query to be answered. This is the scenario envisioned for the ACABAI-PH Information Resource Assistant (AIRA), a chatbot initially designed to provide quick answers to questions about the ACABAI-PH program (described in the Acknowledgments section) which has since attracted interest from local and national government agencies in the Philippines.

This paper explores a low-code deployment of a local LLM server designed for users with little or no background in computer programming. The goal is to let them leverage the capabilities of an AI model as an assistant for document

processing without requiring them to learn Python, HTML, R, or other programming languages. Specifically, the system provides a localized, RAG-enabled environment that allows users to upload and process their own documents in support of their desired end goals.

II. Setup details

The two core software applications used to deploy the low-code, self-hosted instance described in this paper are Ollama and Open WebUI (OWU). Both are free open-source tools: users can download and use them at no cost, but active customer support and certain advanced features are available only through a paid subscription. Experienced programmers can often bypass the need for such paid support by implementing customizations themselves or by turning to the applications' online user communities. At the same time, the free versions are sufficiently versatile for use by nontechnical personnel.

A. Ollama

Ollama is a backend self-hosting application for running open-weight LLMs on a user's PC. Deployment is straightforward: users visit the Ollama website, download the installer, and install it on their system.

In addition to LLMs, Ollama can also load embedding models, which are machine learning models that convert input information into numerical representations called embeddings, enabling the LLM to work with that information. An embedding model is essential for RAG, and choosing an appropriate one is critical because the quality of retrieved references directly affects the LLM's generated outputs [10].

Ollama provides its own console interface for interacting with hosted LLMs, but it can also serve as a model hosting platform for several frontend AI/ML applications, including Open WebUI, GPT4All, and LibreChat.

B. Open WebUI

Open WebUI (OWU) is a Python-based frontend platform that provides an interface for running AI and ML models and is designed to operate without an Internet connection. Installing and launching OWU requires Python 3.12 or an earlier version, primarily so that users can run the 'pip' package manager from a Windows or Linux command-line interface to install OWU and its dependencies. Once pip is installed, deploying the platform requires a single command:

```
pip install open-webui
```

Running the platform after installation also requires a single command:

```
open-webui serve
```

To access the OWU interface, navigate to the following address in a web browser:

```
http://localhost:8080/
```

Beyond this one-time setup, users do not need to write or modify any code to operate the self-hosted service.

OWU includes a built-in RAG component as one of its core features, which further supports a "low-code" configuration. It also allows users to configure the optical character recognition (OCR) engine used to read documents, the embedding model used to store document content, and the initial RAG template prompt.

C. System architecture

Figure 1 shows a simple process diagram that uses Ollama and OWU to self-host a RAG setup. In the pre-configuration stage, end users set up various language and embedding models in Ollama. OWU can then be used to select one or more LMs for querying and a single embedding model for RAG. Users with additional programming and container deployment experience may also choose to replace OWU's default OCR tool with an alternative.

Next, users upload documents and references into OWU. The OCR tool and embedding model process these files and populate a RAG-enabled knowledge base, as shown in Figures 2 and 3. Finally, users interact with the LM and the knowledge base through OWU by manually entering their queries.

III. Setup and Testing

A. Testing on different laptops

The combined Ollama and OWU setup is tested using two PC units, one having no GPU and the other with an NVIDIA CUDA-compatible GPU, simulating cases when end users may have equipment with or without a GPU. The specifications for both laptops are listed in Table I.

TABLE I
The specifications of the two laptops used in this study

PC Unit Model	CPU Specs	GPU Specs	PC RAM
Lenovo Thinkpad (2023)	13th Gen Intel(R) Core(TM) i7-1355U, 12 CPUs 1.7GHz	(no CUDA-compatible GPU)	16 GB
ASUS TUF Dash F15 (2022)	11th Gen Intel(R) Core(TM) i5-11300H @ 3.10GHz, 3110 Mhz, 4 Core(s), 8 Logical Processor(s)	NVIDIA GeForce RTX 3060 6GB	24 GB

The following are the configurations made for the RAG-enabled self-hosting service:

- OCR: SentenceTransformers (the available default option for OWU)
- Ollama-hosted LLMs: gpt-oss:20b (the LLM in focus for this paper), gemma3:12b, phi4:14b, llama3.1:8b
- Ollama-hosted embedding model: embeddinggemma

To evaluate the RAG capability of the setup, PDF documents about the ACABAI-PH Program and its DOST-NAIRA component (see the Acknowledgments section for details) were loaded into the knowledge base.

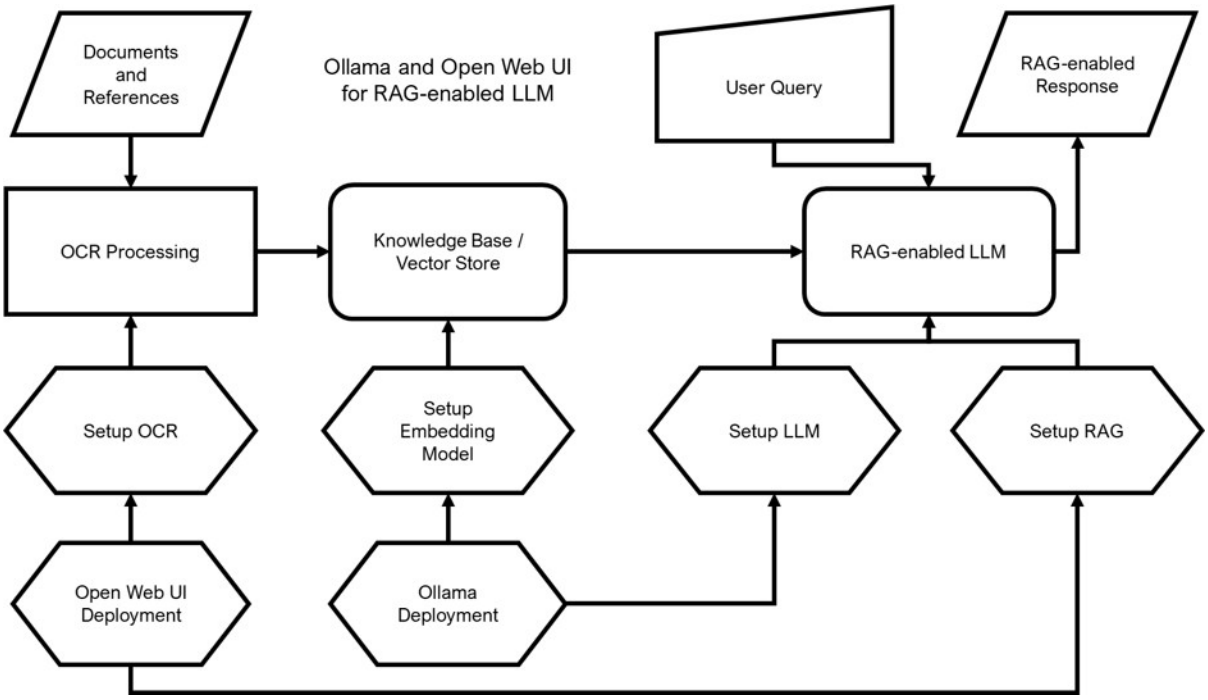

Fig. 1. Process diagram for enabling RAG using Ollama and Open WebUI.

Fig. 2. Screenshot of OWU showing various test knowledge bases in a card catalog-like form.

To compare response times between the two laptops, we evaluated 24 queries (8 factual, 8 synthesis, 8 reasoning) across three corpora sizes (6, 20, and 100 documents). Each query was executed 10 times per setup; we report mean ± standard deviation, and we apply a Mann–Whitney U test for CPU-only vs GPU-assisted latency. The first query concerned generally known information, while the second required RAG-specific knowledge drawn from the loaded documents. A third setup was also tested by sending the same queries to a GPT-Oss:20b model hosted by Ollama's high-performance cloud service under a free trial, representing an idealized control scenario in which users have access to powerful computing resources. For each setup, three runs were performed, and the average response time was recorded in minutes (or in seconds for the cloud-hosted LLM). The results are summarized in Table II.

As the values in Table II indicate, having a working GPU for the locally hosted setup leads to much faster response times.

TABLE II
RECORDED RESPONSE TIMES IN OWU OF THE TWO LAPTOPS USED IN THIS STUDY

RAG-enabled GPT-Oss:20b setup	**Common Non-Technical Prompt:** ***"Please provide me a good recipe for Filipino adobo. Thanks!"***	**RAG-Dependent Prompt:** ***"Hello! What is the NAIRA project? Thanks!"***
Setup 1: Work-issued laptop without GPU	23 minutes	28 minutes
Setup 2: Personal gaming laptop with GPU	2 minutes	6 minutes
Setup 3: Using Ollama Cloud	2 seconds	8 seconds

In our tests, the GPU-enabled laptop reduced waiting time to roughly one quarter of that on the non-GPU laptop. The LLM hosted on Ollama Cloud outperformed both local setups, which is expected given its access to the provider's high-performance computing infrastructure.

To examine the impact of enabling OWU's RAG feature on the performance of a locally hosted LLM, a "fourth setup" was conducted. The service was made to run three separate tests, the first one simultaneously running non-RAG and RAG-enabled instances of the Ollama-hosted GPT-Oss:20b LLM, and the other two tests running the non-RAG LLM and RAG LLM in their respective instances. For all three tests, the same nontechnical prompt was given, and the resulting log times are provided in Table III.

TABLE III
RECORDED RESPONSE TIMES OF OLLAMA-HOSTED GPT-OSS:20B LLM IN OWU TO THE SAME NON-TECHNICAL PROMPT
"HELLO! CAN YOU GIVE ME A GOOD RECIPE FOR FILIPINO ADOBO? THANKS!"

Test Type	**RAG-enabled setup**	**Non-RAG-enabled setup**
Both RAG-enabled and RAG-disabled	Start: 4:35PM First Token: 4:41PM End: 4:42PM	Start: 4:35PM First Token: 4:36PM End: 4:41PM
RAG-enabled only	Start: 4:43PM First Token: 4:48PM End: 4:53PM	N/A
RAG-disabled only	N/A	Start: 4:54PM First Token: 4:54PM End: 5:00PM

The simultaneous run showed that the non-RAG-enabled LLM took around a minute to begin its thinking phase and provided a working answer five minutes after the first token. The RAG-enabled LLM, on the other hand, needed six minutes to begin its thinking phase, but as it refused to answer the query, reasoning that the question is out of the context of its knowledge base, it only took a minute from its first token to complete.

For the individual runs of each of the two LLM instances, the non-RAG-enabled setup performed closely to the simultaneous test: it took less than a minute to begin thinking phase, and then completed its working answer in around five minutes after the first token. The RAG-enabled setup, on the other hand, performed differently: similar to the simultaneous run, it took time to reach its thinking phase at around five minutes, but unlike the simultaneous run, it chose to provide a working answer that took around five minutes after the first token to finish.

As a conclusion, regardless of whether the RAG-enabled OWU-configured LLM chose to answer out of context or not, the tests nevertheless demonstrate that enabling RAG adds noticeable overhead to an Ollama-hosted LM's response time even when actual referencing to the embedded corpus is not strictly needed, as manifested by the significant differences in times to the first token.

B. Assessment of self-hosted RAG with Ragas

In terms of subjective response quality, all setups produced reasonably accurate answers to the ACABAI-related questions, although this judgment relied on the authors' familiarity with the program. In this first round of testing, to obtain a more impartial measure of correctness, the GPU-enabled Ollama-hosted GPT-Oss:20b LLM was evaluated twice in a single experiment on Ragas - an evaluation module introduced by researchers from Exploding Gradients that combines LLM-driven metrics with systematic experimentation [12] - using two streamlined PDF documents about ACABAI and NAIRA. The results of this evaluation are summarized in Table IV.

Both Ragas experiments were completed in an average of nine minutes, and subsequent tests were also completed within the nine-minute range. The correctness metrics for each conducted test also averaged to 'Pass' marks and one 'Fail' mark.

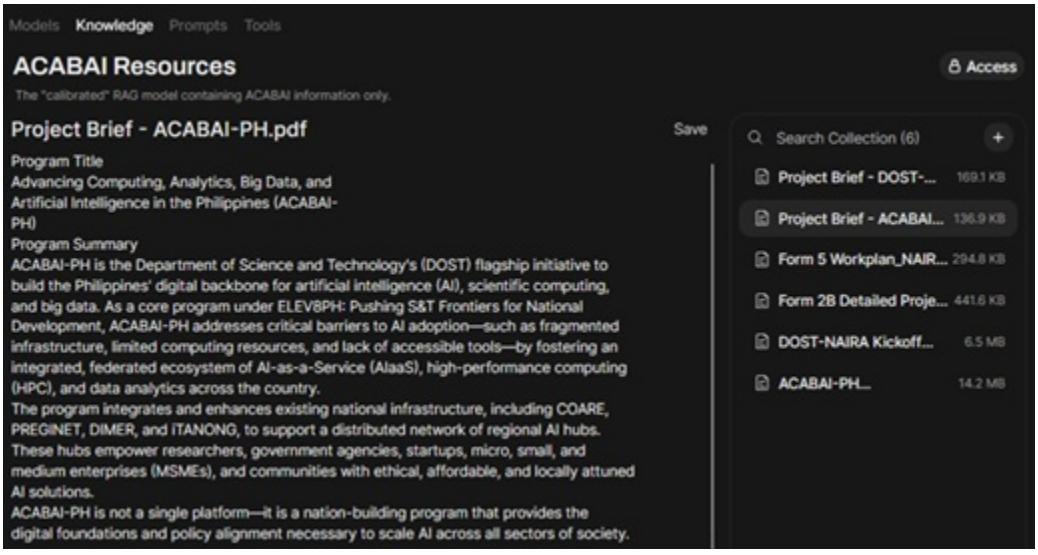

Fig. 3. Previewing the contents of one of the files listed in the knowledge base.

TABLE IV
RAGAS-BASED RESPONSE TIMES AND CORRECTNESS ASSESSMENT OF THE GPU-ENABLED OLLAMA-HOSTED GPT-OSS:20B LLM

Question	**Test 1**	**Test 2**
	Start: 4:50PM First Token: 4:53PM End: 4:59PM	Start: 5:06PM First Token: 5:08PM End: 5:15PM
What is the ACABAI program?	Pass	Fail
What is the NAIRA project?	Fail	Pass
What skillsets are expected of NAIRA employees?	Pass	Pass

An additional three Ragas tests were then conducted to assess the performance of the GPU-enabled Ollama-hosted GPT-Oss:20b LLM. The documents used for this RAG test consist of twenty text-heavy PDF documents sourced from institutions such as UNESCO, the World Economic Forum, and the Alan Turing Institute among many others, discussing ethics and governance in the use of AI and ML, and each averaging 60 pages.

The focus of this set of tests is not only on the speed of the LLM's response to ten questions, but also on manipulating its context length and its effect on correctness. The recorded times and subsequent correctness metrics are summarized in Table V.

For all the tests conducted, it is observed that increasing the LLM's context length also increases the time it takes for the first token to arrive: four minutes for Test 1 with 4000 context length, nine minutes for Test 2 with 16000 context length, and eight minutes for Test 3 with 32000 context length. Increasing the context length also increases the time it takes Ragas to complete its evaluation for the same number of questions: Test 1 took 45 minutes, Test 2 took 53 minutes, and Test 3 took 64 minutes.

As for the recorded correctness metrics, Test 1 performed categorically the worst with only three 'Pass' marks and seven 'Fail' marks, whereas Tests 2 and 3 performed slightly better, with an equal number of 'Pass' and 'Fail' marks.

Two notable caveats the authors wish to share are that the quality of responses by the self-hosted setup is 1) highly dependent on the quality of the documents used in RAG, and 2) dependent on the LM or LLM processing the query. The second point of the two conditions is given its respective discussion in the Further Activities section at the latter part of this paper.

IV. CHALLENGES

The most notable limiting factor of a self-hosted RAG-enabled LLM service is the capacity of the nontechnical user's computer unit, especially if GPUs are unavailable. Because PC and laptop units with powerful GPUs are considered luxury items in the Philippines, local and national government agencies are content with providing their personnel with mid- to low-range (and often expensive) computer hardware [13]. As such, the majority of local administrators have to make do with the underpowered tools that are provided to them.

Yet while the performance of self-hosted RAG-enabled LLMs in non-GPU-enabled computer hardware is slow, they are still faster at producing answers than asking an untrained staff member to browse through physical paperwork and digital PDFs just to come up with a proper answer. Staff can also let the system run in the background while they attend to other tasks.

An additional limitation addressed in this paper is that the knowledge bases used in the study may be considered small for testing the setup's RAG capability. The ACABAI-PH knowledge base consists of six text-heavy documents in PDF format, which can be deemed too small, whereas the AI Ethics knowledge base is moderately sized at twenty documents (note that the researchers have gathered more than a hundred documents on AI ethics to make a larger corpus, but then a subset of twenty documents of sufficient quality were selected for testing purposes). Further testing of the robustness of Ollama-hosted embedding models for use by OWU's RAG will require more voluminous documentary resources, such as training and other reading materials that can be provided by stakeholder agencies, or by using benchmarks such as four corpora that are provided by IBM [15].

One more potential limitation for the RAG setup is the unavailability of additional free and ready-to-deploy OCR options for OWU besides its default offering of SentenceTransformers, as Ollama currently does not provide hosting support for OCR engines. OWU does allow for other OCR models (e.g. Docling, Mistral OCR, Datalab Marker, MinerU) but setting them up will require additional steps such as making Docker containers or obtaining service API keys which, while ideal to explore for process optimization studies, defeats this paper's intent of an LCNC paradigm for the general benefit of nontechnical end users.

For this study, the researchers prioritized PDFs in digital text-based formats, effectively minimizing the contribution of OCR to the system. Moving forward, the need for a good OCR engine may manifest as government agencies begin providing scanned documents as information resources.

V. FURTHER ACTIVITIES

This study is part of DOST-NAIRA's broader effort to harness LLMs for internal knowledge management and business intelligence. By indexing key project documents, reports, and guidelines in a self-hosted RAG system, agencies can quickly retrieve prior decisions, lessons learned, and policy commitments to support planning and reporting. Within this context, the Ollama–OWU stack is being tested as a RAG-enabled low-code application platform for capacity building in government offices, academic institutions, and micro, small, and medium enterprises (MSMEs).

A. AIRA

NAIRA's proof-of-concept (POC) for the self-hosted RAG-enabled LCAP service currently exists as the *ACABAI-PH Information Resource Assistant (AIRA)*. Originally intended as

TABLE V
RAGAS-BASED RESPONSE TIMES AND CORRECTNESS ASSESSMENT OF THE GPU-ENABLED OLLAMA-HOSTED GPT-OSS:20B LLM

Question	**Test 1** Start: 12:21AM First Token: 12:25AM End: 1:05AM Context Length: 4k	**Test 2** Start: 12:52PM First Token: 1:01PM End: 1:45PM Context Length: 16k	**Test 3** Start: 1:46PM First Token: 1:54PM End: 2:50PM Context Length: 32k
What is AI Ethics?	Pass	Fail	Pass
What are the benefits of AI to society in general?	Fail	Pass	Fail
What are the practical uses of AI to businesses and industries?	Fail	Pass	Pass
What are the practical uses of AI to governments?	Fail	Pass	Pass
What are the potential dangers of unrestricted AI development?	Fail	Fail	Fail
Which countries are considered advanced in the field of AI?	Pass	Fail	Pass
How much does the United States spend on AI development?	Pass	Fail	Pass
What steps are being proposed to prevent the abuse of AI?	Fail	Pass	Fail
Which sectors are the targeted beneficiaries of AI in the Philippines?	Fail	Pass	Fail
What are the programs the Philippine government is currently implementing or planning to implement to foster AI development in the country?	Fail	Fail	Fail

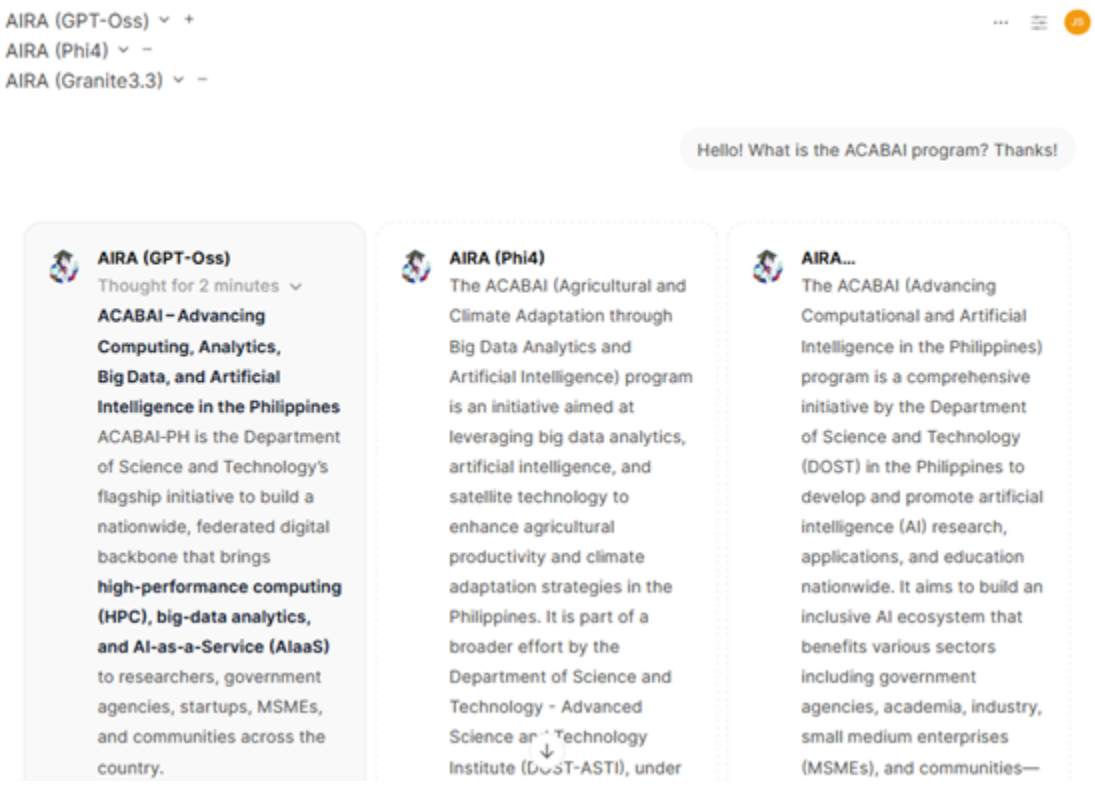

Fig. 4. Screenshot of OWU running multiple test AIRA chatbots configured using different LLMs for testing the response performance of said LLMs.

a use case for simultaneous testing of multiple open-weight LLMs (as shown in Figure 4), AIRA evolved into a specialized chatbot capable of performing the following:

- Providing answers to questions about ACABAI and its component projects using a knowledge base containing information about the program
- Formulating recommendations for up-skilling personnel based on the resumé and/or curriculum vitae uploaded by the user and the ACABAI knowledge base
- Providing outlines of possible training courses the above-mentioned personnel can take using a knowledge base containing training materials provided by several agencies
- Processing documents on AI ethics and policy formulation for quick reference, once again using its dedicated knowledge base

AIRA is intended to serve as a promotional figure in future roadshows and technology demonstrations across the Philippines, helping raise awareness of the versatility of LLMs. This follows the example of a previous DOST LLM project that successfully showcased the potential of practical AI and ML tools to government agencies and the broader Filipino public [14]. Looking ahead, AIRA is envisioned to evolve into an LLM-powered learning management system (LMS), providing accessible educational materials to interested government agencies and private enterprises.

B. Behaviors of various LLMs

An additional avenue for exploration is the feasibility of different open-weight LLMs for enterprise-level RAG, an active area of study because it addresses the high licensing costs and restrictive usage policies associated with proprietary LLMs [11]. From preliminary testing of the growing number of LLMs that Ollama can host, it was observed that each model, *in its default setting*, responds to the same query in a different way, exhibiting distinct response styles and varying degrees of hallucination. These behaviors can be quantified in future work using RAG-oriented evaluation frameworks such as Ragas, DeepEval, and ARES [12]. Some of the response patterns observed during the testing of various LLMs for AIRA are summarized in Table VI.

In AIRA's case, the tendency of the GPT-Oss model to structure responses in table and list form, and its constant inclusion of a "thinking phase" to explain its answer (thus pointing to either an inherently high reasoning effort parameter or an automatic feature of explainable AI or xAI), made it the favored LLM to use in the POC.

Another potentially viable research direction is to manipulate an open-weight LLM's parameters to optimize RAG performance, and OWU provides a UI for tweaking such model's weights. To see if the manipulated parameters are causing the LLM to give better answers, the RAG-enabled GPT-Oss:20b model hosted in Ollama Cloud can serve as a benchmark, since it provides very accurate responses to queries about ACABAI and NAIRA.

TABLE VI
OBSERVED PATTERNS ON THE BEHAVIOR OF DIFFERENT OPEN-WEIGHT LLMS IN AN OLLAMA-HOSTED RAG ENVIRONMENT

Tested Open-Weight LLM	**Typical Response Pattern**	**Response Verbosity**	**Proneness to Hallucinations**
GPT-Oss	Usually in table and list form, uses bold and/or italicized fonts, and with a "thinking phase"	High	Medium
LLaMA 3.1	Usually in paragraph form	Low	Low
Phi 4	Usually in list and paragraph form	Medium	Low
Gemma 3	Usually in list and paragraph form and uses bold and/or italicized fonts	High	Medium
Granite-3	Usually in table and list form	Medium	High
DeepSeek-R1	Usually in paragraph form, and with a "thinking phase"	Medium	Medium

VI. ACKNOWLEDGMENTS

This research is made possible by the Nexus for Artificial Intelligence Research and Applications (NAIRA), one of the major projects under the Advancing Computing, Analytics, Big Data, and Artificial Intelligence in the Philippines (ACABAI-PH) program of the Philippines' Department of Science and Technology (DOST). This work is being funded and monitored by the DOST Philippine Council for Industry, Energy, and Emerging Technology Research and Development (PCIEERD) with Project No. 1213385.

A copy of the low-code/no-code user manual created for the purposes of deploying the localized hosting service described in this paper can be requested by sending an email to julius-noah.sempio@asti.dost.gov.ph

REFERENCES

[1] L. Yun, S. Yun and H. Xue, "Improving citizen-government interactions with generative artificial intelligence: Novel human-computer interaction strategies for policy understanding through large language models," PLOS One, vol. 19, no. 12, 2024.

[2] K. Lam, "ChatGPT for low- and middle-income countries: a Greek gift? (Comment)," The Lancet Regional Health - Western Pacific, vol. 41, pp. 1-2, December 2023.

[3] M. O. Ajimati, N. Carroll and M. Maher, "Adoption of low-code and no-code development: A systematic literature review and future research agenda," The Journal of Systems & Software, vol. 222, pp. 1-25, April 2025.

[4] G. F. Huribert, "Low-Code, No-Code, What's Under the Hood?," IT Professional, vol. 23, no. 6, pp. 4-7, 17 December 2021.

[5] N. Kanehira, M. Abdon and M. G. Mirandilla-Santos, "Upgrading Philippine internet for faster and inclusive growth," World Bank, 4 April 2024. [Online]. Available: https://blogs.worldbank.org/en/eastasiapacific/upgrading-philippine-internet-for-faster-and-inclusive-growth. [Accessed 5 November 2025].

[6] R. C. Dela Cruz, "NTC OKs registration of Elon Musk's Starlink," Philippine News Agency, 27 May 2022. [Online]. Available: https://www.pna.gov.ph/articles/1175290. [Accessed 5 November 2025].

[7] C. Luci-Atienza, "REIINN: This DOST project connects rural, far-flung areas to online world," Manila Bulletin, 9 December 2021. [Online]. Available: https://mb.com.ph/2021/12/09/reiinn-this-dost-project-connects-rural-far-flung-areas-to-online-world/. [Accessed 5 November 2025].

[8] G. Hickey, "Best Large Language Models (LLMs) of 2025," TechRadar, 17 July 2025. [Online]. Available: https://www.techradar.com/computing/artificial-intelligence/best-llms. [Accessed 5 November 2025].

[9] G. Huckins, "OpenAI has finally released open-weight language models," Massachusetts Institute of Technology, 5 August 2025. [Online]. Available: https://www.technologyreview.com/2025/08/05/1121092/openai-has-finally-released-open-weight-language-models/. [Accessed 5 November 2025].

[10] S. Chen, Z. Zhao and J. Chen, "Each to Their Own: Exploring the Optimal Embedding in RAG," Cornell University, 20 August 2025. [Online]. Available: https://arxiv.org/abs/2507.17442. [Accessed 5 November 2025].

[11] G. Balakrishnan and A. Purwar, "Evaluating the Efficacy of Open-Source LLMs in Enterprise-Specific RAG Systems: A Comparative Study of Performance and Scalability," in IEEE 21st India Council International Conference (INDICON), Kharagpur, 2024.

[12] M. Antal and K. Buza, "Evaluating Open-Source LLMs in RAG Systems: A Benchmark on Diploma Theses Abstracts Using Ragas," Acta Universitatis Sapientiae, Informatica, vol. 17, no. 5, pp. 1-15, 2025.

[13] M. Hernando-Malipot, "DepEd responds to Ombudsman charges over alleged overpriced laptop procurement," Manila Bulletin, 12 July 2025. [Online]. Available: https://mb.com.ph/2025/07/12/deped-responds-to-ombudsman-charges-over-alleged-overpriced-laptop-procurement/. [Accessed 11 December 2025]

[14] D. Solano, "DOST to launch 'Filipino-style' ChatGPT," Philippine Star Tech, 10 October 2023. [Online]. Available: https://philstartech.com/news/2023/10/10/1128/dost-to-launch-filipino-style-chatgpt/. [Accessed 11 December 2025]

[15] Y. Katsis, S. Rosenthal, K. Fadnis, C. Gunasekara, Y.S. Lee, L. Popa, V. Shah, H. Zhu, D. Contractor and M. Danilevsky, "MTRAG: A Multi-Turn Conversational Benchmark for Evaluating Retrieval-Augmented Generation Systems," Transactions of the Association for Computational Linguistics, vol. 13, pp. 784–808, 2025, doi: https://doi.org/10.1162/TACL.a.19. Available: https://arxiv.org/abs/2501.03468. [Accessed 22 December 2025].