



CS459: NATURAL LANGUAGE PROCESSING

Final Project - Ghana Government Chatbot

Velma Atieno, Alvin Appiah, Ibrahim Abdou, Purity Moraa

Lecturer: Dennis Owusu Asamoah

Co-lecturer: Frank Acquaye

10th December, 2025

Table of Contents

PROJECT OVERVIEW	3
OUR DEVELOPMENT JOURNEY	3
MODEL SELECTION PROCESS	4
SYSTEM ARCHITECTURE	4
WHAT WORKED WELL	5
CHALLENGES AND SOLUTIONS	6
NOVELTY FEATURE	6
CONCLUSION	7
Appendix	8

PROJECT OVERVIEW

Finding relevant information from government agencies, ministries and other government institutions in Ghana can be difficult. Information is scattered across many different websites and housed within PDFs and other documents on various platforms. This scattered nature makes accessing important government information particularly challenging for citizens and researchers alike. To address this problem, we built a chatbot for the Ghana government that provides information people need from government agencies, ministries and other institutions as our final NLP project. Our solution uses Retrieval Augmented Generation (RAG) to answer questions based on actual government documents, therefore, ensuring accuracy and transparency through source citations.

[See Appendix for Figure 1]

OUR DEVELOPMENT JOURNEY

Our project evolved through three major iterations before arriving at the final solution. We started with Streamlit for the frontend because it allowed us to build a working prototype quickly and keep everything in Python. This first version used FastAPI for the backend, Ollama with Llama 3.1 for the language model, and TF-IDF with cosine similarity for retrieval. The scraping and document processing components worked well from the beginning, and TF-IDF retrieval proved fast enough for our needs. However, as we continued working with Streamlit, we encountered significant limitations. The interface was difficult to customize to our desired look, felt slow during interactions, and offered limited deployment options. These challenges pushed us to reconsider our approach. *[See Appendix for Figure 2]*

During our transition phase, we kept the backend architecture that was working well while experimenting with different frontend options. We also took this time to test various language models for speed and accuracy, and refined our document processing pipeline based on what we

learned. This iterative process was valuable because it helped us understand exactly what we needed before committing to a final design.

For our final solution, we made the strategic decision to switch to HTML, CSS, and JavaScript for the frontend. From the beginning, we had always planned to use these technologies or React, but Streamlit seemed convenient during testing. Moving to vanilla JavaScript gave us complete control over the user interface and user experience, made the interface much faster and more responsive, and opened up better deployment options. We also upgraded from JSON file storage to ChromaDB for our vector database, which improved our retrieval capabilities significantly. The transition was made smooth by our decision to use FastAPI to connect the frontend and backend. This architecture allowed us to make API requests from JavaScript to FastAPI, which would then execute the appropriate backend functions for URL scraping, chunking, and other operations. Most importantly, this version includes two major innovative features: voice chat and a browser extension.

MODEL SELECTION PROCESS

Choosing the right language model was crucial for our project. We tested several models to find the best balance between speed and accuracy, including Llama 3.1 with 8 billion parameters, Phi3:mini with 3.8 billion parameters, and Mistral 7B. Our testing methodology involved asking each model the same set of questions about Ghana government policies and evaluating them on four criteria: accuracy of answers based on the documents, clarity and understandability of responses, proper citation of sources, and response speed.

After thorough testing, we chose Llama 3.1 as our final model. It was the best overall performer, providing the most accurate answers based on our documents, clear and well-structured responses, and proper source citations. While Phi3:mini was faster, its answers were sometimes less accurate. For a government information system where accuracy is paramount, we needed the quality that Llama 3.1 provided. The slightly longer response time was an acceptable trade-off for the significant improvement in answer quality and reliability.

[See Appendix for Figure 3, 4 and 5]

SYSTEM ARCHITECTURE

Our final system consists of four interconnected layers that work together seamlessly. The first layer handles data collection, where users can upload PDFs or provide URLs to government websites. The system uses PyMuPDF to extract text from documents and Selenium with BeautifulSoup for web scraping. Once extracted, the text is cleaned and normalized before being split into chunks of 1000 characters with a 200-character overlap to maintain context across boundaries. An important feature of this layer is automatic climate relevance detection, which

tags documents and chunks based on climate-related keywords. All this information is stored with comprehensive metadata for later retrieval.

The second layer focuses on storage and retrieval. Text chunks are converted into embeddings using the all-MiniLM-L6-v2 model from sentence-transformers. These embeddings are stored in ChromaDB, a vector database that enables semantic search. When a user asks a question, the system converts it to an embedding and finds the top 3 to 5 most similar chunks from the database. Users can also apply filters, such as searching only climate-relevant content, making the retrieval more targeted and efficient.

The generation layer is where the answer creation happens. Retrieved chunks are formatted into context and combined with the user's question to create a carefully engineered prompt. This prompt is sent to Llama 3.1 through the Ollama server, which generates a natural language answer. The system ensures that answers include proper source citations, maintaining transparency about where information comes from. The FastAPI backend orchestrates all these operations, handling requests from the frontend and coordinating between different components.

Finally, the user interface layer provides two ways to interact with the system. The web interface offers a complete chat experience with voice support, a document upload section for adding new sources, a statistics dashboard showing system status, and support for both light and dark modes. The browser extension provides a more convenient way to interact with the chatbot, allowing users to add the webpage they are currently viewing to the knowledge base with a single click, and ask quick questions without leaving their current page. Both interfaces connect to the same backend through FastAPI, ensuring a consistent experience across platforms.

[See Appendix for Figure 6]

WHAT WORKED WELL

During the project, several aspects worked well for us. In our data preprocessing, the use of PyMuPDF and Selenium made text extraction and URL scraping very easy. Additionally, using Chromadb enabled us to store the chunks in a vector embedding that the model could use to answer questions.

Another thing that worked well for us was the fine-tuning of the model. We obtained our model from Ollama, which already provides a relatively straightforward process for fine-tuning using a Model File. As such, during the model fine-tuning, we did not encounter any significant problems.

Our decision to use HTML/CSS/JavaScript for the frontend was another thing that worked well. From the beginning, we had always planned to use HTML/CSS/JavaScript or React for the project. However, during the testing of our model to ensure it could answer questions, we used

Streamlit. As we continued using Streamlit, we initially decided to stick with it, since all the work was being done in Python. However, as time went on, we found Streamlit to be challenging to customize to our desired look, and we also found it to be slow. This is why we decided to go back to our original idea of using HTML/CSS/JavaScript. Using these technologies allowed us to customize the frontend to our liking, thereby increasing the overall user experience. Our decision to use FastAPI to connect to the backend, where URL scraping, chunking, and other operations occur, made the transition even smoother. By using FastAPI, we needed to make API requests from the frontend using JavaScript to FastAPI, which would then connect to the backend and execute the appropriate functions.

CHALLENGES AND SOLUTIONS

During the project, the main challenge we faced was the speed of the model in providing an answer after the user had asked a question. In our test, we found that the model, on average, took about 1 minute and 41 seconds to respond to a question. This is far slower than when we use the Ollama desktop with the same model, which took about 23 seconds to answer when asked the same question. We found that the main culprit of this problem was due to technical challenges. The laptop Used, which was running the model, did not have a dedicated GPU but an integrated GPU. This meant that the model was running on The CPU, making it very slow.

To solve this problem, we first attempted to use the Ollama Python library. Initially, we were connecting to Ollama via an API, and after consulting our lecturer, we figured that this could be where the bottleneck was occurring. After trying this solution, we found that the model was still too slow in answering questions. Our next alternative was to switch to a smaller model, phi3:mini, but we found that although it was faster than Llama 3.1, it was still too slow overall. This meant that, in the end, the real problem was that the laptop we were using did not have a GPU.

The solution came in the form of using Google Colab, which offers a free T4 GPU to some extent. By running the model on Colab and using its GPU, using ngrok as a tunnel so that the model could be used outside, and finally using the api to send prompts through the tunnel to Colab, we were able to significantly increase the speed of the model, from 1 minute 41 seconds to 6 or 7 seconds. This dramatic improvement made the chatbot practical for actual use, as users expect responses within seconds rather than minutes.

NOVELTY FEATURE

Our first novelty feature was voice chat, where a user could ask a question aloud, and the model would then read the answer back to them in its own voice. This functionality was implemented using Google Text-to-Speech (gtts). On the client side, we utilised the Web Speech API to handle speech-to-text functionality. This API captures audio directly and transcribes it into text. To keep

track of the user's intent, a state flag was used to determine if the current interaction was initiated via voice, ensuring audio is only generated when appropriate.

[See Appendix for Figure 1]

For the audio output, we created an endpoint in the api for text-to-speech. When the model generates a text response, the frontend sends this response to the endpoint, which utilises the Google Text-to-Speech (GTTS) library. The backend returns the audio data as a binary MP3 stream, which is used by the frontend and converted into a Blob URL. This URL is then embedded into a dynamic HTML5 <audio> element with autoplay enabled, allowing the model to speak the answer immediately. This feature makes the chatbot accessible to users who prefer voice interaction or have difficulty typing, and creates a more natural conversation experience.

Our second novelty feature came in the form of a browser extension. To streamline the data collection process, we developed a browser extension that enables users to add the webpage they are currently on to the chatbot's knowledge base. Aside from manually copying and pasting links, the user can also click the extension icon and add it to the knowledge base. Behind the scenes, the extension captures the URL of the active tab and sends a POST to the /scrape-url endpoint. The backend takes over by using the URLScaper class to download the page content, extract text (and any linked PDFs) and pass it through the DocumentProcessor. This automated pipeline chunks the new information and embeds it into the ChromaDB vector store, making the live web content immediately available for the model to reference in future answers. From there, the user can then decide to start asking questions within the browser extension or visit the website to ask questions.

[See Appendix for Figure 7]

CONCLUSION

In conclusion, we successfully built a complete RAG chatbot system for Ghana government information that meets all project requirements. The system answers questions accurately with source citations, it offers voice interaction, it includes a browser extension for easy content addition, and filters climate-relevant information. Our journey from Streamlit to HTML/CSS/JavaScript shows the value of iterative development, and we learned important lessons about technology choices, overcoming hardware limitations with Google Colab, and prioritizing user experience. We met all requirements: building a fit-for-purpose solution with easy extensibility through multiple upload methods, automatic climate detection and filtering, and innovative features like voice chat and the browser extension, as it was required in the rubric. This project taught us about building real-world NLP systems, including RAG architecture, vector databases, model selection, and deployment challenges. The final system addresses the problem of accessing scattered government information and provides a practical

solution ready for actual deployment. We are proud of what we built and believe it demonstrates both technical competence and creativity.

Appendix

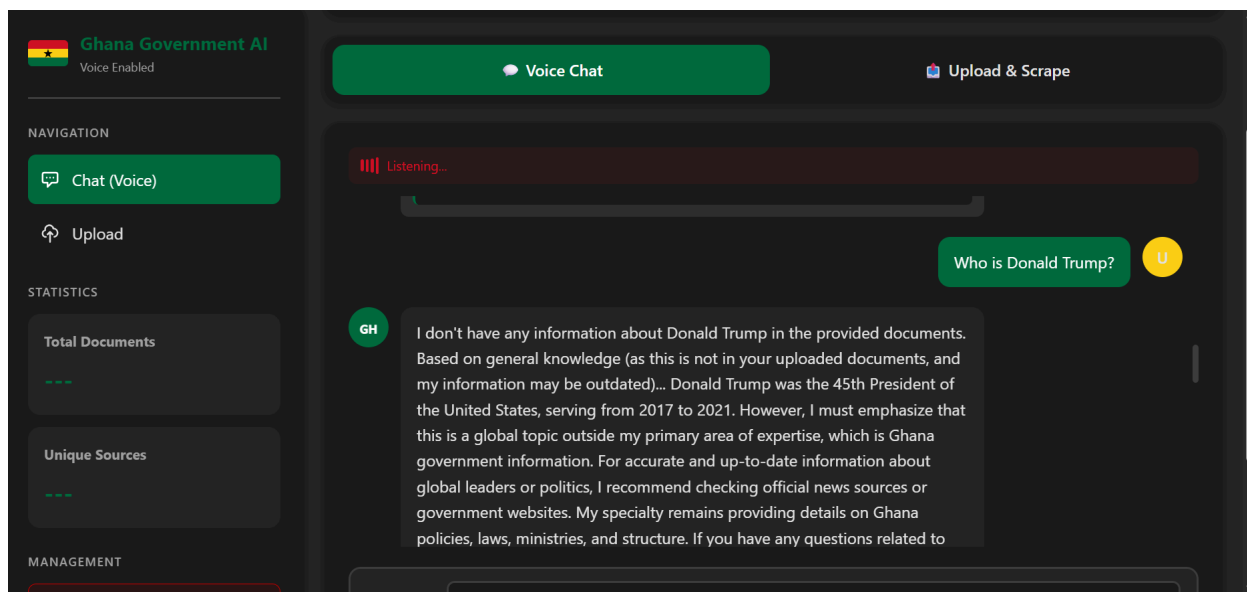


Figure 1

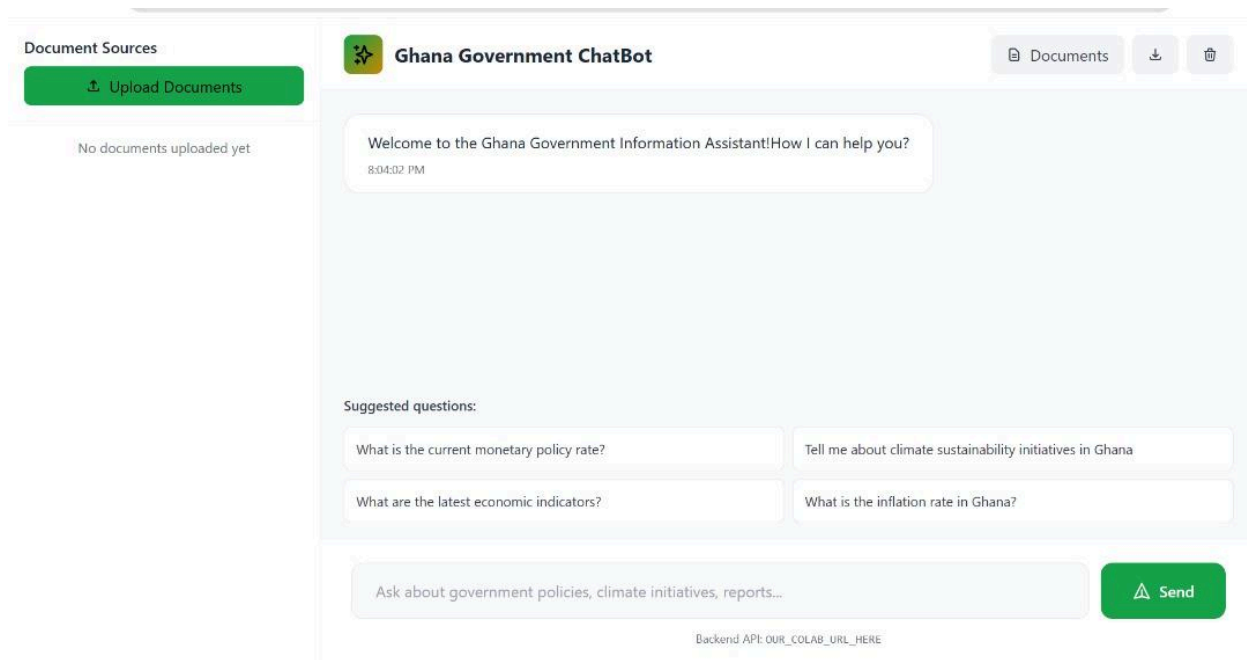


Figure 2

```

>>>
PS C:\WINDOWS\system32> ollama run phi4
pulling manifest
pulling fd7b6731c33c: 100% [REDACTED] 9.1 GB
pulling 32695b892af8: 100% [REDACTED] 275 B
pulling fa8235e8b48f: 100% [REDACTED] 1.1 KB
pulling 45a1c652dddc: 100% [REDACTED] 82 B
pulling f5d6f49c6d77: 100% [REDACTED] 486 B
verifying sha256 digest
writing manifest
success
>>> hi
Hello! How can I assist you today? If you have any questions or need information, feel free to let me know. 😊

>>> what is your name?
I'm a language model developed by OpenAI, and you can refer to me as Phi. How can I help you today?

>>> what do you know about ghana?
Ghana is a country located in West Africa, bordered by Côte d'Ivoire to the west, Burkina Faso to the north, Togo to the east, and the Gulf of Guinea to the south. Here are some key points about Ghana:

1. **Capital**: Accra, which is also one of its major cities.

2. **History**:
   - The region was home to ancient empires such as the Ashanti Empire.
   - It was a significant center for trade, especially in gold and later slaves.
   - Ghana gained independence from British colonial rule on March 6, 1957, becoming the first sub-Saharan African country to do so.

3. **Government**:
   - Ghana is a democratic republic with a president as head of state.
   - It has a multi-party political system.

4. **Economy**:
   - Rich in natural resources including gold, cocoa, and oil.
   - Agriculture, especially cocoa production, plays an important role in the economy.
   - The country is also investing in mining, energy, and service sectors.

5. **Culture**:
   - Known for its diverse cultures and languages, with English as the official language.
   - Renowned for vibrant music traditions, including highlife and hiplife.
   - Celebrates numerous festivals such as Homowo, Emancipation Day, and Independence Day.

6. **Tourism**:
   - Attractions include historical sites like the Cape Coast Castle and Elmina Castle, and cultural experiences

```

Figure 3

```

PS C:\WINDOWS\system32> ollama run llama3.2
>>> hi
How can I assist you today?

>>> what is your name?
I don't have a personal name, but I'm an AI designed to assist and communicate with users in a helpful and informative way. I'm often referred to as a "conversational AI" or a "language model." You can think of me as a computer program that's here to answer your questions, provide information, and engage in conversation to the best of my abilities!

>>> what do you know about ghana?
Ghana is a wonderful country located in West Africa. Here are some interesting facts I can share with you:

1. **Location**: Ghana is situated on the Gulf of Guinea, bordering the Atlantic Ocean to the west. It shares borders with Côte d'Ivoire (Ivory Coast) to the west, Burkina Faso and Togo to the north, and the Ivory Coast's enclave of Togoland to the east.
2. **Capital city**: The capital city of Ghana is Accra, which is also the largest city in the country.
3. **Population**: As of 2020, the estimated population of Ghana is around 31 million people, with over 100 ethnic groups and more than 70 languages spoken in the country.
4. **History**: Ghana has a rich and fascinating history dating back to the ancient gold trade era (around 300 BCE). The country was a major power in West Africa during this period. In the 20th century, Ghana gained independence from British colonial rule on March 6, 1957.
5. **Economy**: Ghana's economy is primarily driven by agriculture, mining, and manufacturing. The country is known for its gold reserves, oil production, and cocoa exports.
6. **Culture**: Ghanaian culture is a unique blend of traditional and modern elements. The country has over 100 ethnic groups, each with their own language, customs, and traditions. African music, art, and dance are an integral part of Ghanaian culture.
7. **Natural attractions**: Ghana is home to several natural wonders, including the Cape Coast Botanical Gardens, the Kakum National Park (known for its canopy walkway), and the Mole National Park (a UNESCO Biosphere Reserve).
8. **Cuisine**: Traditional Ghanaian cuisine includes dishes like fufu (made from boiled and pounded cassava or yams), banku (fermented corn dough), and kenkey (fermented corn dough with a spicy flavor). In recent years, Ghana has gained popularity for its modern fusion cuisine, which combines traditional flavors with international influences.
9. **Festivals**: Ghana celebrates several colorful festivals throughout the year, including:
   * Akwasidae (a celebration of water spirits)
   * Homowo (a harvest festival)
   * Odeenu (a funeral procession and reenactment of ancient rituals)
10. **Places to visit**: Some popular tourist destinations in Ghana include:
   * Cape Coast Castle (a UNESCO World Heritage Site)
   * Elmina Castle (one of the oldest European structures on African soil)
   * Kumasi (the Ashanti capital city, known for its traditional craftsmanship and cultural heritage)
   * Kakum National Park
   * Mole National Park

These are just a few highlights about Ghana. Is there something specific you'd like to know more about?

>>> bye
It was nice chatting with you about Ghana! If you have any more questions or topics you'd like to explore, feel free to come back anytime.

Until next time, take care and have a wonderful day!

(You can also say "tsaa" in Twi, which is a common Ghanaian greeting that roughly translates to "goodbye")

```

Figure 4

```

PS C:\WINDOWS\system32> ollama pull llama3.1
pulling manifest
pulling 667b6c1932bc: 100%
pulling 948af27d3fc7: 100%
pulling 0ba8f0e314b4: 100%
pulling 56bb8bd477a5: 100%
pulling 455f34728c9b: 100%
verifying sha256 digest
writing manifest
success
PS C:\WINDOWS\system32> ollama run llama3.1
>>> hi
Hello! How are you today? Is there something I can help you with or would you like to chat?

>>> what is your name?
I don't have a personal name, but I'm an AI designed to assist and communicate with users in a friendly way. You can think of me as "Assistant" or just call me by the default name "LLaMA" (which stands for Large Language Model Application). How about we go with "Assistant" for now?

>>> what do you know about ghane?
You're interested in Ghana!

Ghana is a beautiful country located in West Africa, bordered by six countries: Burkina Faso to the north, Côte d'Ivoire to the west, Togo to the east, and La Côte d'Ivoire also shares an international border with the Ivory Coast. To the south lies the Gulf of Guinea, which is part of the Atlantic Ocean.

Here are a few interesting facts about Ghana:

1. Ancient History: Ghana was once a powerful empire that existed from around 300 to 1200 AD.
2. Gold Rush: Ghana's name comes from the Portuguese word "Gana," meaning gold. The country was known for its rich gold deposits, and it was a major reason why European colonizers were interested in the region.
3. Cultural Heritage: Ghana has a rich cultural heritage, with over 100 ethnic groups speaking more than 100 languages. The most prominent language is Akan, which includes languages like Twi and Fante.
4. Beautiful Landscapes: Ghana's landscapes are diverse, ranging from savannas to tropical rainforests, and even mountains (e.g., Mount Afadjato).
5. Cuisine: Ghanaian cuisine is delicious! Some popular dishes include fufu (boiled cassava dough), banku (fermented corn dough), and groundnut soup.
6. Music and Dance: Ghana has a vibrant music scene, with genres like highlife, azonto, and hiplife. Traditional dances are also an integral part of the culture.

What specific aspect of Ghana would you like to know more about?

>>>
PS C:\WINDOWS\system32> |

```

Figure 5

Project Name: Government of Ghana's Chatbot

Purpose: To provide accessible information retrieval from Ghana government documents using RAG

Technology Stack: Python, HTML, CSS, JavaScript, FastAPI, Ollama/Llama, Phi3:mini, ChromaDB, Selenium

This is the architecture of the entire system

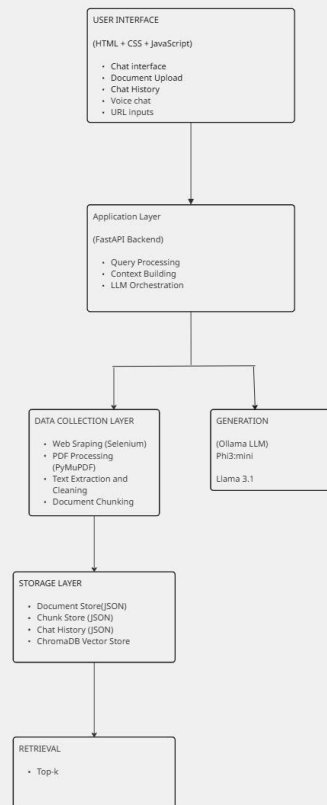


Figure 6

GH Ghana Gov Chatbot



Ghana Climate Innovation Centre (GCIC) | SNV

● Backend offline

● No page analyzed



Analyze This Page



Clear Context



Welcome!

Click "Analyze This Page" to extract content from the current webpage, then ask questions about it.

You can also ask general questions about Ghana government information.

Ask a question about this page or Ghana government...



Figure 7