

INTELLIGENT MULTILINGUAL CONVERSATIONAL AI WITH INTEGRATED SPEECH RECOGNITION AND SYNTHESIS

Shaik Neha Afreen
Independent Researcher

ABSTRACT

With the rapid growth of Artificial Intelligence, conversational systems have become an essential part of modern applications. This project presents a multilingual chatbot that supports both text and voice interaction. The system is designed to understand user input in multiple languages and generate meaningful responses using a large language model.

The chatbot integrates a backend powered by an advanced language model API to process user queries. It supports multiple languages such as English, Telugu, Hindi, and others. Additionally, the system includes voice capabilities where user input can be converted from speech to text, and responses are converted back into speech using text-to-speech technology.

The application is developed using Streamlit for the user interface, ensuring an interactive and user-friendly experience. This solution enhances accessibility and usability by allowing users to communicate in their preferred language and mode (text or voice).

Keywords:

Multilingual Chatbot, Artificial Intelligence, Natural Language Processing, Speech Recognition, Text-to-Speech, Streamlit, Large Language Models

INTRODUCTION

In today's digital world, chatbots play a significant role in automating communication between users and systems. Traditional chatbots are often limited to a single language and text-based interaction, which restricts their usability for a diverse audience.

This project introduces a multilingual chatbot that overcomes these limitations by supporting multiple languages and voice interaction. By leveraging Natural Language Processing (NLP) and large language models, the system can understand user queries and generate relevant responses.

The addition of speech recognition and text-to-speech capabilities makes the chatbot more interactive and user-friendly. This is particularly useful for users who prefer speaking over typing or those with limited literacy in a particular language.

OBJECTIVES

The primary objectives of this project are:

- To develop a chatbot capable of understanding and responding in multiple languages
- To integrate voice input using speech-to-text technology
- To generate voice output using text-to-speech conversion
- To maintain chat history for better conversational context
- To create a user-friendly interface using Streamlit
- To provide accurate and meaningful responses using a large language model

PROPOSED WORK

The proposed system is a voice-enabled multilingual chatbot designed to facilitate seamless communication between users and AI systems.

The system accepts user input either through text or voice. If the input is provided as speech, it is first converted into text using speech recognition techniques. The processed text is then sent to a backend language model, which generates an appropriate response based on the input and conversation history.

The generated response is displayed on the interface and optionally converted into speech using text-to-speech technology. The chatbot supports multiple languages by guiding the model through prompt engineering techniques.

This system improves accessibility, enhances user interaction, and provides a flexible communication platform.

METHODOLOGY

The system follows a structured workflow:

- The user provides input through text or voice
- If voice input is used, it is converted into text using speech recognition
- The text is processed along with previous chat history
- A system prompt ensures responses are generated in the selected language
- The backend model generates a context-aware response
- The response is displayed and optionally converted into speech

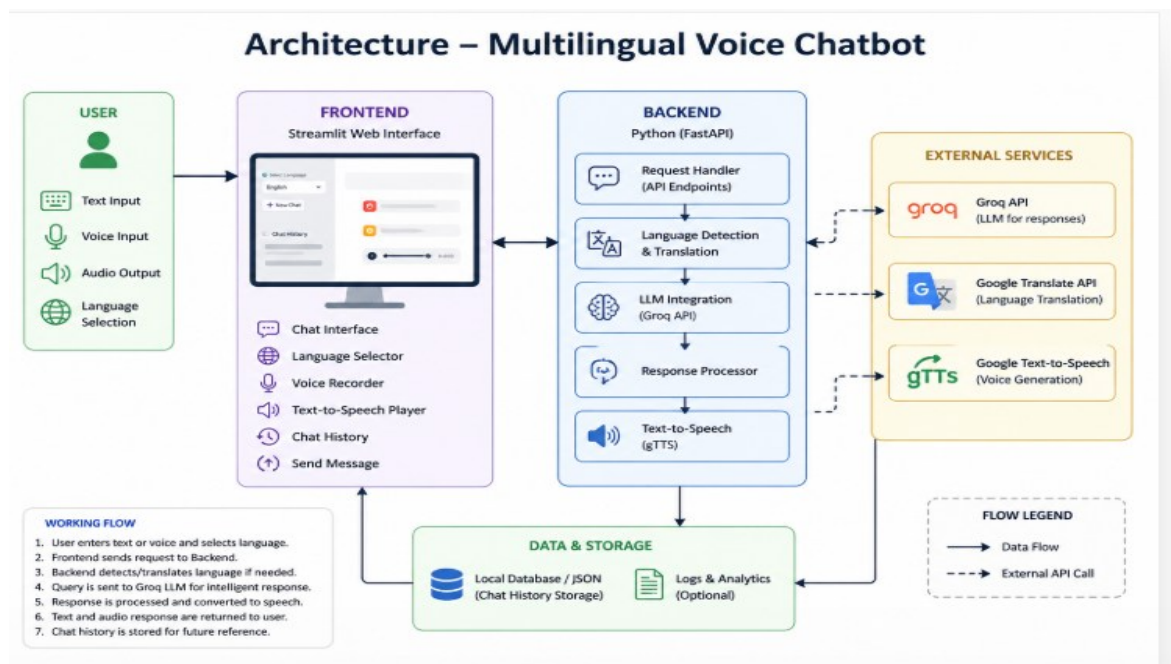


Figure 1.1 Architecture for Multilingual Chatbot

The architecture of the multilingual voice chatbot begins with the user providing input through text or voice and selecting a preferred language. The frontend, built using Streamlit, captures this input and sends it to the backend for processing. The backend handles the request and forwards it to a large language model through the Groq API to generate an appropriate response. If voice input is given, it is first converted into text using speech recognition. The generated response is then converted into audio using text-to-speech technology. Finally, both text and voice outputs are displayed to the user, while chat history is stored for future interactions.

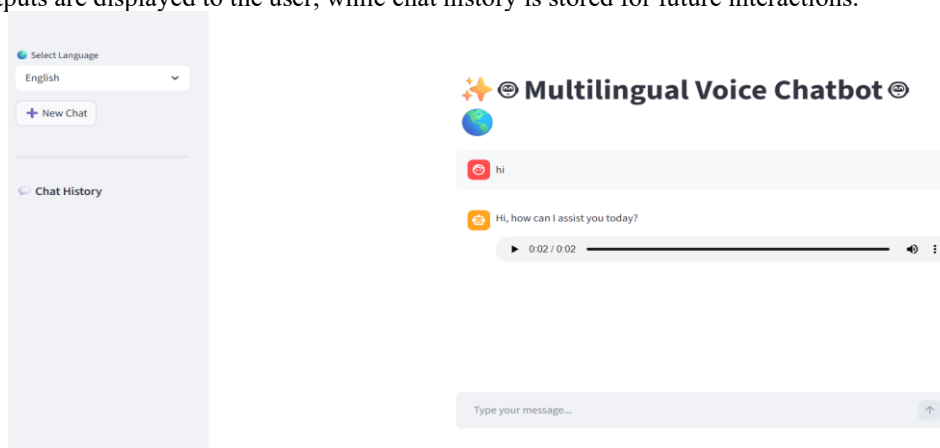


Figure 1.2 Interface

IJETRM

International Journal of Engineering Technology Research & Management (IJETRM)

Journal Article

<https://ijetrm.com/issue/>

This interface represents a multilingual voice chatbot developed using Streamlit, where users can interact through text input in a simple and user-friendly environment. The system processes user queries using a large language model and generates meaningful responses in real time. It supports multiple languages, allowing users to communicate in their preferred language without limitations. Additionally, the chatbot enhances interaction by converting responses into audio using text-to-speech technology. Overall, the application provides an efficient and interactive conversational experience by combining both text and voice capabilities.

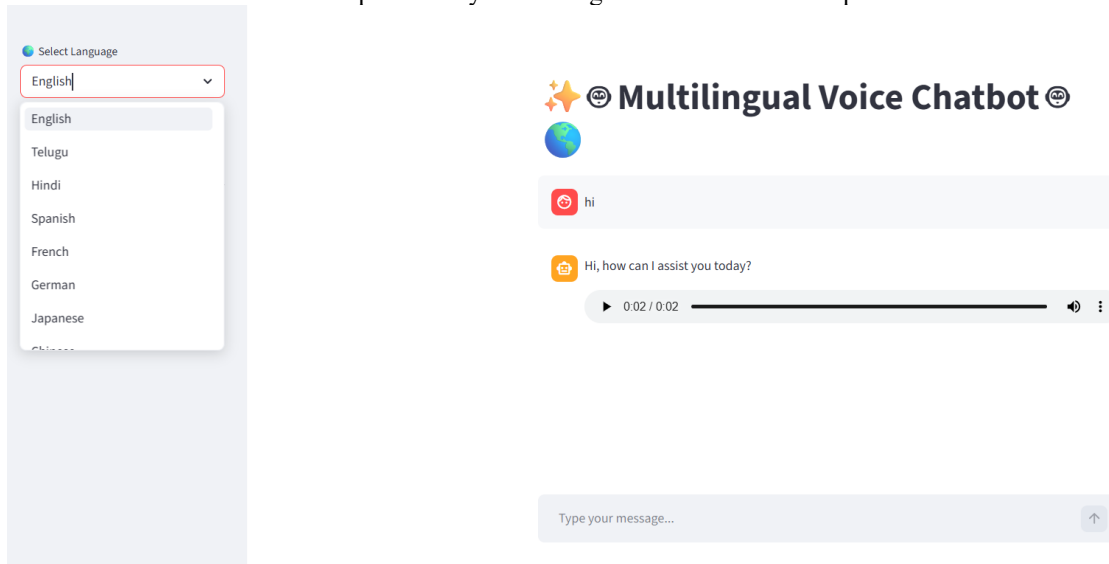


Figure 2.1 Language selection Successfully

This interface shows the language selection feature of the multilingual voice chatbot, where users can choose their preferred language from the dropdown menu. The chatbot supports multiple languages such as English, Telugu, Hindi, Spanish, and others, enabling flexible communication. Once a language is selected, the system generates responses in that chosen language, ensuring a personalized user experience. The interaction displayed includes a user message and the chatbot’s reply along with voice output. This functionality enhances accessibility by allowing users to communicate comfortably in different languages..

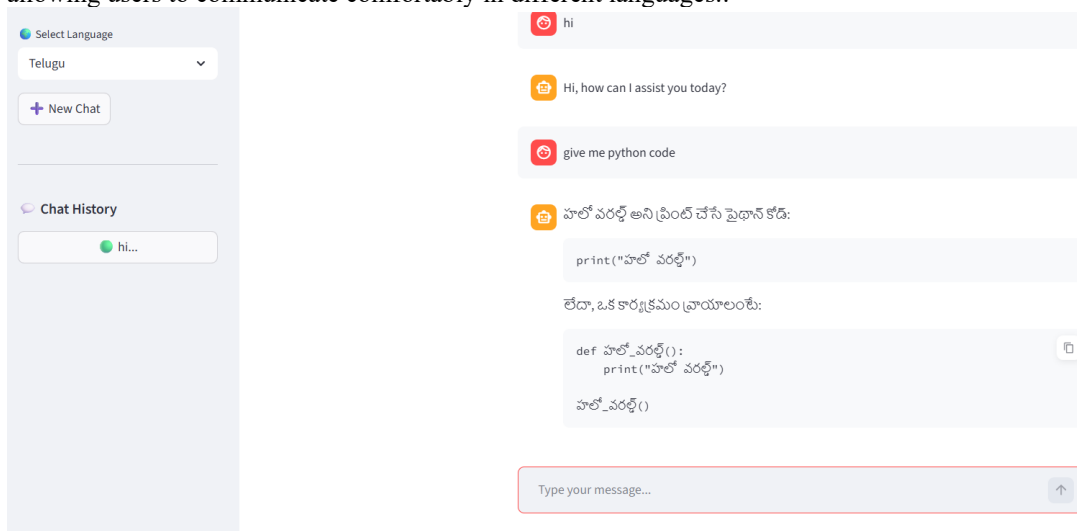
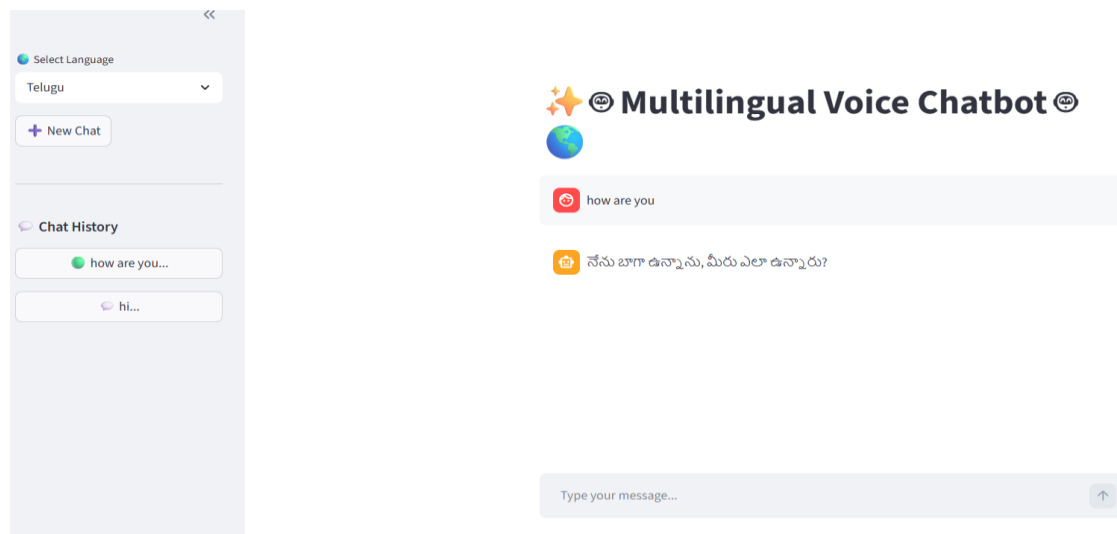


Figure 2.2 Coding

This interface demonstrates the multilingual capability of the chatbot, where the selected language is Telugu. The user provides a query in English, and the system intelligently generates the response in Telugu, showing its ability to adapt across languages. It also supports code generation, as seen in the Python example provided within the chat. The chatbot maintains conversation flow while delivering both explanations and executable code snippets. This highlights the system’s effectiveness in handling multilingual queries along with technical assistance.

**Figure 3.1 Final Predictions**

This interface highlights additional features of the multilingual chatbot, including chat history management and UI controls. The chat history section on the left stores previous conversations, allowing users to quickly revisit and continue earlier discussions. Each chat is saved with a preview, making it easy to identify past interactions. The minimize (collapse) option helps in hiding the sidebar, providing a cleaner and wider chat view for better readability. The system also maintains context by switching between chats seamlessly without losing data. Overall, these features improve usability, organization, and user experience while interacting with the chatbot.

System Performance and Observation

The developed multilingual voice chatbot was evaluated to assess its performance in handling text and voice-based interactions across multiple languages. The system was tested using different types of user inputs, including English, Telugu, Hindi, and mixed-language queries.

During testing, the chatbot successfully generated relevant and context-aware responses using the large language model backend. The integration of chat history allowed the system to maintain conversational continuity, improving the quality of responses over multiple interactions.

For voice-based interaction, the speech-to-text module was tested with various audio inputs. The system was able to convert speech into text with reasonable accuracy when the audio was clear and the language was properly specified. It was observed that providing language-specific parameters significantly improved recognition accuracy, especially for regional languages such as Telugu and Hindi.

The text-to-speech module effectively converted chatbot responses into audio output. The generated speech was clear and understandable for most supported languages, enhancing user experience by enabling hands-free interaction.

The system demonstrated stable performance in real-time interaction scenarios. However, minor variations in response time were observed depending on network conditions and API response latency. Additionally, speech recognition accuracy was influenced by factors such as background noise, pronunciation clarity, and recording duration.

Overall, the chatbot performed effectively in delivering multilingual conversational support through both text and voice. The results indicate that the integration of large language models with speech processing techniques can significantly improve user interaction and accessibility in conversational AI systems.

ACKNOWLEDGEMENT

I would like to express my sincere appreciation for the opportunity to work on this project, which has allowed me to explore and apply advanced concepts in Artificial Intelligence and Natural Language Processing.

This project was carried out as a self-driven effort, and it provided valuable hands-on experience in building a real-world multilingual voice chatbot system. Throughout the development process, I gained deeper insights into machine learning models, speech processing, and user interface design.

I would also like to acknowledge the contributions of the open-source community and platforms such as Streamlit, Groq API, and various Python libraries, which made the development and implementation of this project possible.

IJETRM

International Journal of Engineering Technology Research & Management (IJETRM)

Journal Article

<https://ijetrm.com/issue/>

Finally, I am grateful for the continuous motivation and support from my family and peers, which encouraged me to successfully complete this work.

CONCLUSION

The multilingual voice chatbot developed in this project demonstrates how artificial intelligence can enhance human-computer interaction. By combining natural language processing, speech recognition, and text-to-speech technologies, the system provides a flexible and interactive communication platform.

This solution allows users to interact in their preferred language and mode, making it highly accessible and user-friendly. Future improvements can focus on enhancing speech accuracy, supporting more languages, and integrating real-time voice conversation capabilities.

REFERENCES

- [1] OpenAI
GPT-based Language Models Documentation. Available: <https://platform.openai.com/docs>
- [2] MetaAI
LLaMA: Large Language Model Research Paper. Available: <https://ai.meta.com/llama/>
- [3] Groq
Groq API Documentation. Available: <https://console.groq.com/docs>
- [4] Google
Google Speech Recognition API. Available: <https://cloud.google.com/speech-to-text>
- [5] Google
Google Text-to-Speech (gTTS) Library Documentation. Available: <https://pypi.org/project/gTTS/>
- [6] Streamlit
Streamlit Documentation. Available: <https://docs.streamlit.io>
- [7] Jurafsky and Martin
Speech and Language Processing. Pearson Education, 3rd Edition
- [8] Ian Goodfellow, Yoshua Bengio, Aaron Courville
Deep Learning. MIT Press, 2016
- [9] Hugging Face
Transformers Documentation. Available: <https://huggingface.co/docs>
- [10] Natural Language Processing
Research articles and online resources related to multilingual conversational AI systems