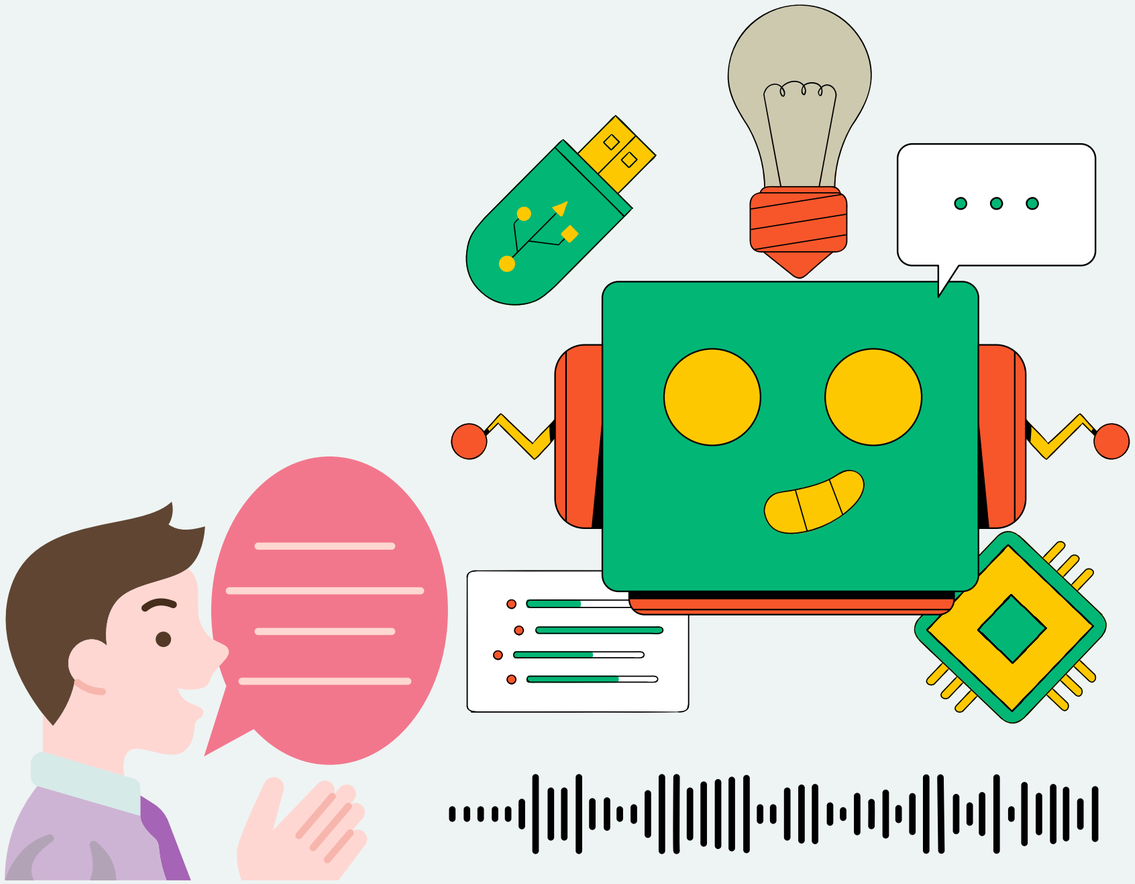


VOICE POWERED CHATBOT

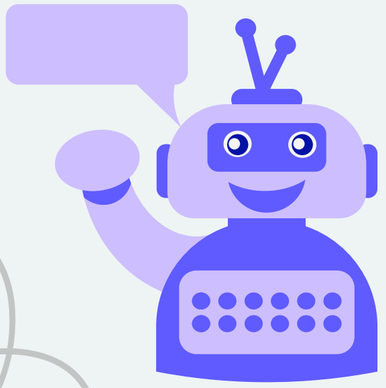
Group 14

TEAM MEMBERS

Aiswarya Satheesh	AM.EN.U4AIE21007
Nandita Sangeeth	AM.EN.U4AIE21046
Parvathi PK	AM.EN.U4AIE21050
Swathy Rajesh	AM.EN.U4AIE21063
Vaishnavi C M	AM.EN.U4AIE21065

TABLE OF CONTENTS

INTRODUCTION



Speech processing: Analysis and manipulation of spoken language by computers.

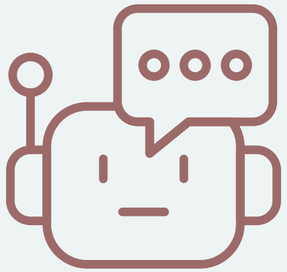
Voice chatbots: Intuitive way for users to interact with technology, mimicking human conversation.

Offer quick response to user queries, reducing wait times compared to traditional customer service methods.

Integrate AI, speech recognition and speech technologies for a voice-powered chatbot using GPT-4-turbo model.

LITERATURE REVIEW

- Significant advancements in voice recognition, like Google's speech-to-text.
- Evolution of NLP with GPT models from GPT-2 to GPT-4.
- Enhancements in Text-to-Speech (TTS) systems, improving synthetic voice quality.
- Effectiveness of interactive AI systems, such as mental health chatbots.
- Overall transformation of user interaction across various sectors.



METHODOLOGY

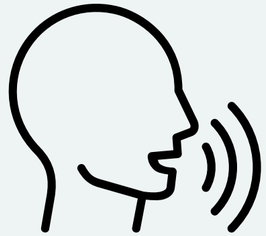
- Speech Recognition
- User Interface with Streamlit
- Text to Speech(TTS)
- Thread Management
- Integration with OpenAI’s GPT Models
- Error Handling

SPEECH RECOGNITION

```python
recognizer = sr.Recognizer()

def transcribe_audio_to_text(language_code):
    with sr.Microphone() as source:
        st.write("Say something...")
        audio = recognizer.listen(source)
        try:
            text = recognizer.recognize_google(audio, language=language_code)
            st.write(f"You said: {text}")
            return text
        except sr.UnknownValueError:
            st.write("Sorry, I did not understand that.")
            return ""
        except sr.RequestError:
            st.write("Sorry, the service is down.")
            return ""
```

- Uses the speech_recognition library's Recognizer and Microphone classes.
- Captures user speech through the microphone and converts it into text using Google’s speech recognition API.
- Implements error handling to manage unclear audio and service interruptions.



TEXT TO SPEECH(TTS)

```python
tts_engine = pyttsx3.init()

# Supported languages for transcription and TTS
supported_languages = {
    "English": "en",
    "Malayalam": "ml"
}

def speak_text(text):
    def run():
        tts_engine.say(text)
        tts_engine.runAndWait()
    # Ensure each speech runs in a new thread
    threading.Thread(target=run).start()
```

- Employs the pyttsx3 library.
- Converts chatbot textual responses back into audible speech, enhancing user interaction.
- Supports accessibility, catering to users with visual impairments or those preferring auditory feedback.
- Supports multiple languages like English and Malayalam

INTEGRATION WITH OPENAI'S GPT MODELS

```python
# Set your OpenAI API key
openai.api_key = 'your-openai-api-key'

def generate_gpt_response(prompt):
    messages = [
        {"role": "system", "content": "You are a helpful assistant."},
        {"role": "user", "content": prompt},
    ]
    response = openai.ChatCompletion.create(
        model="gpt-4-turbo",
        messages=messages
    )
    return response['choices'][0]['message']['content'].strip()
```

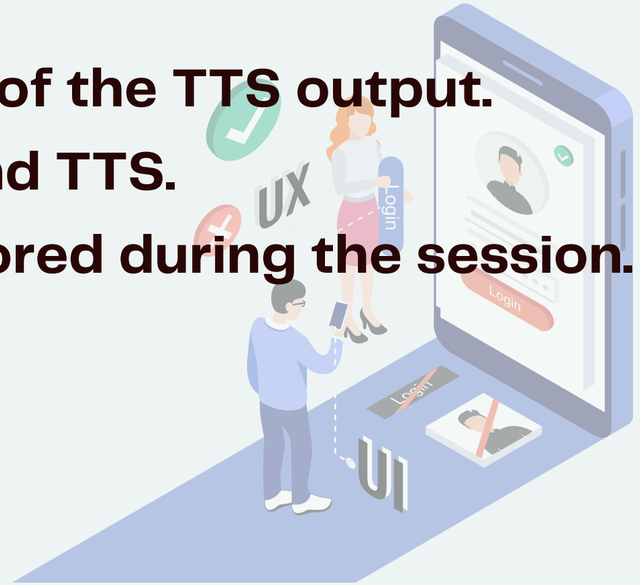
- Utilizes the openai library to send transcribed text to GPT models for generating conversational responses.
- Constructs a message array with predefined system context and user input to set the dialogue tone.
- Configures the model to function as a helpful assistant, enhancing user interaction

USER INTERFACE WITH STREAMLIT

- Built with Streamlit for ease of development and deployment.
- Features interactive components like start buttons for speech recognition and settings for TTS.
- Custom CSS is applied to improve the aesthetic appeal of the application.

FEATURES

- Captures audio in real-time and converts it into text.
- Provides a slider in the sidebar for users to adjust the volume of the TTS output.
- Allows users to select the language for speech recognition and TTS.
- Clear conversation history, providing control over the data stored during the session.
- Can review conversation history in the sidebar.
- Enables users to download conversation history as a text file.

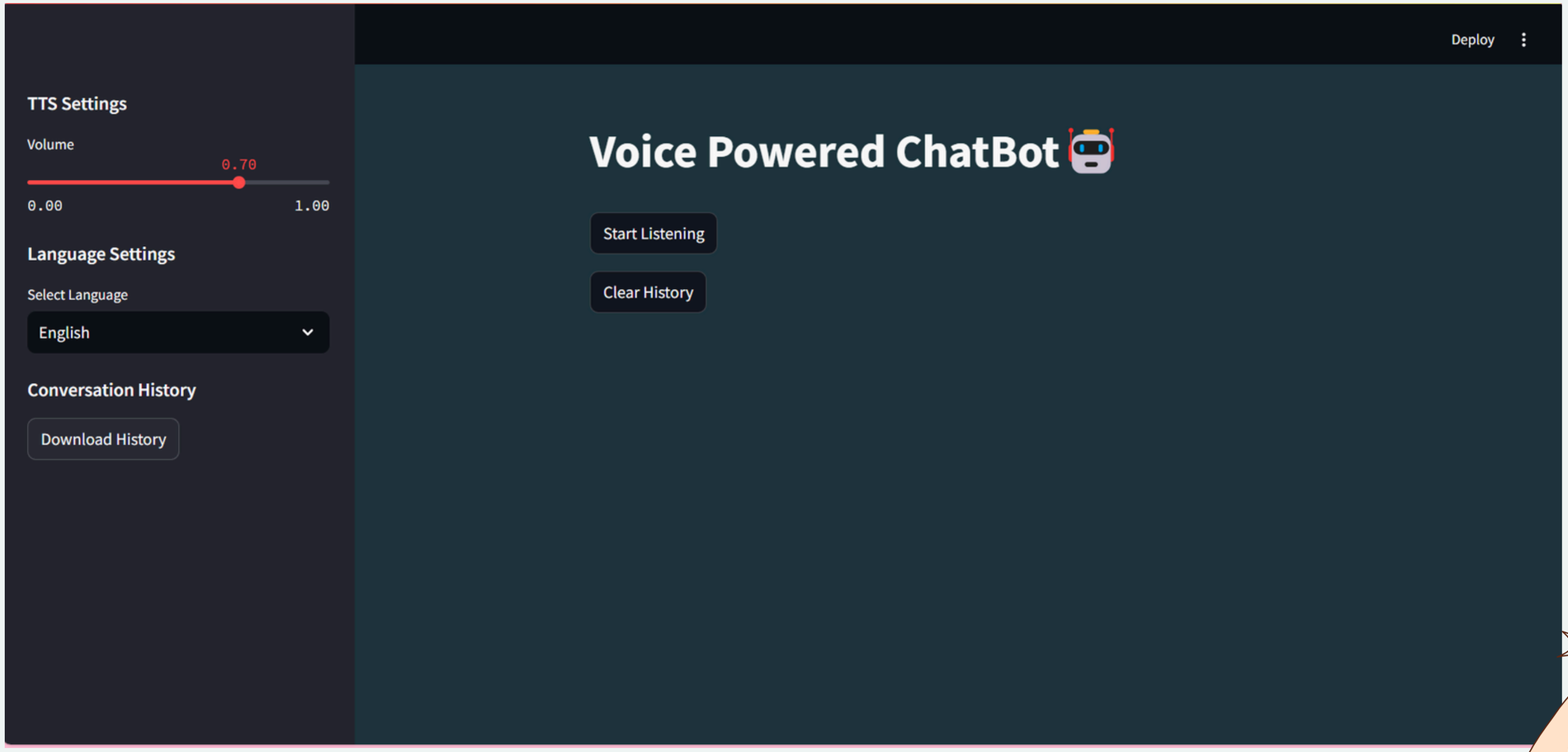


THREAD MANAGEMENT AND ERROR HANDLING

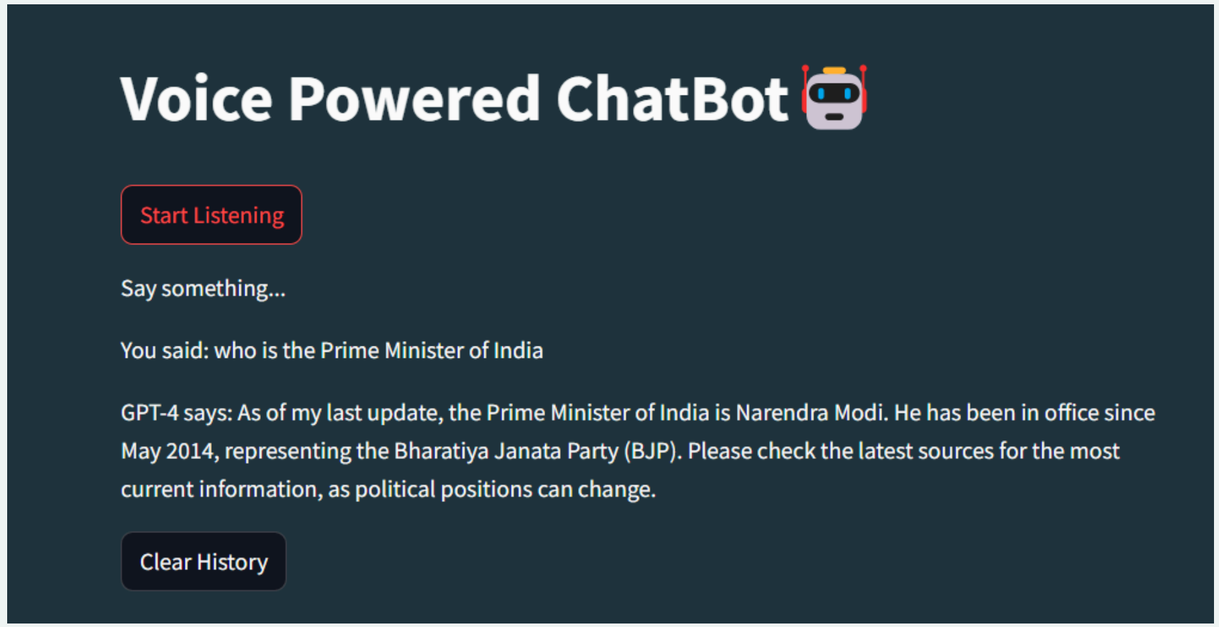
- **Non-blocking Operations: Implements Python's `threading` module to run TTS tasks in separate threads, keeping the UI responsive during speech synthesis.**
- **Concurrency Control: Manages multiple simultaneous TTS requests efficiently, preventing delays and avoiding bottlenecks during fast-paced interactions.**
- **Speech Recognition Errors: Addresses `UnknownValueError` by informing users when their speech isn't recognized, prompting them to try speaking again.**
- **Service Availability Checks: Alerts users immediately when the speech recognition service is down due to network issues or maintenance.**
- **Robust Feedback Mechanisms: Ensures continuous user engagement with clear, actionable feedback during errors in speech recognition and TTS, enhancing user trust and system transparency.**

RESULTS

USER INTERFACE

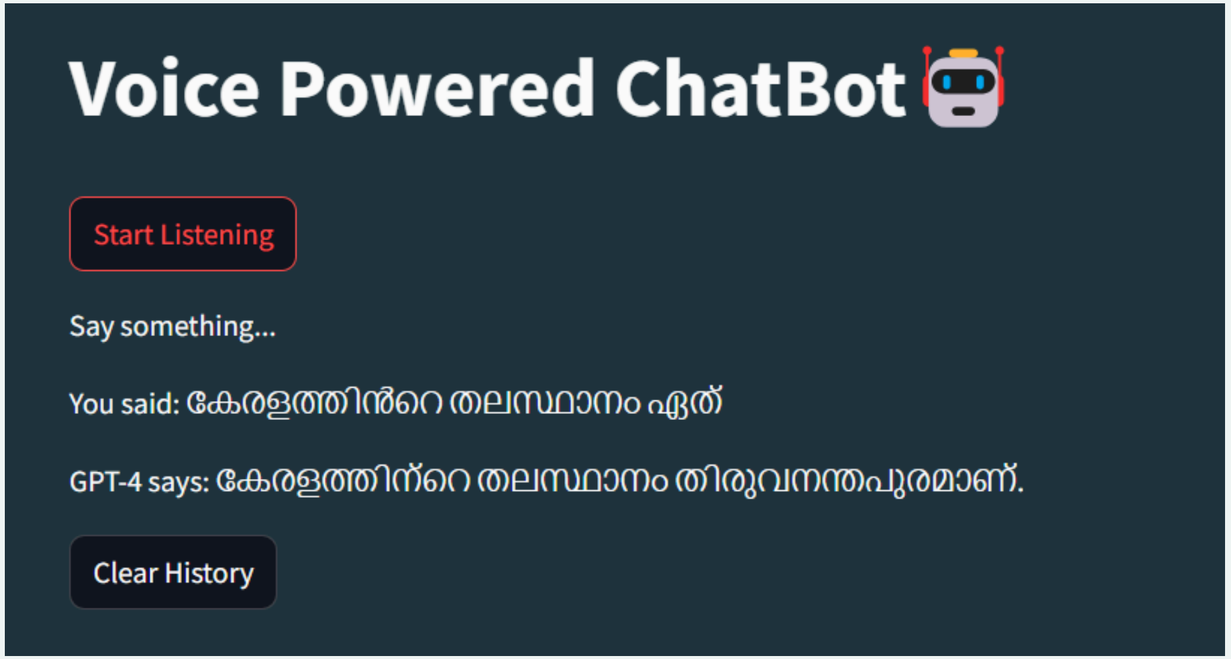


ENGLISH

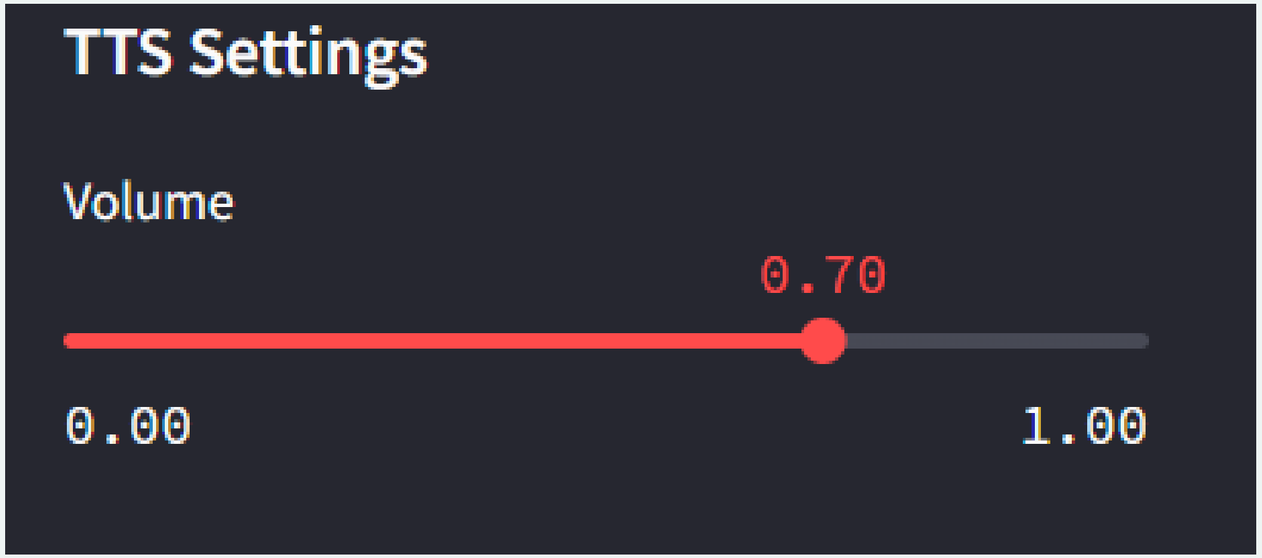


MALAYALAM

↓

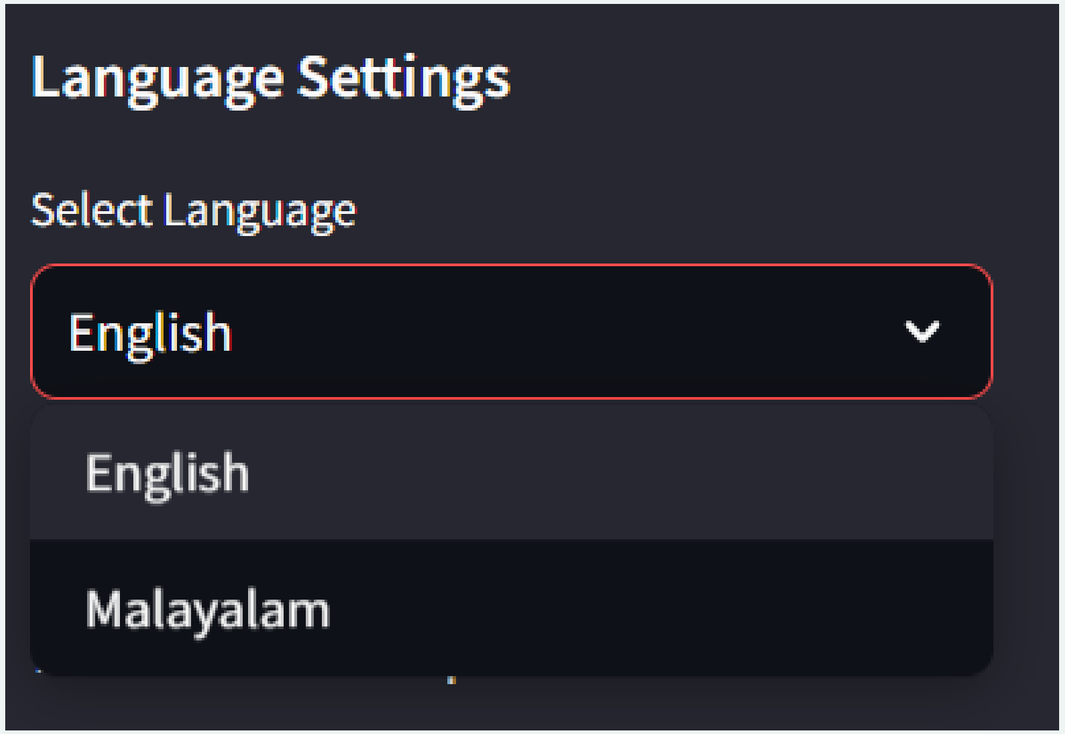


MULTIPLE LANGUAGE INPUTS AND RESPONSE IN MULTIPLE LANGUAGES



VOLUME SETTINGS

LANGUAGE SETTINGS



Conversation History

You: what is the capital of India

GPT-4: The capital of India is New Delhi.

You: who is the Prime Minister of India

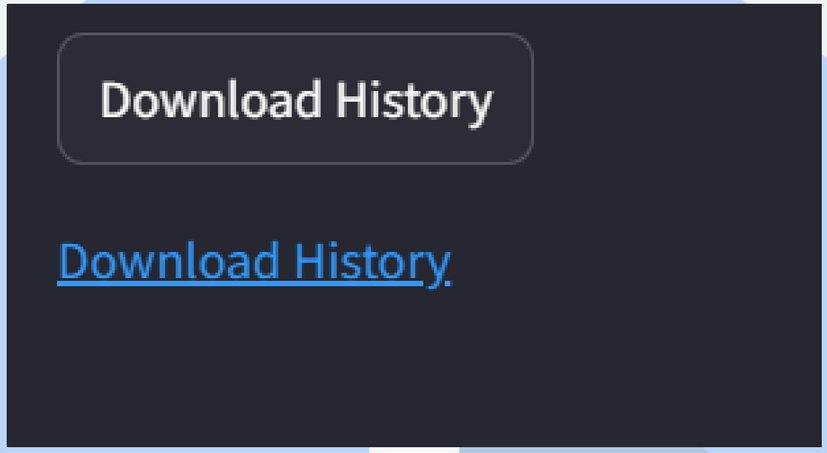
GPT-4: As of my last update, the Prime Minister of India is Narendra Modi. He has been in office since May 2014, representing the Bharatiya Janata Party (BJP). Please check the latest sources for the most current information, as political positions can change.

You: കേരളത്തിൻറെ തലസ്ഥാനം ഏത്

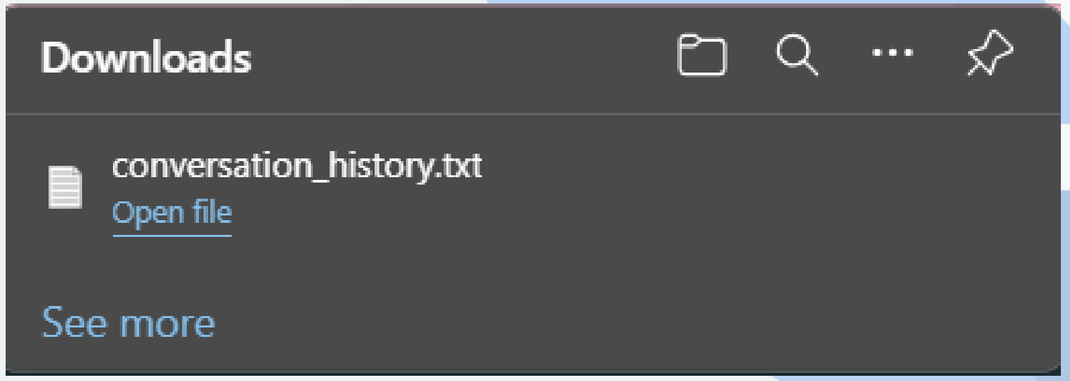
GPT-4: കേരളത്തിന്റെ തലസ്ഥാനം തിരുവനന്തപുരമാണ്.

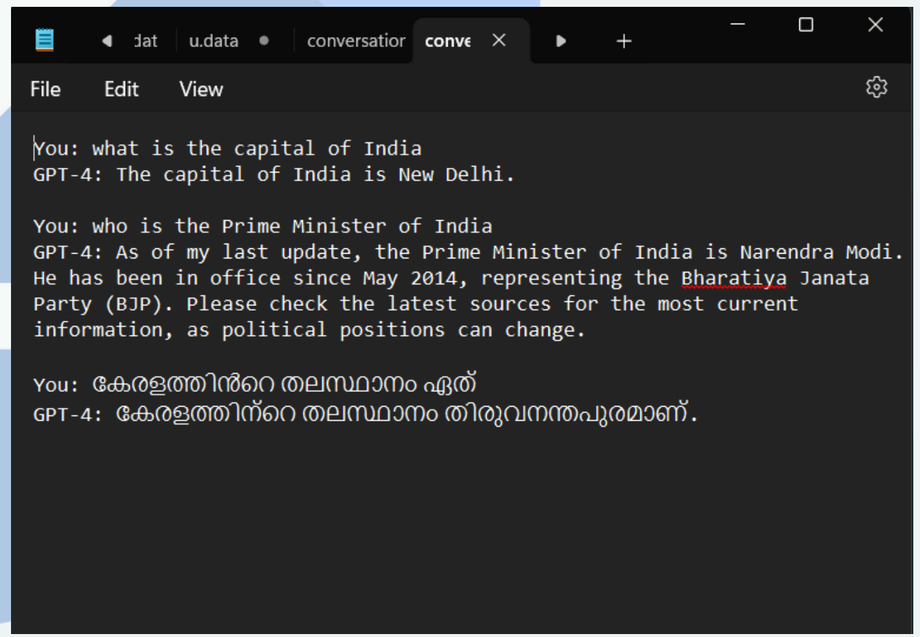
CONVERSATION HISTORY

DOWNLOAD HISTORY OPTION



DOWLOADED TEXT FILE





CONCLUSION

- The project successfully integrates speech recognition and text-to-speech technologies, enhancing user interaction and accessibility.
- Utilizing OpenAI's GPT models, the chatbot achieves a high level of conversational intelligence, effectively simulating human-like interactions.
- The chatbot boosts accessibility, empowering visually impaired users and those uncomfortable with text-based interfaces, making digital services more inclusive.
- Various applications like scheduling and information retrieval can be done, reducing cognitive burden and boosting productivity and quality of life.
- The modular design and scalable architecture of the chatbot lay a strong foundation for future enhancements and broader application possibilities.

REFERENCE

- https://www.youtube.com/watch?v=OtOTjanvMxM
- https://www.researchgate.net/publication/224564336_An_intelligent_web-based_voice_chat_bot
- https://www.researchgate.net/publication/370884435_Chatbots_and_Voice_Assistants_Digital_Transformers_of_the_Company-Customer_Interface-A_Systematic_Review_of_the_Business_Research_Literature
- https://ieeexplore.ieee.org/abstract/document/5167660

CODE LINK

https://github.com/TheAiswaryaSatheesh/Group14_Speech-Processing-Project.git

Thank You