

Overview:

The voice bot application captures speech from the microphone, processes it to generate a conversational response, converts the response to speech, and plays it back to the user. The application uses Streamlit for the web interface, Google Cloud services for storage and speech recognition, and various Python libraries for audio processing and model inference.

File Structure:

- main.py: The main Streamlit application file.
- generate_audio.py: Contains the function to convert text to speech.
- config.json: Configuration file for Google Cloud services.
- env.json: Google Application Credentials file (not included in the repository).

Functions:

generate_uuid(length: int = 8) -> str

Generates a random unique identifier.

load_config()

Loads configuration from config.json.

get_base64_of_audio(audio_path)

Converts an audio file to base64 for embedding in HTML.

autoplay_audio(audio_path)

Creates an HTML5 audio autoplay element.

upload_to_gcs(bucket_name, source_file_name, destination_blob_name)

Uploads a file to Google Cloud Storage.

recognize_speech_from_mic(unique_number)

Captures speech from the microphone, saves it as a temporary file, uploads it to Google Cloud Storage, and returns the file URI and transcription.

text_to_speech(unique_number, text)

Converts text to speech using the VitsModel and saves the output as a WAV file.

main()

The main function that sets up the Streamlit application, handles user interactions, and manages the chat history.

Configuration:

- config.json: Contains the model ID, project ID, region, and bucket name for Google Cloud services.
- env.json: Contains the Google Application Credentials.

Dependencies:

- torch
- transformers
- scipy
- streamlit
- vertexai
- speech_recognition
- google-cloud-storage

Running the Application:

1. Ensure all dependencies are installed and the configuration files are set up.
2. Run the Streamlit application using the command:
streamlit run main.py
3. Open the application in your browser and interact with the voice bot.

Troubleshooting:

- Ensure your Google Cloud credentials are correctly set up and the env.json file is in the project root directory.
- Check the configuration in config.json for any errors.
- Verify that all required packages are installed.