Logo, company name

Description automatically generated

**Submitted By:**

Veda Samhitha Dyawanapally

Vigneshwar Muriki

Deepthi Gade

Mounika Kasaragadda

**INTRODUCTION**

**Speech to Text Recognition:**

Automatic speech recognition (ASR), often known as speech to text technology, is the process of converting spoken words into written text utilizing algorithms and machine learning techniques. Many different sectors, including healthcare, banking, education, and telecommunications, use applications like virtual assistants, voice-controlled gadgets, transcription services, and dictation software. STT technology has significantly advanced as a result of advances in machine learning algorithms and natural language processing (NLP) techniques. The reliable transcription of many speakers, coping with regional accents and dialects, and recognizing non-standard speech patterns are still challenges. Despite these challenges, STT technology remains a valuable tool for a range of applications, enabling people to communicate and connect with technology in a more relaxed and natural way.

**Application areas of this Technology:**

* Siri, Alexa, and Google Assistant, among other virtual assistants, all use speech-to-text technologies.
* For meetings, interviews, and court procedures, audio and video recordings are converted into text using speech-to-text technology.
* For movies, television shows, and online videos, closed captions are produced using speech to text technology.
* Dictation software uses speech-to-text technology to let users dictate text directly into a computer.
* Speech-to-text technology is employed in the healthcare industry for medical transcription and dictation, speech therapy, and support with communication for those who are speech-impaired.
* Applications for speech to text technology are utilized in customer service, enabling businesses to provide voice-controlled self-service alternatives and to track client interactions.
* For the benefit of students with hearing problems, speech-to-text technology is utilized in the classroom to provide transcription and captioning for video and audio content.

Streamlit is a popular Python framework that simplifies the process of building interactive data science applications. To implement Speech-to-Text using Streamlit, one can leverage existing libraries such as Speech Recognition, PyAudio, and pocketsphinx. These libraries provide speech recognition capabilities that can be easily integrated into a Streamlit app. One can start by creating a simple user interface using Streamlit's interactive widgets such as sliders, buttons, and text inputs. The user can then speak into the microphone, and the speech recognition algorithm converts the speech into text, which is displayed on the Streamlit app.

**There are several additional features that can be added to a speech-to-text Streamlit application to enhance its functionality and user experience. Some of these features include:**

* Real-time recording: This feature enables the software to continuously type out speech as it is being spoken, giving the user a direct feedback loop.
* Support for several languages can increase the app's adaptability and usefulness to a wider group of users by being able to identify and transcribe voice in several different languages.
* Customized vocabulary: When transcribing for fields or businesses, such as law or medicine, using a customized vocabulary can increase the accuracy of the transcription.
* Presenter recognition: Including speaker identification features can make it easier to distinguish between different talkers in a discussion and match each speaker's words to the appropriate transcript.
* Additional formatting options: Giving users the opportunity to highlight certain words or phrases in the transcription by bolding, underlining, or other formatting techniques might assist them concentrate on the most crucial portions of the transcript.
* Controls for playback: By including playback controls like pause, rewind, and play, users will be able to conveniently browse the transcription and go over particular passages.

**Basic Code:**

The Python code demonstrates a basic Speech to Text (STT) implementation using the Speech Recognition and Streamlit modules. The code creates a simple internet application that allows users to record their voice and convert it to text using the Google speech recognition API. By pressing the "Start Recording" button, the app launches the speech recognition library and starts listening for audio input from the user's built-in microphone. The audio is then translated into text by the code, which then displays the text on the screen, using the Google speech recognition API. If there are any problems, an appropriate error message will be given. This code can serve as the foundation for the creation of more complex STT applications.

Graphical user interface, text, application

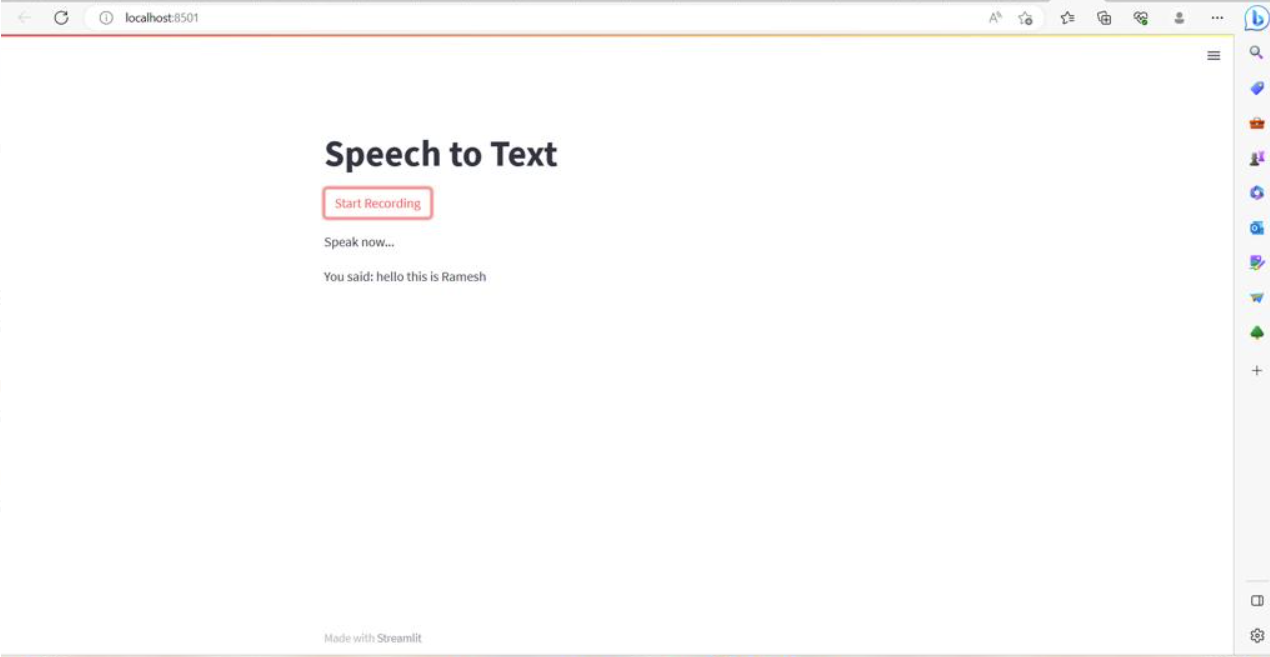
Description automatically generated

**Initial Page:**

Graphical user interface, text, application, email

Description automatically generated

**Test 1: Start recording and get correct output.**

****

**Test 2: Error check when no one talks.**

**Graphical user interface, application

Description automatically generated**

**Test 3: Network Failure**

**Graphical user interface, text, application, email

Description automatically generated**

**Feature1: Download the output transcription text**

The revealed web application for text-to-speech conversion is Python script. A user interface is made using the speech Recognition and Streamlit libraries. The speech transcription feature and the text file export feature are the two key features of the program. The software listens for audio input from the built-in microphone when the user presses the "Start Recording" button, and then it uses the Google voice recognition API to transcribe the audio. The application shows the text in the Streamlit interface if the transcription is successful. The program exports the transcription to a text file and sends it to the user's browser if the user hits the "Export Transcription" button. The statement if \_\_name\_\_ == '\_\_main\_\_' makes sure that the app () function is called.

**Output:**

Graphical user interface, text, application, email

Description automatically generated

**Feature2: Highlight\_text in Yellow**

The revealed A web application for text-to-speech conversion is Python script. A user interface is made using the Speech Recognition and Streamlit libraries. The software listens for audio input from the built-in microphone when the user presses the "Start Recording" button, and then it uses the Google voice recognition API to transcribe the audio. The script also inserts HTML code to alter the background color of specific terms in the transcription text to yellow, such as "important," "highlight," and "focus." The app displays the highlighted text in the Streamlit interface if the transcription is successful. The app displays the relevant message if the speech cannot be transcribed or if there is a problem processing the request. The statement if \_\_name\_\_ == '\_\_main\_\_' makes guarantee.

**Output:**

Graphical user interface, text, application, email

Description automatically generated

**Feature 3:** **Detection of sensitivity of the output text using Sentiment Analysis.**

The program is a Streamlit app that converts spoken words into text and analyzes sentiments in the text as it is being transcribed. The program uses the Text Blob library to assess the sentiment of the transcribed text and the speech Recognition library to translate the audio. The application asks the user to talk while it checks the built-in microphone for audio input. The app displays the transcript of the text along with the findings of the sentiment analysis. An appropriate error message is displayed if the voice recognition procedure fails.

**Output: Positive**

Graphical user interface, text, application, email

Description automatically generated

**Output: Negative**

Graphical user interface, text, application, email

Description automatically generated

**Feature4: Edit the Output Transcription Text**

This straightforward Python code generates a speech-to-text application using the Speech Recognition and Streamlit modules. The program begins hearing the user's voice through the built-in microphone as soon as they click the "Start Recording" button. The program converts the audio to text using the Google speech recognition API when the user has finished speaking. Using the Streamlit text\_input method, the transcription is shown in a text input box. An appropriate error message is presented whenever a transcription mistake is made.

**Initial Output:**

Graphical user interface, application

Description automatically generated**Final Output After Editing:**

Graphical user interface, application

Description automatically generated