

PROJECT

Report on

Speech-to-text-Conversion

by

Udit Ranjan (202410116100229)

Suchita Singh (202410116100212)

Shyam Sundar (2024101161100207)

Shweta Patel (202410116100206)

Session:2024-2025 (II Semester)

Under the supervision of

Ms. Komal Salgotra
Assistant Professor

KIET Group of Institutions, Delhi-NCR, Ghaziabad



DEPARTMENT OF COMPUTER APPLICATIONS
KIET GROUP OF INSTITUTIONS, DELHI-NCR,
GHAZIABAD-201206
(2024- 2025)

TABLE OF CONTENTS

S. No	Section	Page Number
	Table of Contents	i
1.	Introduction	1
2.	Methodology	2
3.	Code	3-5
4.	Outcomes	6-8
5.	Conclusion	9

CHAPTER – 01

INTRODUCTION

This project demonstrates a robust **speech-to-text conversion system** implemented in Python, capable of handling both **pre-recorded audio file input** and **real-time microphone capture**. In an era where voice interaction is becoming increasingly prevalent, this system lays the groundwork for building efficient transcription services that can be applied in various domains such as:

- **Voice Note-Taking**
- **Speech-Based Virtual Assistants**
- **Automated Transcription Services**

By leveraging powerful open-source libraries and Google's Web Speech API, the project ensures both **flexibility** and **accuracy** in converting spoken language to written text.

Key Objectives

- **Dual Input Modes:** Support for both uploaded audio files and live microphone input.
- **Format Conversion:** Automatically converts common audio formats (MP3, OGG, etc.) to WAV for processing.
- **Accurate Transcription:** Utilizes Google's Web Speech API to ensure reliable speech recognition.
- **Robust Error Handling:** Manages real-world issues such as unclear speech, silent input, and API failures.
- **User-Friendly Interface:** Offers a clean and minimal Command-Line Interface (CLI) for interaction.

CHAPTER – 02

METHODOLOGY

Module 1: File-Based Audio Transcription

1. Upload & Format Detection

- Uploads any audio file using Google Colab's files. Upload().
- Automatically detects the file format via filename extension.

2. Conversion to WAV

- Uses pydub and ffmpeg to convert the audio into .wav format, compatible with speech_recognition.

3. Transcription

- Loads the .wav file using sr.AudioFile().
- Processes and transcribes it via recognize_google() method.

4. Error Handling

- Identifies audio issues (no speech, silence).
- Handles API connectivity problems.

Module 2: Real-Time Microphone Transcription

1. Microphone Selection

- CLI menu offers microphone-based speech input.

2. Capture and Convert

- Listens via system microphone with ambient noise calibration.
- Transcribes the captured speech using the same API.

3. Robust Error Handling

- Provides clear feedback on audio clarity or internet access issues.

CHAPTER – 03

CODE IMPLEMENTATION

```
!pip install SpeechRecognition pydub
!apt-get install ffmpeg

from google.colab import files
uploaded = files.upload()

import speech_recognition as sr
from pydub import AudioSegment
import os

# Initialize recognizer
recognizer = sr.Recognizer()

# Get the uploaded file name
file_name = list(uploaded.keys())[0]
audio_format = file_name.split('.')[-1]

# Convert to WAV if necessary
if audio_format != 'wav':
    sound = AudioSegment.from_file(file_name, format=audio_format)
    sound.export("converted.wav", format="wav")
    audio_path = "converted.wav"
else:
    audio_path = file_name
```

Transcribe

```
with sr.AudioFile(audio_path) as source:
    audio_data = recognizer.record(source)
    print("Converting audio to text...")
    try:
        text = recognizer.recognize_google(audio_data)
        print("\n Transcribed Text:\n", text)
    except sr.UnknownValueError:
        print("Could not understand the audio.")
    except sr.RequestError as e:
        print(f' Could not connect to the API: {e}')
```

!pip install SpeechRecognition pydub

```
import speech_recognition as sr
from pydub import AudioSegment
import os
```

Initialize Recognizer

```
recognizer = sr.Recognizer()
```

Menu (Only Microphone Option)

```
print("Choose an option:")
print("1. Use Microphone")
choice = input("Enter 1: ")
```

```
if choice == "1":
```

Speech from Microphone

```
try:
    with sr.Microphone() as source:
```

```
print("\nListening... Please speak clearly.")
recognizer.adjust_for_ambient_noise(source)
audio = recognizer.listen(source)
print("Converting speech to text...")
```

```
# Google Web Speech API
```

```
text = recognizer.recognize_google(audio)
print("\n You said:\n", text)
```

```
except sr.UnknownValueError:
```

```
    print(" Could not understand the audio.")
```

```
except sr.RequestError as e:
```

```
    print(f" Could not connect to the API: {e}")
```

```
except Exception as e:
```

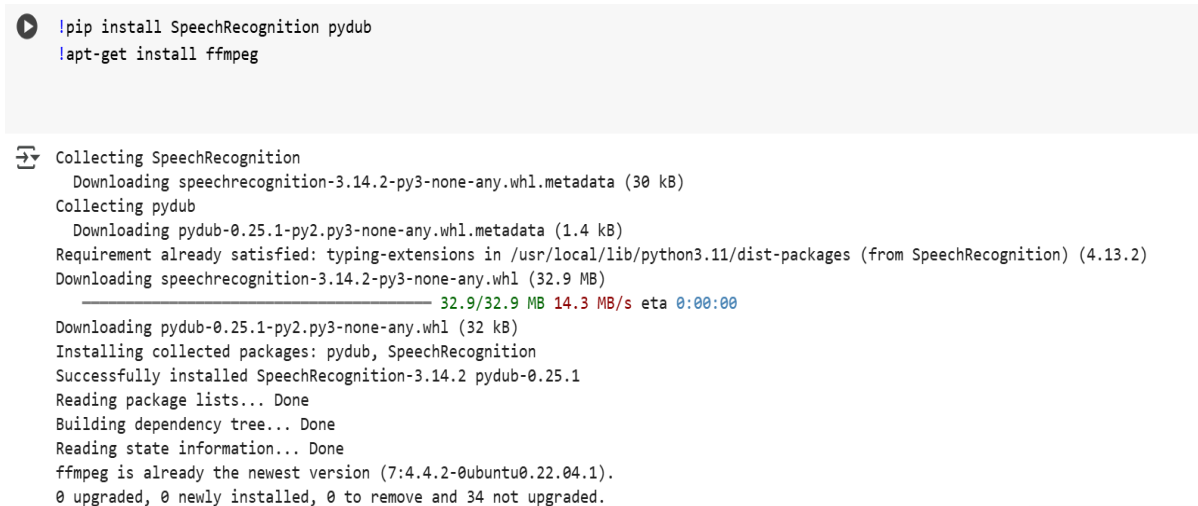
```
    print(" Error:", e)
```

```
else:
```

```
    print("Invalid choice. Please enter 1.")
```

CHAPTER – 04

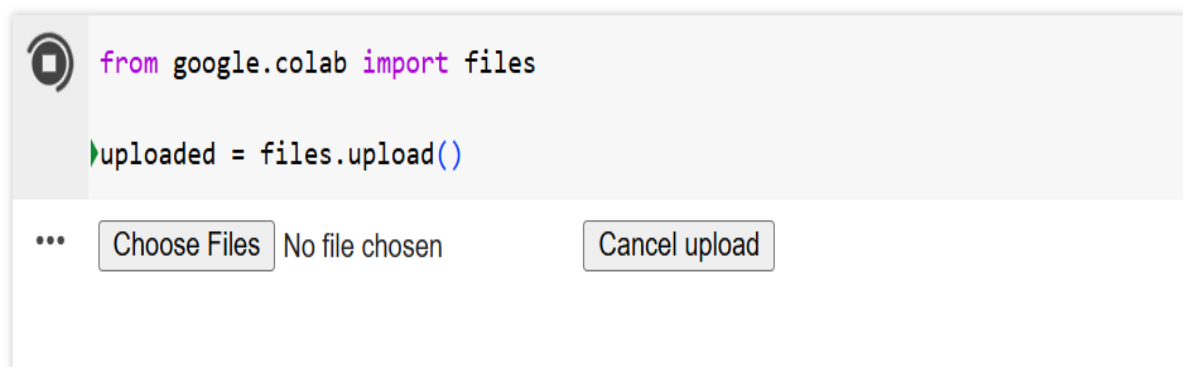
OUTCOMES



```
|pip install SpeechRecognition pydub
|apt-get install ffmpeg

Collecting SpeechRecognition
  Downloading speechrecognition-3.14.2-py3-none-any.whl.metadata (30 kB)
Collecting pydub
  Downloading pydub-0.25.1-py2.py3-none-any.whl.metadata (1.4 kB)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.11/dist-packages (from SpeechRecognition) (4.13.2)
Downloading speechrecognition-3.14.2-py3-none-any.whl (32.9 MB)
  32.9/32.9 MB 14.3 MB/s eta 0:00:00
Downloading pydub-0.25.1-py2.py3-none-any.whl (32 kB)
Installing collected packages: pydub, SpeechRecognition
Successfully installed SpeechRecognition-3.14.2 pydub-0.25.1
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ffmpeg is already the newest version (7:4.4.2-0ubuntu0.22.04.1).
0 upgraded, 0 newly installed, 0 to remove and 34 not upgraded.
```

Fig. 4.1



```
from google.colab import files

uploaded = files.upload()
```

... Choose Files No file chosen Cancel upload

Fig. 4.2

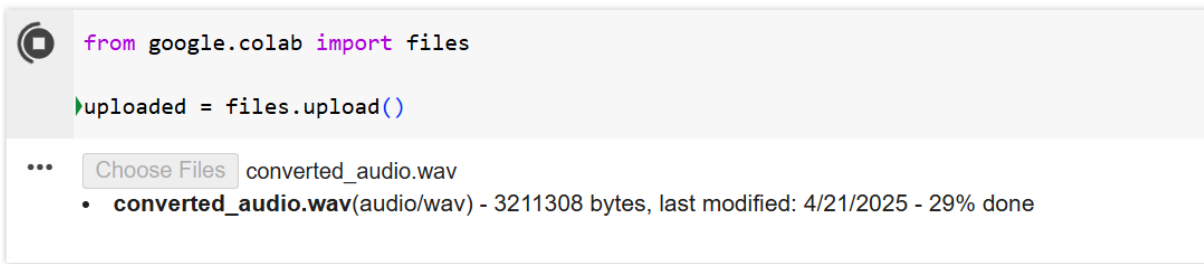


Fig. 4.3

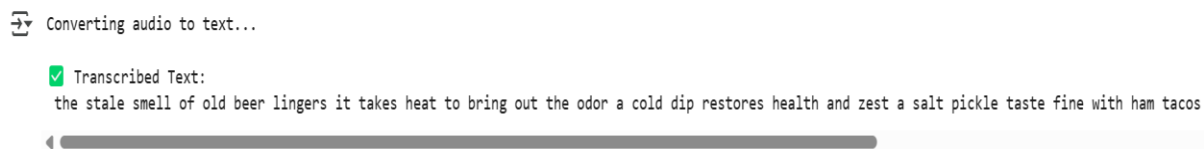


Fig. 4.4

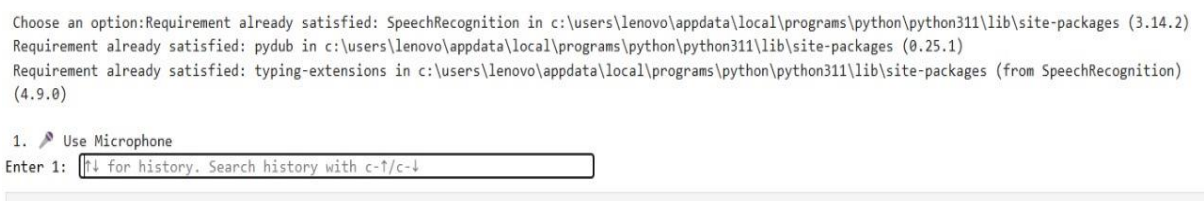


Fig. 4.5

```
Requirement already satisfied: speechrecognition in c:\users\lenovo\appdata\local\programs\python\python311\lib\site-packages (0.14.4)
Requirement already satisfied: pydub in c:\users\lenovo\appdata\local\programs\python\python311\lib\site-packages (0.25.1)
Requirement already satisfied: typing-extensions in c:\users\lenovo\appdata\local\programs\python\python311\lib\site-packages (from SpeechRecognition)
(4.9.0)
Choose an option:
1. 🎤 Use Microphone
Enter 1: 1

Listening... Please speak clearly.
Converting speech to text...
```

Fig. 4.6

```
Choose an option:
1. 🎤 Use Microphone
Enter 1: 1

Listening... Please speak clearly.
Converting speech to text...

✅ You said:
how was the day
```

Fig. 4.7

CHAPTER – 05

CONCLUSION

This project presents a practical and efficient implementation of a speech-to-text system using Python, capable of processing both pre-recorded audio files and real-time microphone input. With the increasing reliance on voice-driven applications in various sectors such as education, healthcare, accessibility, and virtual assistants, the ability to accurately convert spoken language into written text has become more valuable than ever. This system lays the foundation for such solutions by combining open-source libraries like SpeechRecognition and pydub with the powerful transcription capabilities of Google's Web Speech API.

One of the project's core strengths lies in its dual input functionality, which allows users to either upload existing audio files or speak directly through a microphone. It intelligently handles audio format compatibility by converting non-WAV files to WAV using pydub and ffmpeg, ensuring smooth integration with the recognition engine. Furthermore, the project includes robust error handling to manage real-world challenges such as low audio quality, silence, and connectivity issues with the API.

The clean and minimal command-line interface (CLI) ensures that users of all skill levels can easily interact with the system. The modular design also allows for future enhancements, such as multi-language support, offline recognition using models like Vosk or Whisper, and GUI integration for broader usability.