

# Report

## Overview

This project processes an audio file containing multiple speakers, identifies different speakers, and generates a speaker-labeled text transcript. The system processes the audio in a single step using `complete_file.py`, applying speaker diarization and transcription.

## Project Workflow - Running the Project

Run `complete_file.py` with an audio file as input:

```
python complete_file.py <audio_file>
```

## How It Works:

1. Converts the audio file to a standardized format.
2. Applies speaker diarization to identify speakers.
3. Uses Whisper to transcribe speech.
4. Generates a labeled transcript with speaker annotations.
5. Saves the final transcript as `<audio_file_name>_transcript.txt`.

## Technologies Used

- **Python** for scripting and automation.
- **Pydub** for audio processing.
- **Pyannote Audio** for speaker diarization.
- **Whisper (OpenAI)** for transcribing speech to text.
- **FFmpeg** for handling audio file formats.
- **dotenv** for loading environment variables securely.
- **Streamlit** for creating a web-based user interface.

## Streamlit Dashboard

We have also created a **Streamlit dashboard** to provide an interactive interface for uploading and processing audio files.

## Features of the Streamlit Dashboard

- Upload `.wav` files for transcription.
- Process audio files directly from the web interface.
- View the generated transcript in the app.
- Download the transcript as a `.txt` file.
- Clear uploaded files to manage storage.