

PROGRESS REPORT

Group No -01

Prepared By

Shiksha Kaharanthilake

November 16, 2025

Student Name: Shiksha Kaharanthilake

Student ID: 300398278

Group ID: F25_4440_G1

1. Work Log Table

Date	Number of hours	Description of work
25-10-2025	2	Installed Whisper, Pydub, and FFmpeg; configured environment paths.
27-10-2025	1.5	Tested Whisper using sample audio
29-10-2025	2	Collected WhatsApp voice notes and tried to convert .opus to .wav using Pydub.
31-10-2025	1	Collected Messenger audio samples and organized raw audio directories.
2-11-2025	2.5	Set up a preliminary transcription system using Whisper and tested it with sample audio files.
5-11-2025	1	Gathered more Messenger audio samples and identified necessary preprocessing tasks
8-11-2025	2	Compared Wisper models; selected Wisper-base for best accuracy and speed.
12-11-2025	2	Implemented noise reduction and error handling in the transcription workflow
14-11-2025	2	Beginning of Batch-processing(WhatsApp and Messenger audio)

Total Hours : 15

2. Description of Work Done

So far, I have set up the project environment and installed the necessary tools, including Whisper, Pydub, and FFmpeg, and verified their functionality with sample audio files. I collected voice samples from WhatsApp and Messenger and organized them into a structured directory to facilitate processing. I also developed the initial transcription pipeline, converting different audio formats into Whisper-compatible files and generating accurate transcripts.

During this process, I encountered several issues, such as format incompatibility between audio files, occasional corrupted or unsupported files, and differences in transcription accuracy across various Whisper models. To address these, I implemented audio format conversion and normalization using Pydub, added error handling for corrupted or unsupported files, and compared multiple Whisper models to select the one that offered the best balance between speed and accuracy. I also tested batch processing on multiple audio samples to ensure the pipeline works efficiently across real-world data from both WhatsApp and Messenger.

For the next steps, I plan to improve audio preprocessing by enhancing noise filtering, run the full transcription pipeline on all collected samples, clean up transcripts in structured .txt and .json formats, evaluate transcription accuracy against manual checks, and integrate the results into a final report.

AI Use Section:

AI Tool Name	Version, Account Type	Specific feature for which the AI tool was used
ChatGPT	GPT-5 , Free account	learn to convert .opus to .wav using pydub
		To get guides Implement noise reduction and error handling in the transcription workflow using Whisper, Pydub, and FFmpeg
		Batch-processing in WhatsApp and Messenger audio

Value Addition:

- Explored additional materials on audio processing, noise reduction, and speech recognition to make the project more effective.
- Used what I learned about Whisper, Pydub, and FFmpeg to explain how the Voice and Audio Artifact Extractor works.

Appendix:

Prompt 1: How to convert .opus to .wav using Pydub?

ChatGPT Response:

Steps to convert .opus to .wav using pydub

1. **Install dependencies: Python, pydub, and FFmpeg.**
2. **Ensure FFmpeg is in system PATH (or provide its full path in pydub).**
3. **Import pydub.**
4. **Load the .opus file specifying its format.**
5. **Export the audio as .wav, specifying the format.**
6. **Verify the output file is created successfully.**

Prompt 2: How to Implement noise reduction and error handling in the transcription workflow using Whisper, Pydub, and FFmpeg

ChatGPT Response:

1: Organize Your Audio Files

- Collect all voice notes from WhatsApp and Messenger.
- Store them in a structured folder system (e.g., by app or date) so you can easily process them.
- Make a checklist of file types (MP3, WAV, etc.) to ensure compatibility.

Step 2: Preprocess Audio for Noise Reduction

- **Normalize volume:** Make sure all audio files have consistent loudness so the transcription model can detect speech clearly.
- **Reduce background noise:**
 - Remove static, hums, or ambient noise.
 - Focus on speech clarity for better transcription results.
- Tools like Pydub and FFmpeg can perform these tasks automatically on multiple files in batch.

Step 3: Convert Audio to a Compatible Format

- Whisper requires certain audio formats (e.g., WAV).
- Convert all files to a single, supported format before transcription.
- Ensure that conversion preserves audio quality to avoid degrading speech clarity.

Step 4: Implement Error Handling

- **File validation:** Check that the audio file is not missing, corrupted, or empty before processing.
- **Error logging:** Keep a record of files that fail during processing and the reason for failure.
- **Recovery plan:** Decide what to do with problematic files:
 - Skip and continue with the next file
 - Attempt to repair or re-download the audio

This ensures that a single error doesn't halt the entire workflow.

Step 5: Transcription

- Once audio is preprocessed and converted, run it through Whisper for transcription.
- Save outputs in organized formats such as .txt or .json.
- Include timestamps if needed to track speech in long audio recordings.

Step 6: Review and Improve

- Manually check some transcripts to evaluate accuracy.
- If transcription is poor for certain audio types, tweak preprocessing (e.g., additional noise reduction or format adjustments).
- Log lessons learned for future processing and improvements.

Prompt 3: How to do Batch-processing in WhatsApp and Messenger audio

ChatGPT Response:

1. Organize Files

- **Collect all audio files** from WhatsApp and Messenger.
 - WhatsApp: Usually in WhatsApp/Media/WhatsApp Voice Notes/
 - Messenger: Download voice messages from the app or desktop export.
- **Create a structured folder system:**
 - Audio/
 - WhatsApp/
 - 2025-11/
 - 2025-12/
 - Messenger/
 - 2025-11/
- Naming files consistently (e.g., app-date-number.mp3) helps batch-processing.

2. Standardize Audio Format

- Convert all files to a single format (WAV is preferred for Whisper).
- Tools: **Pydub** or **FFmpeg** can process multiple files in a folder automatically.
- Benefits: Ensures compatibility and reduces errors during transcription.

3. Apply Preprocessing to All Files

- **Normalize volume:** Ensures all files are at a similar loudness.
- **Noise reduction:** Remove background noise using filters.
- Apply these preprocessing steps to every file in the folder before transcription.

4. Transcription Pipeline

- Feed all processed files into Whisper in sequence.
- Save each transcript in a structured format (txt or json) corresponding to the original audio file.
- Maintain timestamps if required for analysis.

5. Error Handling in Batch

- **Log errors:** Keep a record of any file that fails processing.
- **Skip corrupted files:** Don't let one problematic file stop the batch.
- **Reprocess if needed:** If a file fails, try fixing it or re-exporting from the app.

6. Automation Tips

- Use **folders** and **consistent naming** so your processing tool (Python, FFmpeg scripts, or audio software) can loop through all files.
- Batch-processing is essentially running the same preprocessing and transcription steps on multiple files automatically, rather than handling each file individually.

Summary of Batch-Processing Steps

1. Collect and organize audio files by app and date.
2. Standardize all audio to the same format.
3. Preprocess: normalize volume + reduce noise.
4. Transcribe all files sequentially.
5. Handle errors without stopping the workflow.
6. Save transcripts systematically for review.