Coordinator: Joseph Jang (jyj3@illinois.edu)
Kiruthika Janakiraman (kj25@illinois.edu)
Sudarshni Ramesh (sr89@illinois.edu)
Sai Shreya Kumar (ssk12@illinois.edu)

CS 510 Project Proposal - Development Track

Functions and Users: A new stand alone AI notes taking web based application that is powered by LLM. This web application will allow users to upload audio/text transcripts and generate meaningful notes based on the content. We aim to improve the reliability of notes taking with post processing and allowing users to input uncommon words or acronyms for reference. The application will also support APIs to fetch audio transcripts from common media streaming platforms (Youtube). Main use case will be as a studying tool for students, as well as a meeting notes taker for business use.

Significance: Manually listening to videos and audios for taking notes is a time consuming process. We address this pain point by enhancing user comprehension where the users will be able to save time by running a lecture video they want or a YouTube video for better understanding and to get concise notes. Additionally, it will be context aware and keyword specific which empowers user personalization. The keyword feature will also help with model reliability especially in business contexts where jargon and acronyms are frequently used which addresses a societal need.

Approach: Our application primarily leverages OpenAI's API to extract notes from videos. To convert audio into text, we will be utilizing the OpenAPI's Whisper model. We plan to use an open-source API like YouTube's Data API to fetch the audio stream for YouTube videos. We then use the ChatGPT API for post-processing, keywords assistance, and notes generation from the extracted text. Our technology stack will incorporate React for the frontend and Node.js for the backend. One potential risk we anticipate is encountering audio files that exceed the Whisper model's input size limit of 25MB. To mitigate this risk, we'll segment larger audio files into manageable chunks for processing.

Evaluation: Traditionally, manually transcribing and extracting key points from an hour long audio recording could take an individual more than the actual length of the recording. In comparison, our application streamlines this process drastically, this means that what traditionally took an hour or more could be done almost instantly. Additionally, the implementation correctness would be tested by comparing our tools's transcriptions against audio source, manually prepared transcripts, and notes focusing on precision of keyword identification, relevance in context and importantly summarization quality.

Timeline:

- Exploration of API/Frameworks feasibility
- Backend (1 Week): LLM API, External source API, Audio file Chunking
- Frontend (1 Week): Media upload, User assigned keywords, UI
- Debugging and Evaluation (1 Week)

Task division:

- Joseph Jang: Frontend UI, notes output formatting and media upload
- Kiruthika Janakiraman: LLM API and audio file chunking
- Sudarshni Ramesh: Fullstack implementation of user assigned keywords
- Sai Shreya Kumar: Adapt external media source APIs to fetch transcripts