

**ITCS 6166 - Computer Communications and Networks**

**Final Project report**

**Submitted by:**

**Group – 9**

**Yaswanth Mareedu (801306490)**

**Aasish Chunduri (801308977)**

**Govind Rahul Mathamsetti (801306027)**

**Manoj Janapal (801306993)**

**Submitted to:**

**Dr. Pu Wang**

# INTRODUCTION

# In today's globalized world, communication across languages and cultures has become increasingly important. With the rise of Artificial Intelligence and Deep Learning, natural language processing and speech recognition technologies have made significant progress in recent years. This project aims to leverage these advancements to develop a real-time voice-to-text conversion and multilingual text-to-speech synthesis application.

# The primary goal of this project is to create an application that can transcribe speech in real-time using the WebRTC protocol, which allows real-time communication between browsers and applications. To achieve this, we employ the deepspeech model, an open-source speech recognition engine developed by Mozilla. The deepspeech model processes the incoming audio stream, converting speech into text, and displaying the results in real-time on a user-friendly interface created using the Streamlit library.

# Another important aspect of this project is the multilingual text-to-speech synthesis, which allows users to convert the transcribed text into speech in various languages. To accomplish this, we utilize the Pywhisper model, an open-source speech synthesis engine. Users can select their desired output language, and the application translates the text accordingly before converting it into speech using the Google Text-to-Speech (gTTS) library.

# In summary, this project aims to provide a user-friendly, efficient, and versatile application for real-time voice-to-text conversion and multilingual text-to-speech synthesis.

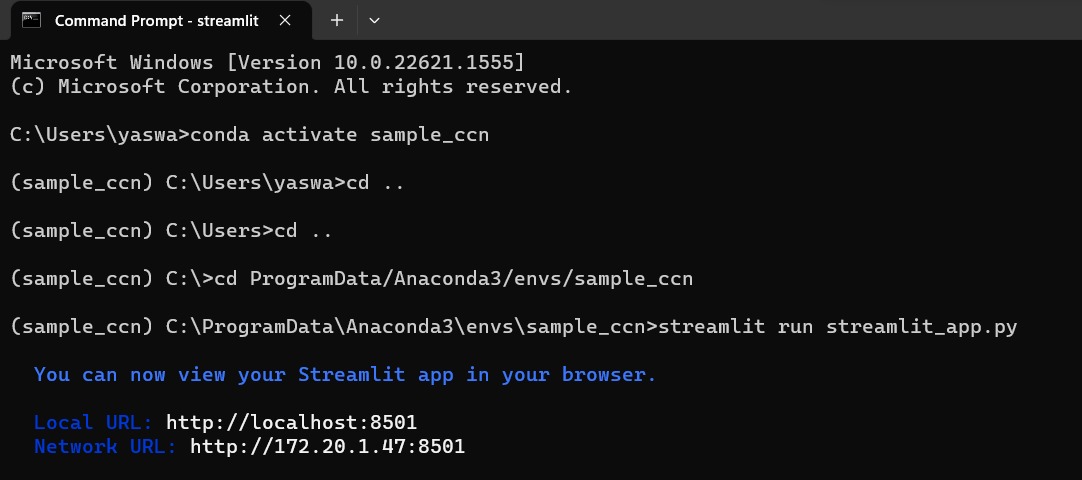
# SYSTEM OVERVIEW

# 

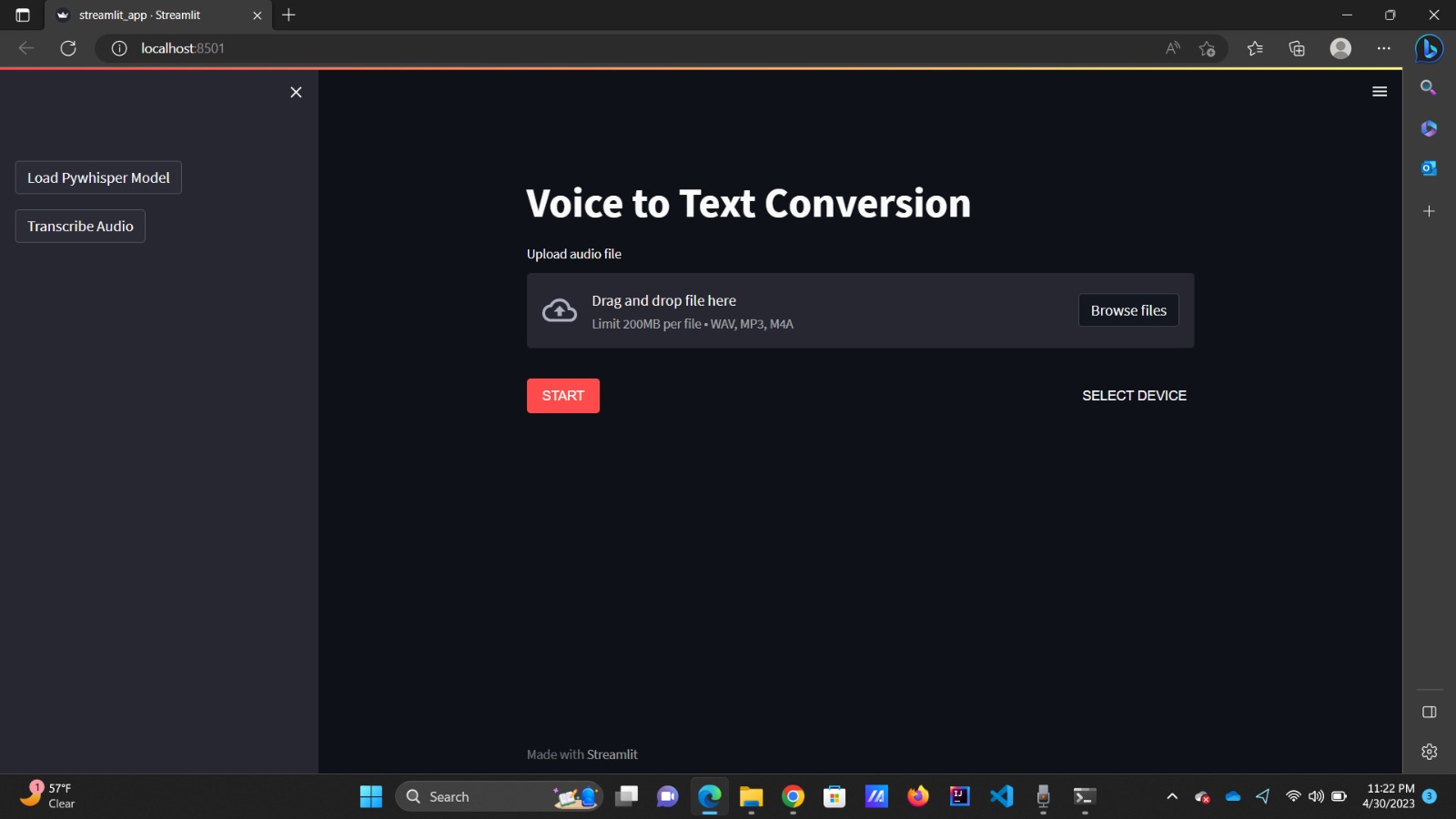
# Fig. 1. Implementation of voice-to-text, and text-to-voice

# IMPLEMENTATION

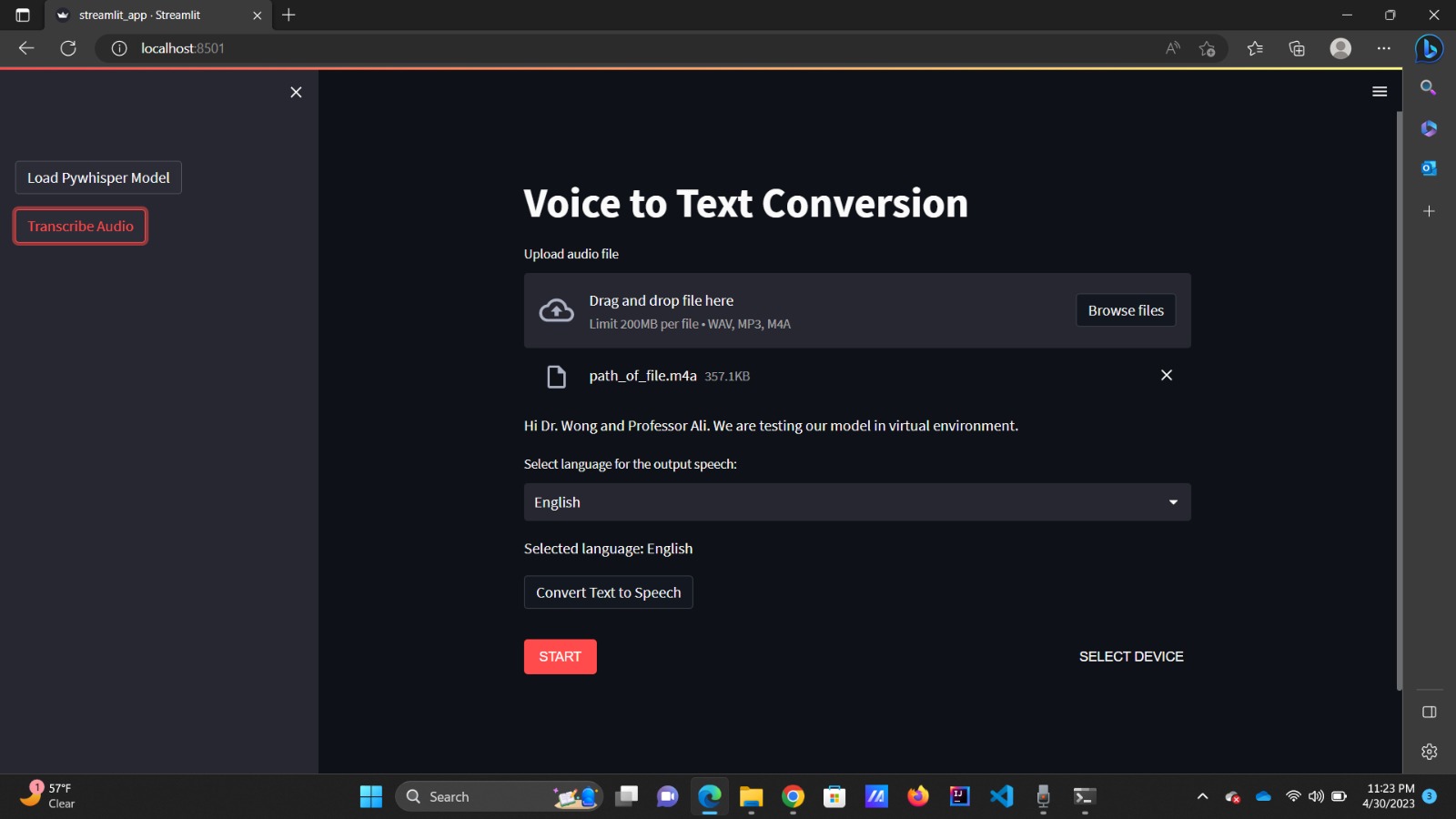
* Install Anaconda on your computer by downloading and running the installation file from the official Anaconda website.
* Create a new virtual environment and install all the packages and libraries described in the "requirements.txt" file. To create a new virtual environment, open a terminal or command prompt and run the following command: “conda create --name myenv --file requirements.txt”, Replace "myenv" with the name you want to give your virtual environment.



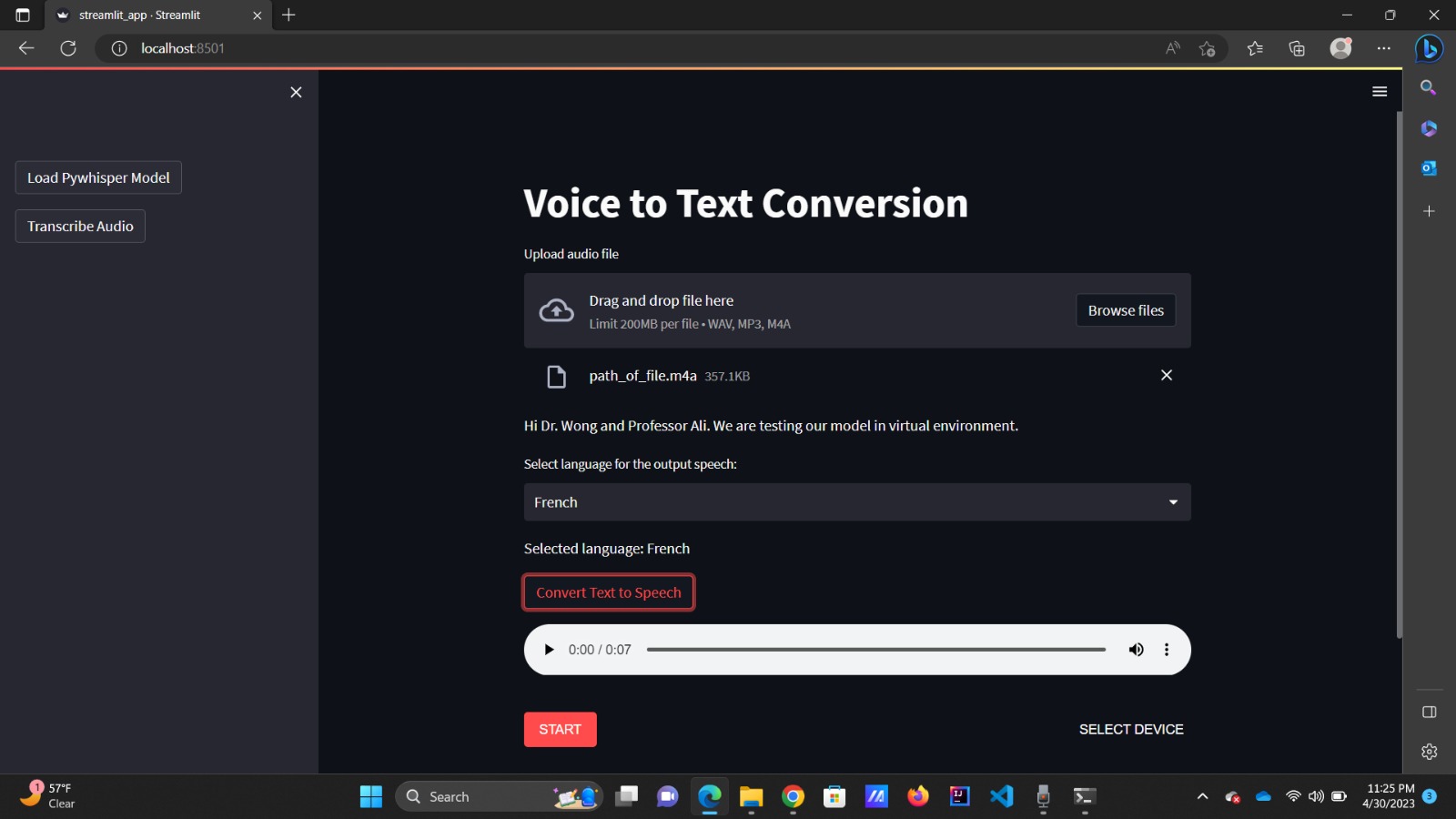
* Place all the required files, such as "streamlit\_app.py" and any audio files needed for the transcribing function, into the virtual environment folder you just created. This will help ensure that your code and data are kept separate from other projects and packages on your system.
* Activate the virtual environment by running the following command: “codeconda activate myenv”, This will activate the virtual environment you just created, allowing you to use the Python packages and environment settings specific to that environment. The python code and the audio file should be placed in the "sample\_ccn" virtual environment. Make sure that streamlit\_app.py and path\_of\_file.wav under same folder. On my laptop the path is C:/ProgramData/Anaconda3/envs/sample\_ccn
* Navigate to the folder where your code and required packages exist. Once in the correct directory, run the command "streamlit run streamlit\_app.py" to start the application. This will launch a local server and open the application in your default web browser.
* Click on the Start button to start the recording of the voice using the microphone and use the browse files option shown in the above pic to use any of the audio file which needs to be transcribed.



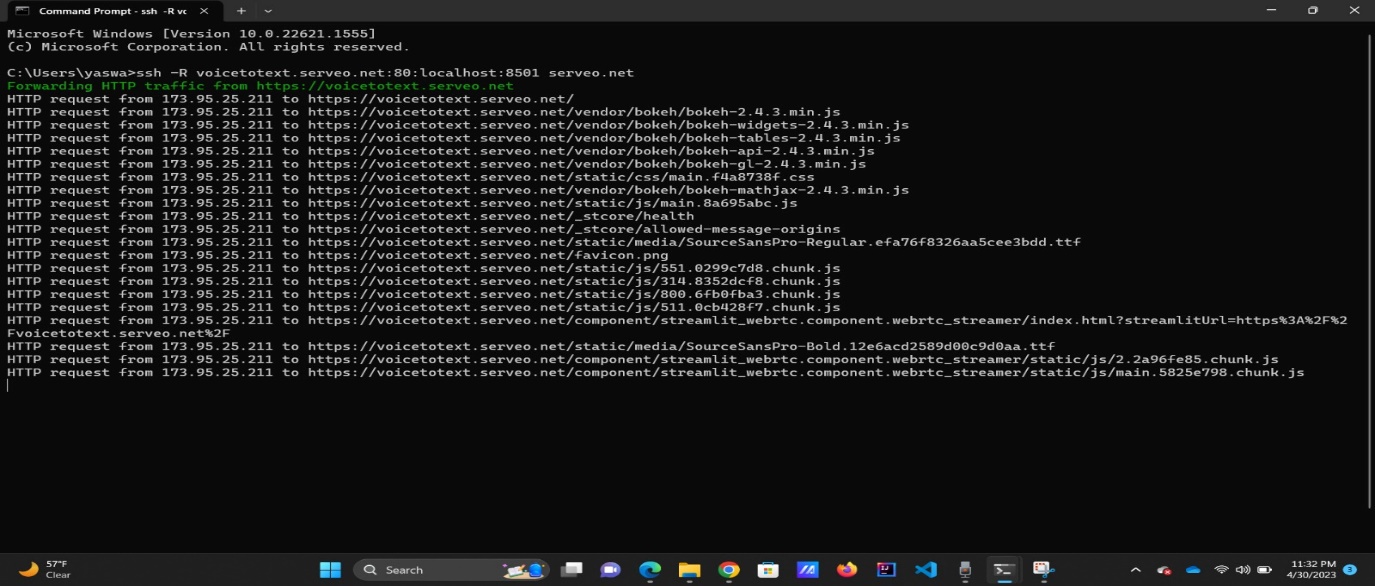
* The below pic represents the transcribed text from the voice input.



* Now we can give the text as input to the Pywhisper model to convert it into any language of our choice from the available languages from the given list.



* We used Serveo to make our local Streamlit web application public and accessible to all users.
* To make it work make sure your Streamlit web application is running locally on your machine and is accessible through your web browser at [http://localhost:8501](http://localhost:8501/)
* Open a terminal or command prompt and enter the following command, replacing example with a unique subdomain name  
  **ssh -R voicetotext.serveo.net:80:localhost:8501 serveo.net**
* Press Enter to run the command. Serveo will establish an SSH tunnel to your local machine and generate a public URL that you can use to access your Streamlit web application.
* Copy the public URL provided by Serveo and share it with your users. They can open the URL in their web browser to access your Streamlit web application.



# CHALLENGES

# First we thought of using Pywhisper to implement real time voice to text conversion but PyWhisper is unable to create a live stream using WebRTC and transfer the audio chunks lively to a model and get the results, so we used deepspeech model.

# Initially, we faced version issues using streamlit webrtc and deepspeech model.

# We tried deploying the model in the Streamlit cloud, but we faced some issues while running the application.

# CONCLUSION

# In conclusion, this project successfully developed an efficient and user-friendly application that enables real-time voice-to-text transcription using WebRTC. By utilizing the power of WebRTC for real-time communication and integrating the deepspeech model, the application provides seamless speech-to-text conversion. Despite challenges such as model selection and version compatibility, the project overcame these obstacles and demonstrated the potential of facilitating real-time transcription and enhance cross-language communication