**CS 2310 – Multimedia Software Engineering**

**Project Milestone 2**

**Dilisha Naidu (din10@pitt.edu)**

**Emotion Detector Using Speech Input**

# Introduction

Detect emotion of the user by uploading the voice samples to a machine learning model which predicts the emotional for each voice sample. The machine learning model is the super component. I have used Vokaturi open API for speech recognition. It will classify the input data into one of the emotion categories like happy, angry, sad, neutral and fear.

# Goal

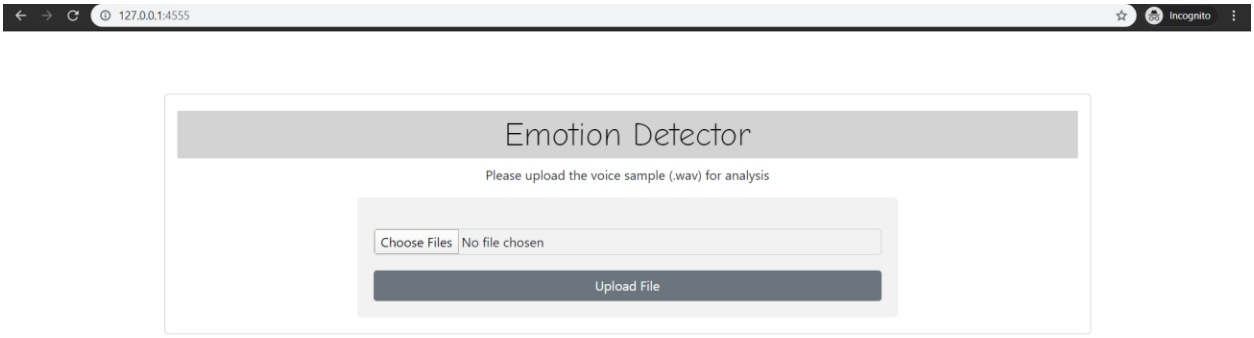
Detect and display the emotion for each voice command by the user.

# Implementation

I have created a web app using python on flask server. The index page of the app allows the users to upload a voice sample. The voice sample should be in the .wav form. Currently the application is working and being tested on my local machine. For handling the web development, I have used HTML5, CSS/Bootstrap and JavaScript.

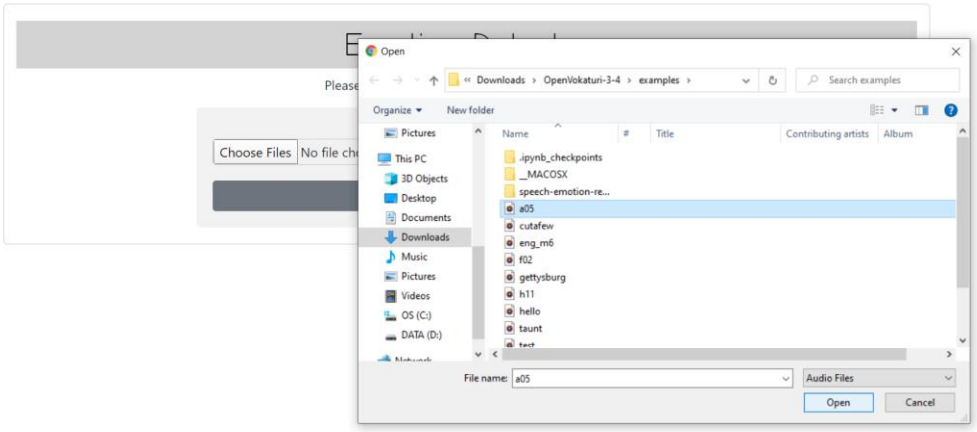
# Working

The index page allows the user to upload a voice sample (.wav only) from their system. They can browse their system for the voice sample file and after selecting the file, click on upload file button.

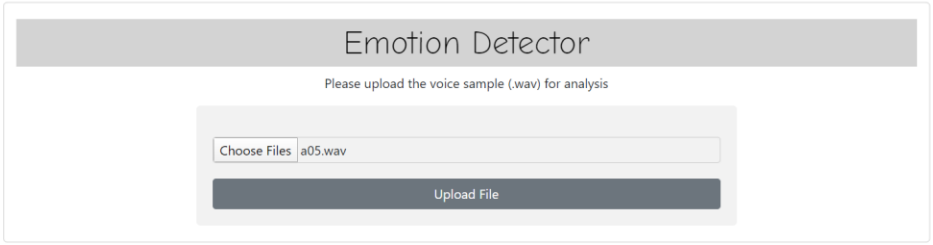


Here I have selected a voice sample-a05.wav, which is an angry emotion voice sample. 

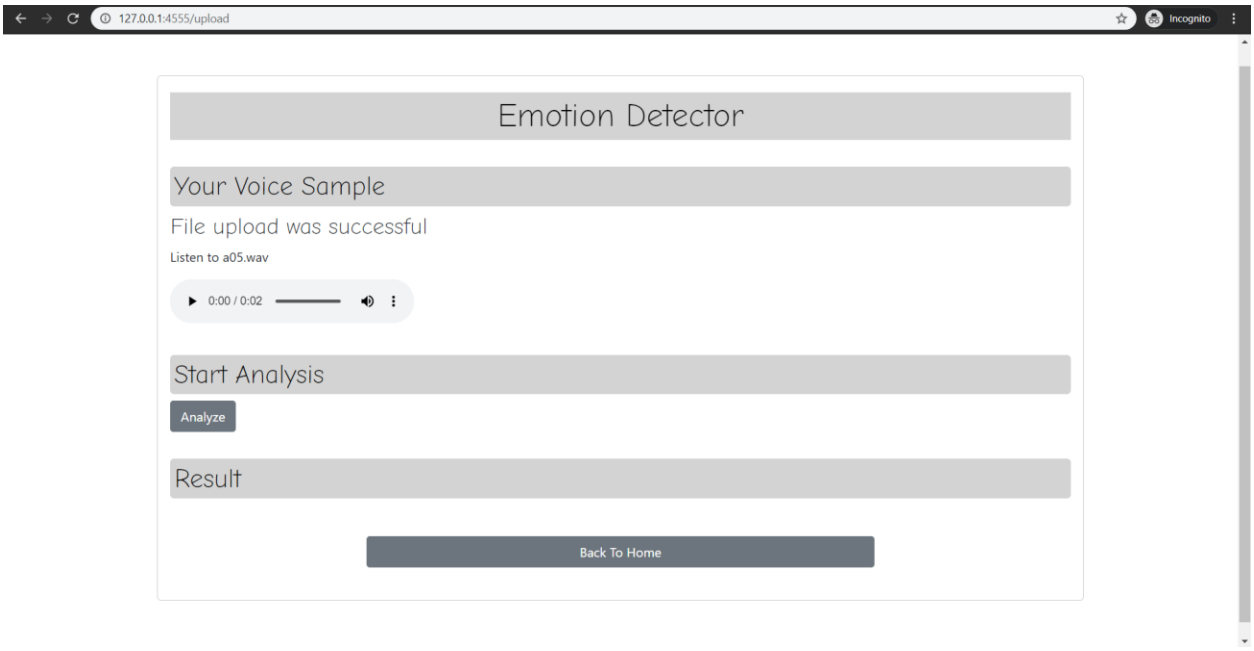
(attached the voice sample).

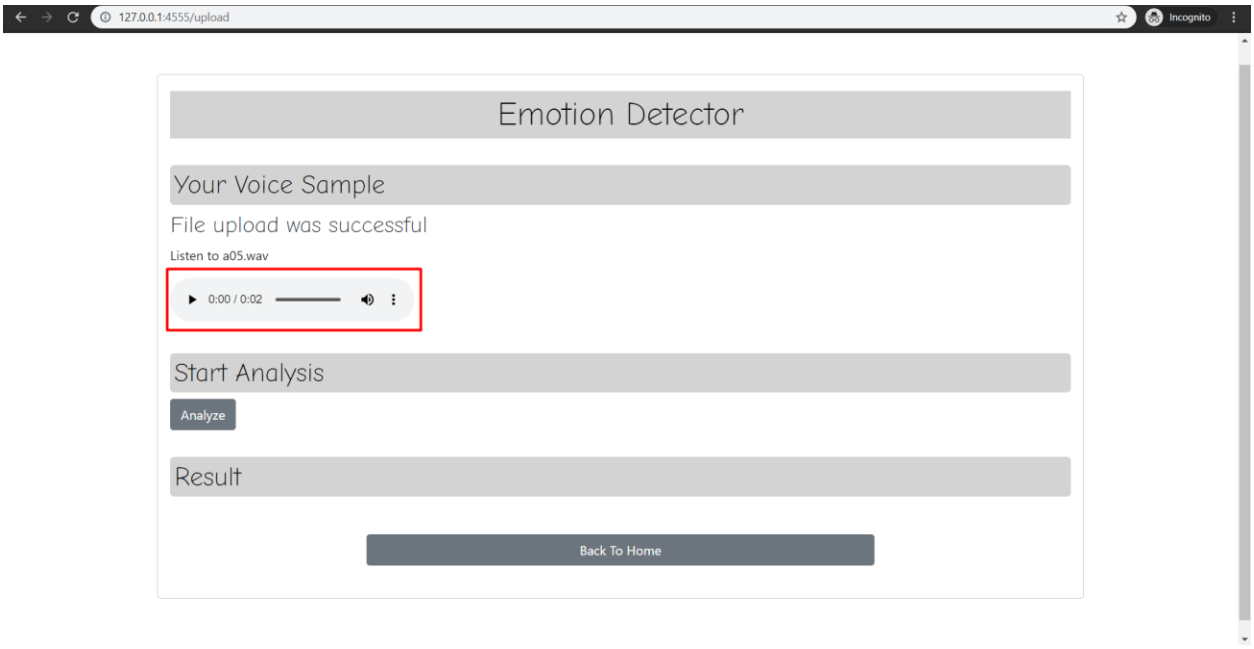




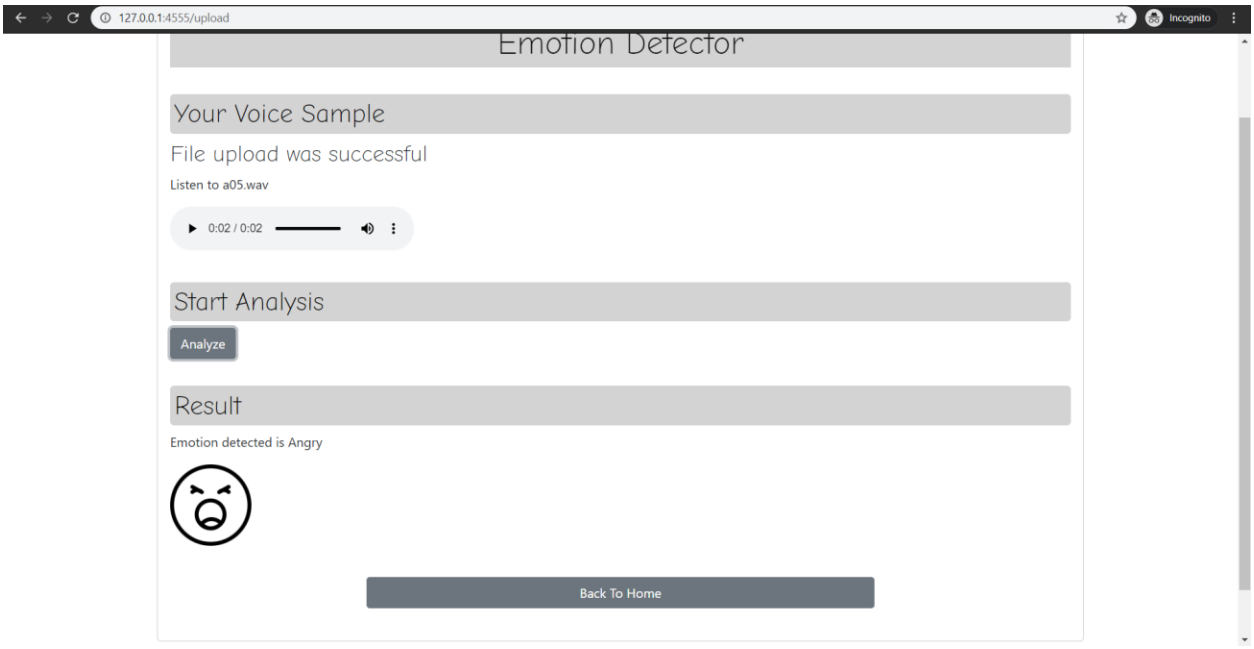
After uploading the voice sample, the user will be taken to the next page. Here it will show if the file upload was successful or not.



With the audio control, the user can also listen to the upload voice sample.




The user can then click on Analyze button to check the results. Clicking on back to home button will take the user to the index page where they can upload new voice samples.



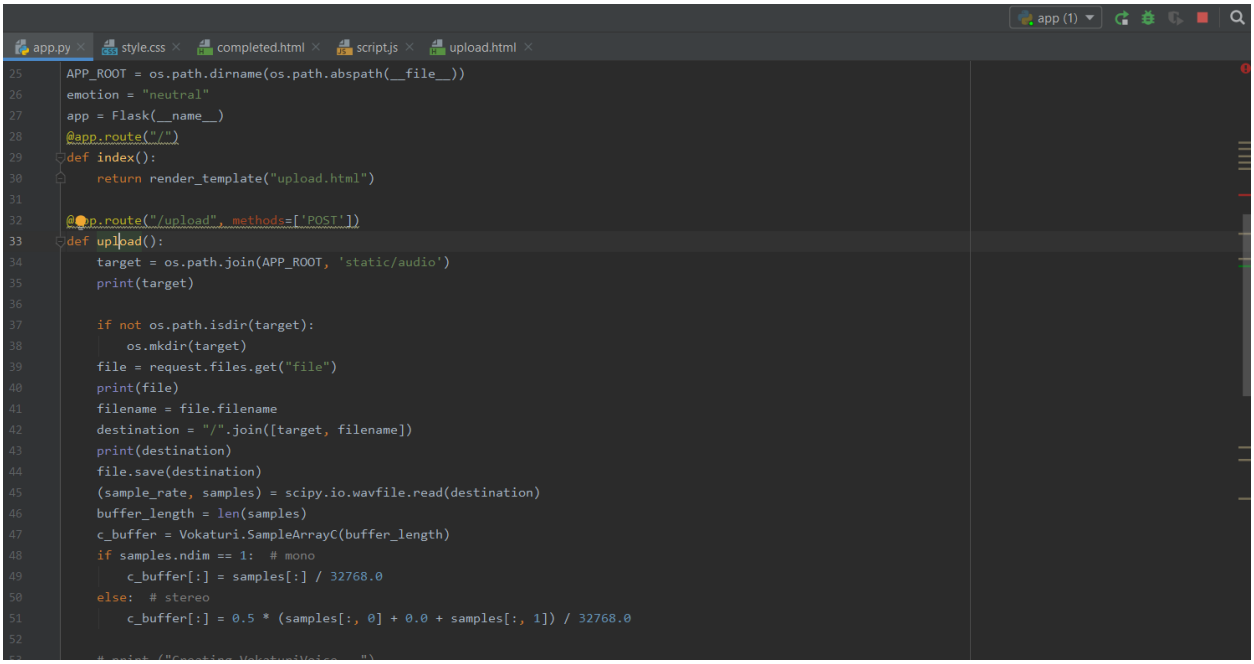
# Code

App.py: This file contains all the routes required in the application

/ : root path for upload.html and index()

/upload : for navigating to the completed.html page and upload ()

upload () function is where the audio input is saved on the server and then passed to the open Vokaturi api for emotion analysis. Once the analysis is completed, the upload () will render the completed.html template and send the emotion string value and audio file name.



**HTML templates**:

*upload.html*: Is the landing page where users can upload their voice samples

*completed.html*: the success page after file upload where user can listen to the uploaded voice sample and check the results

**Static files**: JavaScript, CSS

*script.js*: The js file to dynamically render the audio file and the results received from the server.

*style.css*: The css file to specify the styling rules

## Future work:

- User Sign In/Log In module:
    - A user sign in/log in module can be created, so that the users can keep a track of their uploaded voice samples with timestamp and the resultant emotion detected.
- Data analysis can be implemented to check for the user voice samples and emotion results over a period of month or week to show the mood trends of the user.
- Compatibility with other devices. Currently this is created as a web application but can be extended for android phones, iPhone, iPad or smart watches etc.