

# Speech Based Music Player

## CS 566

Submitted to Dr. P.K. Das

### **Prepared By :**

Kamal Lakhara(194101025)  
Sunil Shinde(194101051)  
Vinay Kumar Gupta(194101056)  
Manish Gupta(194101027)

# **Index**

1. Introduction
2. Front End
  - a. Live Training Module
  - b. Playlist
    - i. English
    - ii. Marathi
    - iii. Hindi
3. Back End
4. Future Scope
5. Conclusion

## **Introduction**

In this project we have developed an “Speech Based Music Player”. This application takes speech as input and performs various actions as required. It can be a very good application for someone who is unable to read and write i.e. who is not good at handling complex applications. In this application,

1. Users can select any one of the 3 Languages (Hindi, English and Marathi) with the help of a recording module.
2. After selecting the language, a list of songs of that language will be displayed.
3. Now the user has to select any song of his/her choice by speaking the name of the song with the help of a recording module.
4. The selected song will be played.
5. Users can even train the model on their voice for good accuracy.

The Speech Recognition part is built using the Hidden Markov Model. It also has a live training module and hence can adapt to new voice inputs.

The following keywords can be recognized:

1. English
  - a. Alone
  - b. Animals
  - c. Believer
  - d. Liar
  - e. Faded
2. Marathi
  - a. Apsara
  - b. Zingaat
  - c. Dhaga
  - d. Morya
  - e. Sairat
3. Hindi
  - a. Naina
  - b. Befikre
  - c. Dil Se
  - d. Chunar
  - e. Lakshya

## Frontend

Frontend is made on Visual Studio to provide GUI for a user friendly interface along with voice command which guides the user with different modules.

1. When the application starts, the user should select the language by speaking the language name.
2. After selecting the language, a new window opens with the playlist of the songs of that language.
3. Now the user can select the song he wants to listen to from the list by speaking the song name.
4. The selected song will be played.

If the user wishes to test how well the application performs on his/her own voice, they might select the “Live Training”.



## Live Training Module

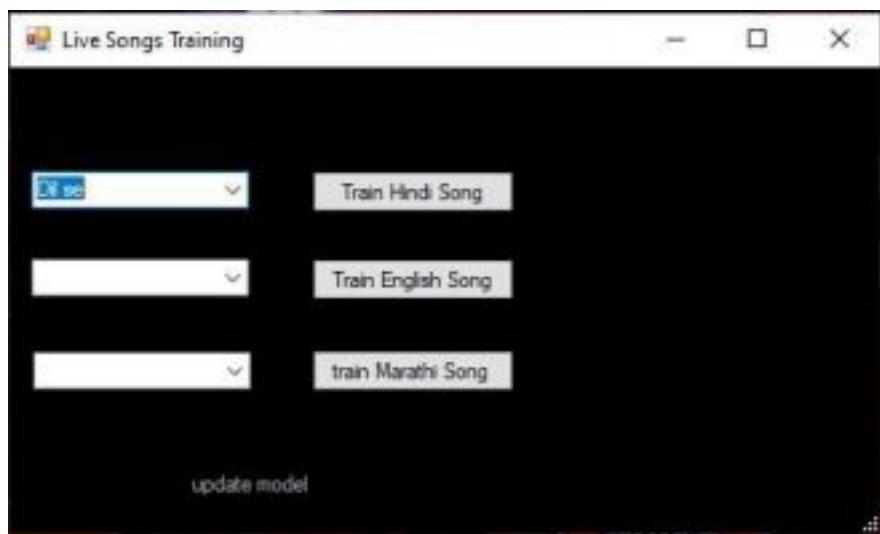
Live training module is for new users. If a user is using the application for the first time, they can check how well the application is performing on their own voice. If they are not satisfied with the accuracy of the model then they can train the keyword till they are satisfied.

Live Training Module takes 5 utterances of each word which are used in the program to train models for any new user and adds the data in the model so if a new user tries to operate the application then it easily recognizes the new user..

The new dataset that the system has received is used for training and a new and updated model is created.



Interface for Training Song Names



## Playlist

After the model is trained, the user can now play the song he wants on his voice commands. User can select the language he/she most likes and play the songs of his choice from the list of songs that are there in the application.

User can select one of the languages from the list i.e. English, Hindi, Marathi.



After selecting the language, the user can select the song from the displayed list.



Selected song is being played.

```
C:\Users\Kamal Lakhani\Desktop\speech proj\guicpp\Debug\guicpp.exe
Configuring the Sound Hardware:
Start Recording.....
Stop Recording.
Language is: Marathi
Configuring the Sound Hardware:
Start Recording.....
Stop Recording.
2
Song Is: Morya
```

## **Back End**

The backend is also made by C programming on Visual Studios. Purpose of backend are:

1. To manage the Recording Module i.e. taking voice input and taking out samples from it.
2. Calculate the Probability of the spoken word by which it matches the most by HMM algorithm.
3. Take input for Live Training module i.e. taking 5 input in voice and making new models i.e. A and B matrix and store it in the pre-stored models for future use.

## **Future Scope**

The current application uses a keyboard input to start the recording module and actually start the speech recognition process. The future implementation of this project can include a hotword feature. The user will be able to catch the recording module's attention by calling it by a name, say "Music Player". As and when the application recognizes the keyword "Music Player", it starts its recording module and becomes ready to listen to the incoming keyword.

- a. This requires the application to always keep another recording module open the whole time.
- b. It also needs to actively cut down noise and recognize the hotword efficiently.
- c. Thus this model needs to be really efficient

The current software application does not support a large vocabulary, the work will be done to train the model for new words and increase the efficiency of the application. This is a desktop application modification that can be done on it so as to implement it in Android Devices.

## **Conclusion**

Speech based "Music Player" will be used to automate the process of playing the song. This tool will minimize the efforts of users to manually search for the song and play it. This application efficiently uses an HMM model to recognize keywords required to play songs. The model has a recognition accuracy of 100% on test data and performs fairly well on new voices.