Speech Emotion Recognition Web App

Name: Akhilesh Kumar Shah
Project: 1 @Feynn Labs ML Intern
Date of Submission: 12-May-2024

Abstract:
Human communication, both verbal and non-verbal, is fundamental to societal interaction. Emotions conveyed through speech have long served as a vital medium of communication, predating the discovery of spoken language. This project focuses on detecting emotions conveyed by speakers during conversation. For instance, fear, anger, or joy often result in speech characterized by increased volume, speed, and pitch range, whereas sadness tends to manifest as slower and lower-pitched speech. Emotion detection through voice patterns holds numerous applications, including enhancing human-machine interaction and developing recommendation systems for movies and music based on mood.

In this project, we propose a classification model utilizing a deep neural network, specifically a Convolutional Neural Network (CNN), to discern emotions conveyed through speech. The model will be trained to classify common emotions such as neutral, calm, happy, sad, angry, fearful, disgusted, and surprised.

Our approach involves building a user-friendly web interface resembling a chat environment using the Flask framework. This interface will facilitate communication between two users, allowing interaction through both text and voice inputs. Sentiment analysis will be performed on text-based chats, while emotions conveyed through audio files will be analyzed using the CNN model.

1. Problem Statement:

Businesses struggle to accurately detect and interpret emotions conveyed through speech, impeding effective communication and customer engagement. Integrating speech emotion recognition into web interfaces presents technical and usability challenges, necessitating seamless integration and real-time analysis capabilities. Additionally, developing a sustainable business model around this technology is essential for viability and scalability.

1. Subscription-based services are needed to sustain continuous development and support.
2. Tailored enterprise solutions must address diverse industry needs for effective implementation.
3. API integration is crucial for developers to seamlessly incorporate emotion recognition into their applications.

4. Consultancy services are required to interpret emotion analysis results and implement strategies for improving communication and engagement.

2. Introduction:

Human Emotions are quite difficult to analyze from a quantitative perspective. One way through which human emotion can be analyzed is through the facial expression that a human being displays with difficult emotional states. Another way by which the emotions of human beings can be analyzed is through the pitch and intensity of the voice. The modulation of voice at different emotional states is quite different and distinct. These characteristic differences in properties of human speech help to guess the emotion of a person.

The main motivation for building this project is to improve the efficiency and accuracy of human-computer interaction and use this feature of human emotion recognition in efficient and accurate recommendation systems. The example of a movie recommendation system can be taken where our project might be useful. Our project will help to give better recommendations based on the current mood of the user. Our project can also be used partially in the medical field especially psychology and mental health-related diagnosis because the human emotions conveyed through speech during interaction with doctors are of paramount importance for the diagnosis of mental diseases like depression.

Speech emotion recognition is a system that identifies the emotional state of the human being from his or her voice. The system that we intend to build will be able to predict 8 different natural human emotions expressed through speech.

3. Market/Customer/Business Need Assessment:

3.1. Market Demand:

1. There is a growing demand for solutions that enhance communication and engagement through emotion recognition in web applications.
2. Businesses across various industries seek ways to better understand customer sentiments and improve user experience.
3. The market shows interest in integrating advanced technologies like speech emotion recognition to stay competitive and meet evolving customer expectations.

3.2. Customer Pain Points:

1. Customers face challenges in accurately conveying emotions through text alone, leading to misunderstandings and ineffective communication.
2. Existing communication platforms often lack real-time emotion detection capabilities, hindering meaningful interactions.
3. Businesses struggle to gauge customer sentiment and respond appropriately, impacting customer satisfaction and loyalty.

3.3. Business Opportunities:

1. Providing speech emotion recognition in web applications opens up opportunities for differentiation and value-added services.
2. Subscription-based models offer recurring revenue streams while catering to varying customer needs and budgets.
3. Enterprise solutions tailored to specific industries enable businesses to address unique communication challenges effectively.
4. API integration empowers developers to enhance their applications with emotion recognition capabilities, fostering innovation and market growth.
5. Consultancy services offer additional revenue streams and opportunities for establishing expertise in emotion analysis and communication strategies.

3.4. Competitive Landscape:

1. Competition in the emotion recognition market is increasing, with companies offering similar solutions across various platforms.
2. Differentiation through features such as real-time analysis, accuracy, and ease of integration will be crucial for gaining a competitive edge.
3. Building strong partnerships with industry players and leveraging first-mover advantages can help capture market share and establish leadership in the segment.

4. Requirement Analysis

4.1. Hardware Requirements

i. Processor: Intel i5 2.4GHz
ii. Hard Disk: 1 TB
iii. RAM: 8 GB

4.1.2. Software Requirements

i. Operating system: Windows 10 64Bit
ii. Programming Language: Python
iii. Front End: HTML, CSS, Javascript
iv. Framework: Flask
v. Database: MySQL
vi. IDLE: PyCharm

4.2 Software Requirements Specification

4.2.1. Product Perspective

The Product is supposed to be an open source under the GNU public user license. The product is a Web Interface-based implementation of the Speech Emotion Recognition System. The product deals with the detection of the human emotion through the speech input. This product provides a real-time analysis of human emotion through the chat, which might be in the

form of text or audio. For the text, the analysis is limited to the sentiment analysis but for the audio input, the analysis is done by using a model generated using a Convolutional Neural Network-based deep learning algorithm.

System Architecture

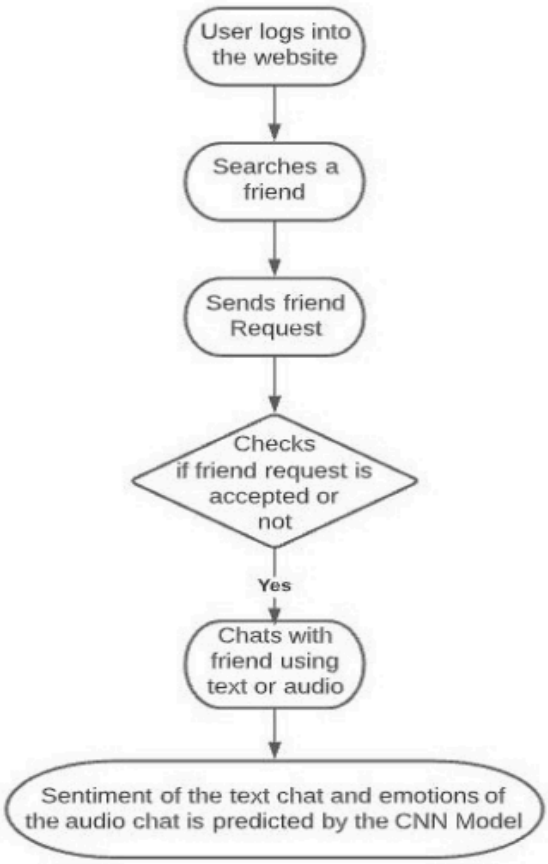

Figure 4.2: System Architecture

4.2.2. Product Features

i. Users can search for a friend and send them a friend Request.
ii. Users can check if the friend request is accepted or not.
iii. Once the friend request is accepted, the user can chat with his/her friend either through text or audio input.
iv. The product is capable of predicting the sentiment of the text as negative, neutral, or positive using the machine learning approach.
v. The product can predict the emotion of the input audio using a model generated by a convolutional neural network.
vi. The count of each type of emotion from the chat is displayed to get a clear perspective of the overall mood of participants in the chat.
vii. The admin has the privilege to log in and view all the chats of all users.

4.2.3. Operating Environment

This project runs on an operating system of any type i.e. it can run on the Windows operating system of with version 7 and higher, Linux operating system, and Mac OS. The PyCharm IDE is used for designing the front end of the application as well as building the back end of the application. The model generation is done using the CNN algorithm in the PyCharm IDE. MySQL Server is used for database creation and storage.

4.2.4. Design and Implementation Constraints

The tables for the login information must be created in the MySQL Database in advance. MySQL Server will be used as a database. The user must have their correct username and password to enter into their account. There should be at least two users to be able to use all the features of the system.

4.2.5. Assumptions and Dependencies

The product needs third-party Libraries like Bootstrap and Flask framework to run the front-end application. It will also use libraries like Librosa, Keras, and Tensorflow for the backup-end. The MySQL server is also one of the third-party products used for running this product.

4.3. Functional Requirements

i. The model should be able to predict the emotion of the user's speech regardless of the language.
ii. The sentiment analysis should be able to predict the negative, neutral, and positive tone of the chat text.
iii. The users should be able to perform basic functionalities like login, search for friends, send friend requests, accept or reject friend requests, and chat with his/her friends.
iv. The admin should be able to view the chat history of all the users using the system.

4.4. Non-functional Requirements

4.4.1. Scalability

Users can communicate with any number of friends when hosted locally whereas the number of friends it can communicate when hosted remotely depending on the capacity of the infrastructure used to host the application

4.4.2. Performance

The loading time of the web application is very low. Also, the accuracy of the prediction is approximately 70% which is comparatively higher than the other existing system.

4.4.3. Safety Requirements

The basic authentication has been implemented through the use of login and access level differentiation as admin and users.

4.4.4. Software Quality Attributes

1. Accessibility: The web application is hosted locally so the current accessibility is limited but it can be easily accessible once it is hosted remotely
2. Accuracy: The model generated by using a convolutional neural network (CNN) can predict the emotion with an accuracy of approximately 70%.
3. Testability: The application can be easily tested with fewer number of test cases.

5. Glossary/Modules

There are 4 major modules in this project
i. Home Page
ii. Admin
iii. User
iv. CNN Model
v. Database

i. Home Page

The home page is the first interface of interaction between the user and the application
It has a navigation bar to navigate to different pages like home, Admin, User, and New user
The home page acts as the single point from where users of different privileges can interact and navigate the whole application.

ii. Admin

a. The admin module is the most important module of the project. The module gives the highest possible privileges to the admin user. In the admin module, the following tasks can be performed
b. The admin is provided with the default credentials.
c. Using the default credentials the admin can log into the system
d. The admin has a home page where he/she can see the list of all the members registered in the application.
e. The admin can also access the chat history of all the users along with the type of emotions shown by the users in their chat.

iii. User

a. The user module is the largest of all the modules in the application.
b. New users can register using the registration form.
c. The existing users can log in using their phone number and password
d. The logged-in user can then perform various functions like viewing his profile details, he/she can also search for friends and send a friend request to the desired person.

e. The user can view all the friend requests that he/she has received. He/she can also see all friend requests he/she has sent.
f. The user can view all his friends by clicking on the friends button. He/she can chat with a particular friend by clicking on the chat button beside the desired friend's chat button.
g. In the chat room, the user is provided with three modes of chat. The first one is using text in which the user can type the chat in the form of text. The sentiment analysis is performed on the text chats.
h. The user can also upload the pre-recorded audio message in the chat. For these pre-recorded audio messages the emotion analysis is performed using deep learning-based CNN model.
i. The user can record live voice through the mic and then send that as a chat. These recorded audio will be analyzed using the CNN model for emotion recognition.

iv. CNN Model

1. **Dataset:** The dataset used to train and test the model is the RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song) dataset. It consists of 1476 Audio files recorded by both male and female actors in the studio environment displaying 8 different kinds of emotions Anger, Happy, Sad, Disgust, Fear, Neutral, Surprise, and Calm. All the eight emotions have been recorded in both male and female voices. https://psychlabs.torontomu.ca/smartlab/resources/speech-song-database-ravdess/

2. **Feature extraction:** The MFCC is an interpretation of the Mel-frequency cepstrum (MFC). It has been demonstrated to be the state-of-the-art of sound formalization in automatic speech recognition tasks. The audio file is divided into frames, usually using a fixed window size, to obtain statistically stationary waves. The amplitude spectrum is normalized with a reduction of the "Mel" Frequency scale. This operation is performed to make the frequency more meaningful for a significant reconstruction of the wave as the human auditory system can perceive.

   For each audio file, 40 features have been extracted. The features have been generated by converting each audio file to a floating-point time series. Then, an MFCC sequence has been created from the time series.

3. **Splitting of dataset:** The dataset is split into the training and testing sets. 70% of the dataset is used as training data and the remaining 30% of the data is used as testing.

4. **Model generation:** The model is generated by training the model using a training set and then the generated model is tested against 30% of the test dataset. After testing the model is saved as a .h5 file for future uses. The generated model is then connected with the Flask chat web application

v. Database: MySQL is used as the database. A database called speech is created in MySQL which holds all the tables required for the application.
The speech database consists of 4 tables

1. **Chat:** The chat table has attributes like cid, uid, fuid, mtype, msg, result, status, datee to store different information related to chat like chat id, userid, friend's id from whom the message is received, message type i.e text or audio, the chat message, date and time of the sent message.

Field Name	Data Type	Description	Constraints
Cid	Int	Chat Id	NOTNULL,AUTO_INCREMENT, Primary Key
Uid	Int	User id	DEFAULT NULL
Fuid	Int	Friend's user id	DEFAULT NULL
Mtype	Varchar	Message Type	DEFAULT NULL
Msg	Varchar	Message	DEFAULT NULL
Result	Varchar	Whether the message is sent or unsent	DEFAULT NULL
Status	Varchar	Message read or unread	DEFAULT 'unread'
Datee	Varchar	Date and time of sent message	DEFAULT NULL

Field Name	Data Type	Description	Constraints
Frid	Int	Friend's id	NOT NULL AUTO_INCREMENT, Primary Key
Rfrom	Int	Friend request from	DEFAULT NULL
Rto	Int	Request to	DEFAULT NULL
Status	Varchar	Friend request sent or unsent	DEFAULT 'friend request sent'
Datee	Timestamp	Timestamp of the message sent	DEFAULT CURRENT_TIMESTAMP

2. Frequest: The frequest table has attributes like frid, rfrom, rto, status, datee to store information related to friend reques like friend id, request from, request to, the status of the sent friend request, and date on which the friend request is sent.

Field Name	Data Type	Description	Constraints
Cid	Int	Chat Id	NOTNULL,AUTO_INCREMENT, Primary Key
Uid	Int	User id	DEFAULT NULL
Fuid	Int	Friend's user id	DEFAULT NULL
Mtype	Varchar	Message Type	DEFAULT NULL
Msg	Varchar	Message	DEFAULT NULL
Result	Varchar	Whether the message is sent or unsent	DEFAULT NULL
Status	Varchar	Message read or unread	DEFAULT 'unread'
Datee	Varchar	Date and time of sent message	DEFAULT NULL

Field Name	Data Type	Description	Constraints
Frid	Int	Friend's id	NOT NULL AUTO_INCREMENT, Primary Key
Rfrom	Int	Friend request from	DEFAULT NULL
Rto	Int	Request to	DEFAULT NULL
Status	Varchar	Friend request sent or unsent	DEFAULT 'friend request sent'
Datee	Timestamp	Timestamp of the message sent	DEFAULT CURRENT_TIMESTAMP

3. **Friends:** The friend's table has attributes like fid,uid, fuid, datee to store information about friends like friend id, user id, friend's user id and date.

Field Name	Data Type	Description	Constraints
Fid	Int	Friend's id	NOT NULL AUTO_INCREMENT, Primary Key
Uid	Int	User's id	DEFAULT NULL
Fuid	Int	Friend's user id	DEFAULT NULL
Datee	Timestamp	Date and time	DEFAULT CURRENT_TIMESTAMP

Field Name	Data Type	Description	Constraints
Uid	Int	User id	NOT NULL AUTO_INCREMENT, Primary key
Name	Varchar	User Name	NOT NULL
Email	Varchar	User's email id	NOT NULL, Unique Key
Phone	Varchar	User's phone number	NOT NULL, Unique Key
Password	Varchar	User's password	NOT NULL
Gender	Varchar	User's gender	NOT NULL
Dob	Varchar	User's Date of birth	NOT NULL
Ppic	Varchar	Profile picture	NOT NULL

4. **Users:** The user's table has attributes like uid, name, email, phone, password, gender, dob, ppic to store information like user id, name, email, phone, password, gender, date of birth, profile pic of the user.

Field Name	Data Type	Description	Constraints
Fid	Int	Friend's id	NOT NULL AUTO_INCREMENT, Primary Key
Uid	Int	User's id	DEFAULT NULL
Fuid	Int	Friend's user id	DEFAULT NULL
Datee	Timestamp	Date and time	DEFAULT CURRENT_TIMESTAMP

Field Name	Data Type	Description	Constraints
Uid	Int	User id	NOT NULL AUTO_INCREMENT, Primary key
Name	Varchar	User Name	NOT NULL
Email	Varchar	User's email id	NOT NULL, Unique Key
Phone	Varchar	User's phone number	NOT NULL, Unique Key
Password	Varchar	User's password	NOT NULL
Gender	Varchar	User's gender	NOT NULL
Dob	Varchar	User's Date of birth	NOT NULL
Ppic	Varchar	Profile picture	NOT NULL

6. System Design

Design Methodology:

The design process for software systems has two levels:

1. System design or Top level design
2. Detailed design or Logical design.

6.1. System design or Top level design

In the system design the focus is on deciding which modules are needed for the system, the specification of these modules and how these modules should be interconnected.

System design is transition from a user-oriented, document-oriented to programmers. The design is a solution, a "how to" approach to the creation of a new system. This is composed of several steps. It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. The design system implements the feature extraction procedure. Once the extraction procedure is done then the analyzing phase is

processed for the given data sample. The machine learning algorithm is then applied to the processed data and the extracted features of the given address link to present a presumed result.

6.2. Detailed design or Logical design.

It follows Architectural design and focuses on the development of each module.

6.2.1 Conceptual Data Modeling.

It is a representation of organizational data which includes all the major entities and relationships. System analysts develop a conceptual data model for the current system that supports the scope and requirements of the proposed system. The main aim of conceptual data modeling is to capture as much meaning of data as possible. Most organizations today use conceptual data using the E-R model which uses special notation to represent as much meaning about data as possible.

6.2.2. Entity Relationship Model

It is a technique used in database design that helps describe the relationship between various entities of an organization.

6.2.3. Architecture

The aim of speech emotion recognition is the transcription of human speech/text into emotions/sentiments. It is a very challenging task because human speech signals are highly variable due to various speaker attributes, different speaking styles, uncertain environmental noises, and so on. In this work, a Convolution Neural Network (CNN) is proposed for speech emotion recognition (SER). We focus on extracting the most salient frames via the proposed CNN structure from the entire sequence to represent the emotion/sentiment of the person. A CNN model can be thought of as a combination of two components: the feature extraction part and the classification part. CNN-based acoustic model processes the raw speech directly as input. This model consists of two stages: the feature learning stage, i.e., several convolutional layers, and the classifier stage, i.e., fully connected layers. Both stages are learned jointly by minimizing a cost function based on relative entropy. In this model, the information is extracted by the filters at the first convolutional layer and modeled between the first and second convolutional layers. In the classifier stage, learned features are classified by fully connected layers and softmax layers. This approach claims comparable or better performance than traditional cepstral features-based systems.

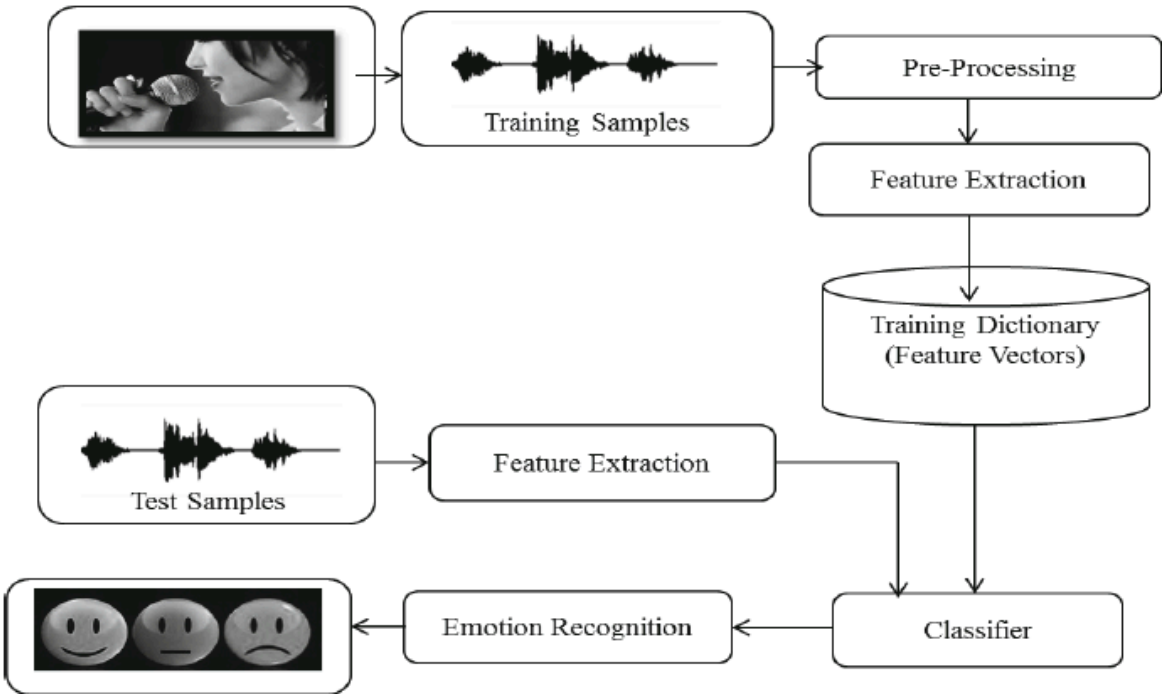

Figure 6.2.3: Dataflow of system architecture

7. System Implementation

System Implementation is the process of defining how the information system should be built (i.e.., physical system design), ensuring that the information system is operational and used, ensuring that the information system meets quality standards (i.e.., quality assurance).

7.1 Selected Software

In this project, there are various system environments used which are given below in detailed explanation.

7.1.1 Python

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). This tutorial gives enough understanding of Python programming language.

Why Python?

Python is a high-level, interpreted, interactive, and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently whereas other languages use punctuation, and it has fewer syntactic constructions than other languages. Python is a MUST for students and working professionals to become great Software engineers especially when they are working in the Web Development Domain. I will list down some of the key advantages of learning Python:

Python is Interpreted: Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
Python is Interactive: You can sit at a Python prompt and interact with the interpreter directly to write your programs.
Python is Object-Oriented: Python supports an Object-Oriented style or technique of programming that encapsulates code within objects.
Python is a Beginner's Language: Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

7.2.2 MySQL
MySQL is an Oracle-based open-source relational database management system (RDBMS) based on Structured Query Language (SQL). MySQL runs on virtually all platforms, including Linux, UNIX, and Windows. Although it can be used in a wide range of applications, MySQL is most often associated with web applications and online publishing.
1. MySQL is a database system used on the web.
2. MySQL is a database system that runs on a server.
3. MySQL is ideal for both small and large applications.
4. MySQL is very fast, reliable, and easy to use.
5. MySQL uses standard SQL.
6. MySQL compiles on several platforms.
7. MySQL is free to download and use.
8. MySQL is developed, distributed, and supported by Oracle Corporation.

7.1.3 PyCharm IDE
PyCharm is the most popular IDE used for Python scripting language. PyCharm offers some of the best features to its users and developers in the following aspects
i. Code completion and inspection
ii. Advanced debugging
iii. Support for web programming and frameworks such as Django and Flask

7.1.4 Flask
Flask is a micro web framework written in Python. It is classified as a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

Why is Flask a good web framework choice?
Flask is considered more Pythonic than the Django web framework because in common situations the equivalent Flask web application is more explicit. Flask is also easy to get started

with as a beginner because there is little boilerplate code for getting a simple app up and running.

7.2 Sample Code

```
from flask import Flask, render_template, request, session, send_from_directory, Response, redirect, url_for
import os
import pymysql
import json
from AnalyseSentiment.AnalyseSentiment import AnalyseSentiment
import pyaudio
import wave
import datetime
from DemoEmotion import DemoEmotion
app = Flask(__name__)
23app.secret_key = 'dfghjjhgfdfghjuytresdcvb'
conn = pymysql.connect(host="localhost", user="root", password="root", db="speech")
cursor = conn.cursor()
APP_ROOT = os.path.dirname(os.path.abspath(__file__))
demoEmotion = DemoEmotion()
@app.route('/')
def index():
return render_template('index.html')
@app.route('/alog')
def alog():
return render_template('alog.html')
@app.route('/alog1',methods=['POST'])
def alog1():
uname = request.form.get('uname')
password = request.form.get('password')

if uname == 'admin' and password == 'admin':
session['role'] = 'admin'
return render_template('admin.html')
else:
return render_template('mmsg.html', msg='Invalid Login Details', color='bgdanger')

@app.route('/ulog')
24def ulog():
return render_template('ulog.html')
```

```
@app.route('/ulog1', methods=['POST'])
def ulog1():
phone = request.form.get('phone')
password = request.form.get('password')
result = cursor.execute("select * from users where password='" + password + "' and phone='"
+ phone + "'")
userDetails = cursor.fetchall()
if result > 0:
for user in userDetails:
session['uid'] = user[0]
session['name'] = user[1]
session['email'] = user[2]
session['phone'] = user[3]
session['gender'] = user[5]
session['dob'] = user[6]
session['ppic'] = user[7]
session['role'] = 'user'
return userHome()

else:
return render_template('mmsg.html', msg='Invalid Login Details', color='bg-danger')
@app.route('/user')
def userHome():
name = session['name']
email = session['email']
phone = session['phone']
gender = session['gender']
dob = session['dob']
ppic = session['ppic']
return render_template('user.html',
name=name,email=email,phone=phone,gender=gender,dob=dob,ppic=ppic)
@app.route('/ureg')
def ureg():
return render_template('ureg.html')
@app.route('/ureg1', methods=['POST'])
def ureg1():
try:
target = os.path.join(APP_ROOT, 'profiles/')
for upload in request.files.getlist("file"):
name = request.form.get('name')
```

```
email = request.form.get('email')
phone = request.form.get('phone')
password = request.form.get('password')
gender = request.form.get('gender')
dob = request.form.get('dob')
filename = upload.filename

destination = "/".join([target, filename])
upload.save(destination)
result = cursor.execute("select * from users where email="'+email+"' and
phone="'+phone+"'")
conn.commit()
if result>0:
return render_template('mmsg.html', msg='User Already Exist', color='bg
danger')
else:
result = cursor.execute("insert into
users(name,email,phone,password,gender,dob,ppic)
values("'+name+"',"'+email+"',"'+phone+"',"'+password+"',"'+gender+"',"'+dob+"',"'+filenam
e+"')")
conn.commit()
return render_template('mmsg.html', msg='User Registration Success', color='bg
success')
except Exception as e:
return render_template('mmsg.html', msg=str(e), color='bg-danger')

@app.route('/alogout')
def alogout():
session.pop('role', None)
return render_template('index.html')
```

8. Testing

Testing is the process of executing a program to find errors. To make our software perform well it should be error free. If testing is done successfully it will remove all the errors from the software.

Test cases:

S. No.	**Test Case Description**	**Actual Value**	**Expected Value (+ve)**	**Failed Value (- ve)**
1	Admin Login	Admin will provide admin login details	The admin home page will open	Invalid login details
2	Create (or) Upload Model	Admin will create an ML model with voice data	Less loss in prediction	Re-run the model to get min loss
3	View User Emotions and Sentiments	Admin can view user emotions and sentiments	Admin can see the user emotions and sentiments based on their chat	Fails to load model
4	User Registration	The user will provide his details to register	Registration successful	Duplicate details
5	User Login	User will provide user login credentials	The user home page will open	Invalid login details
7	Friend Request	User may receive friend request from their friends	Accept friend request	Reject friend request

8	Chat with Friend	Users can chat with their friends through text or voice	If voice record less duration prediction will be accurate if the record duration more the prediction may not be good	If you provide the length audio production may wrong

Table: 8: Test Cases

9. Results and Output

These results and outputs contribute to the overall effectiveness and usability of the speech emotion recognition web application, providing users with valuable insights into their emotional expression and facilitating improved communication and interaction.

Speech emotion recognition models are trained on a large dataset of speech recordings that have been labeled with the corresponding emotion. The model learns to identify features in the spectrogram that are correlated with different emotions. For example, anger might be correlated with a lot of high-frequency energy, while sadness might be correlated with a lot of low-frequency energy.

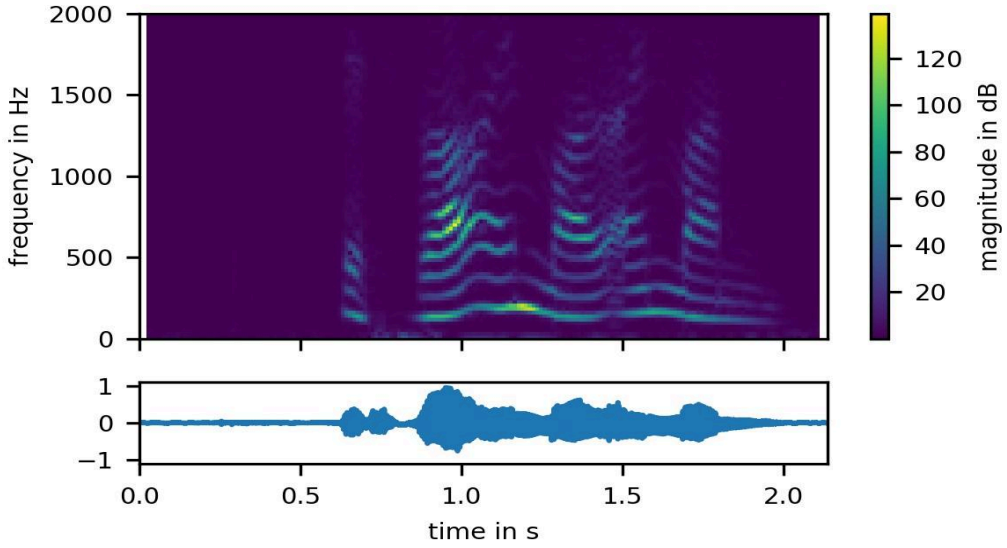

Figure 9.1: Analyzing speech signals in frequency and time

A. Home Page:

The home page is the first interface of interaction between the user and the application. It has a navigation bar to navigate to different pages like Home, Admin, User, and New User.

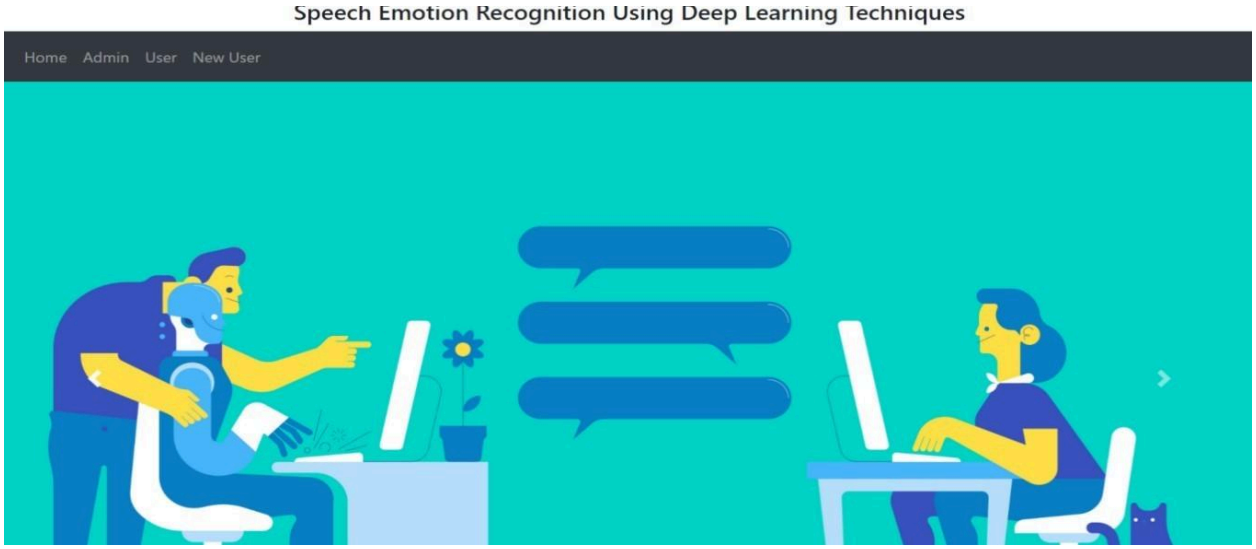

Figure 9.2: Home Page

1. Admin Login:

This Administration Login Page consists of a Username & Password, which is only used by the admin for monitoring these whole websites. It provides default credentials. He can see all the list registered in the application and can also access chat history and emotions of all users chat.

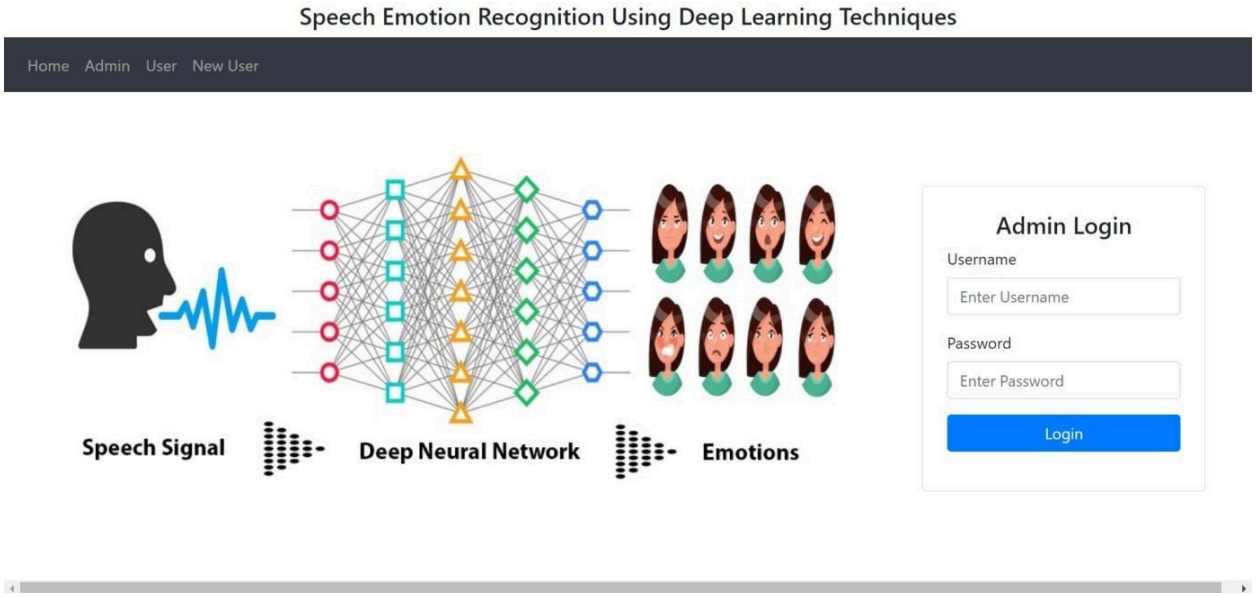

Figure: 9.3: Admin Page

2. New User Registration Page:

New users can register using the registration form. While registering the form, the member needs to fill in the: Name, Email, Phone No., Gender, Password, Date of Birth, and Profile Picture.

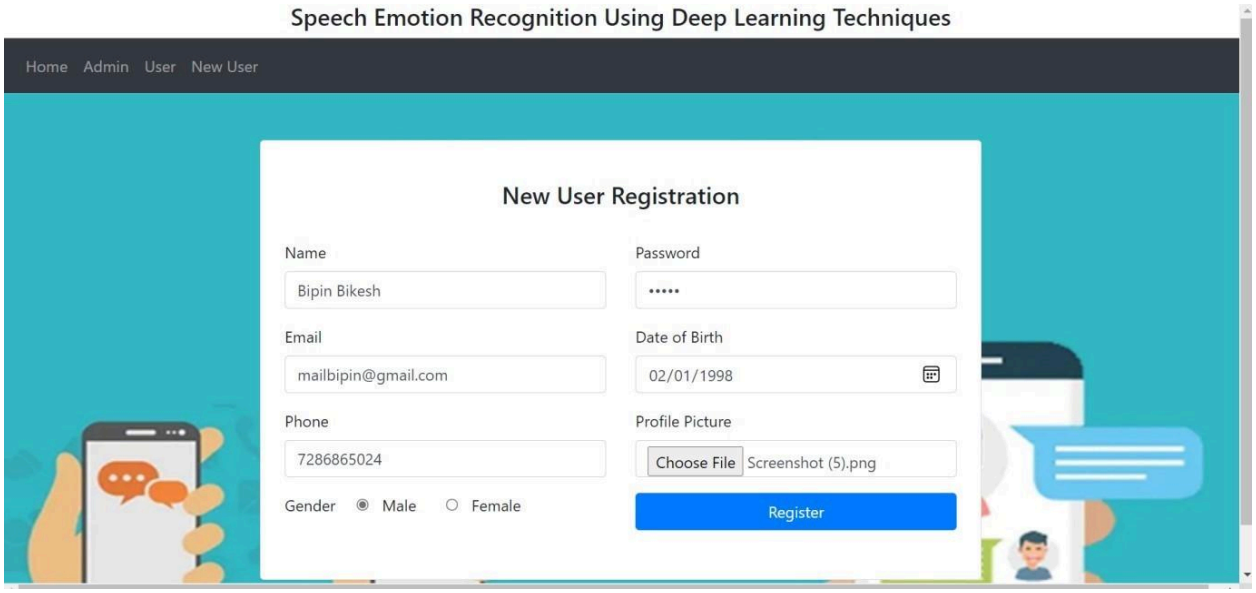

Figure: 9.4: User Registration

3. User Login Page:

The existing user can fill out the registration form before logging in to the user login page. Users need to enter the correct phone no. And the password then only the login page will proceed otherwise it will show incorrect information. The existing users can log in using their phone number and password. The logged-in user can then perform various functions like viewing his profile details, he/she can also search for friends and send a friend request to the desired person.

Figure: 9.5: Existing User Registration

User Profile Details:

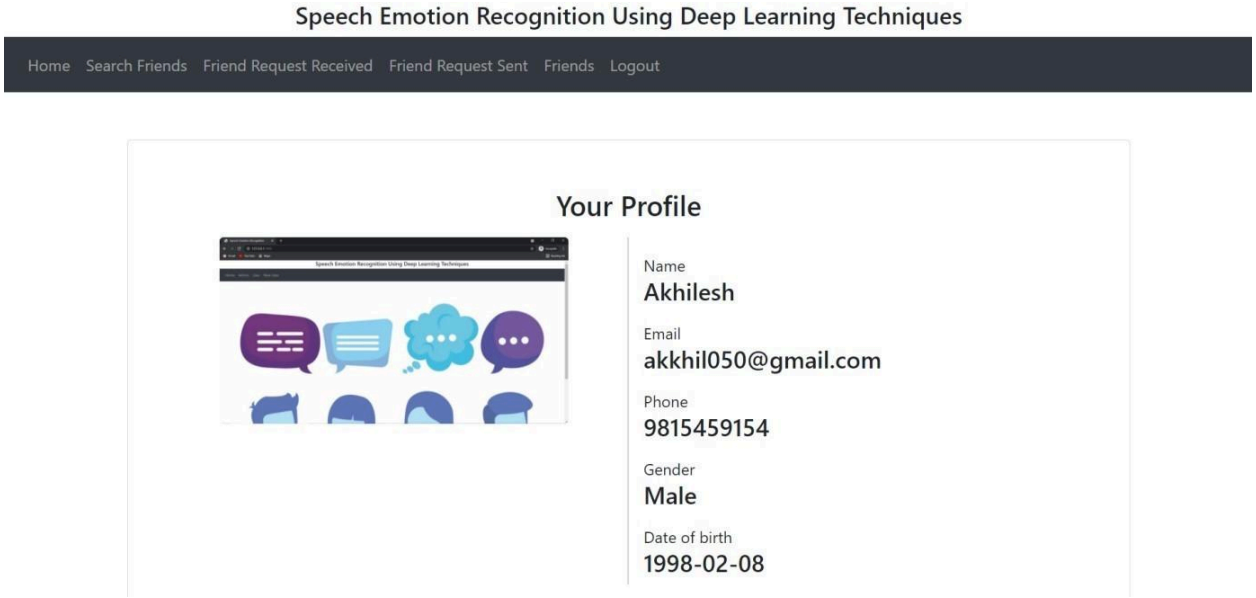

Figure: 9.6: User Profile Details

Sending Friend Request to Registered User:

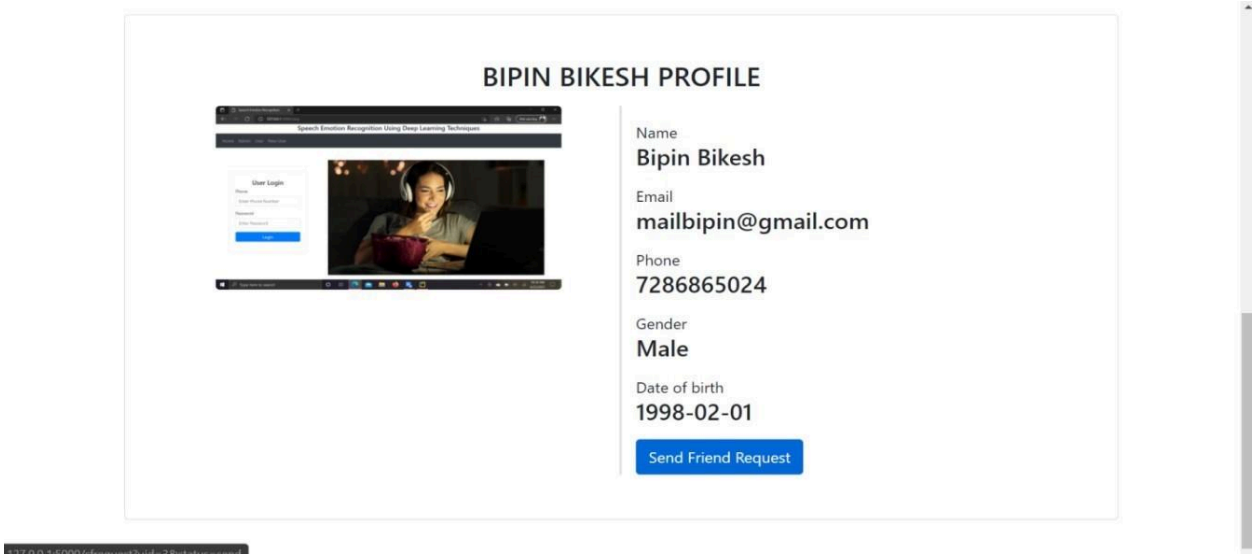

Figure: 9.7: Sending Friend Request to Registered User

Sent Friend Request Received From the User:

The request received from the user, even he can accept or reject the sent request. The below details shown:

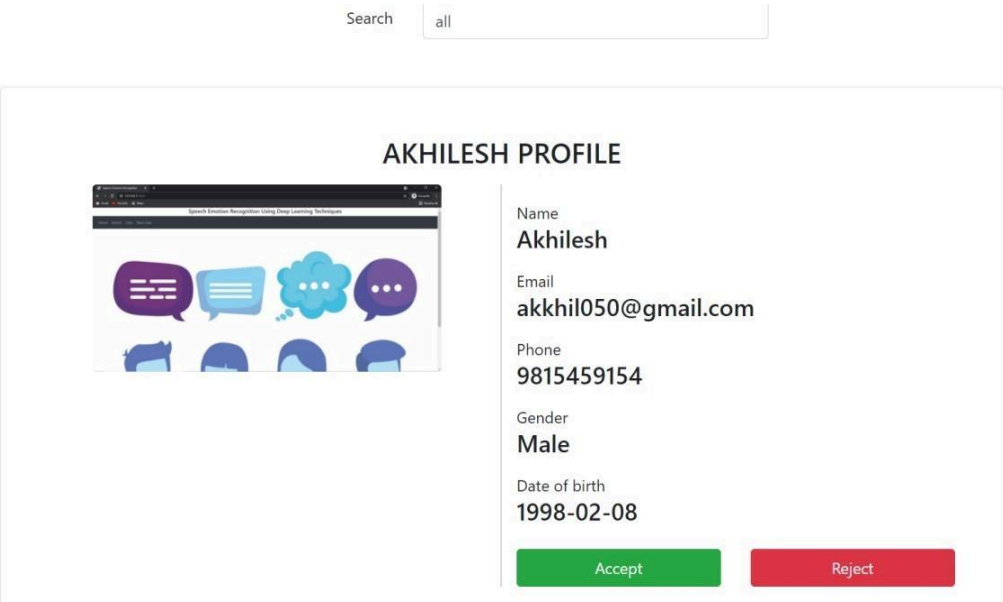

Figure: 9.8: Sent Friend Request Received From the User

Chat Window:

The chat window is shown below:

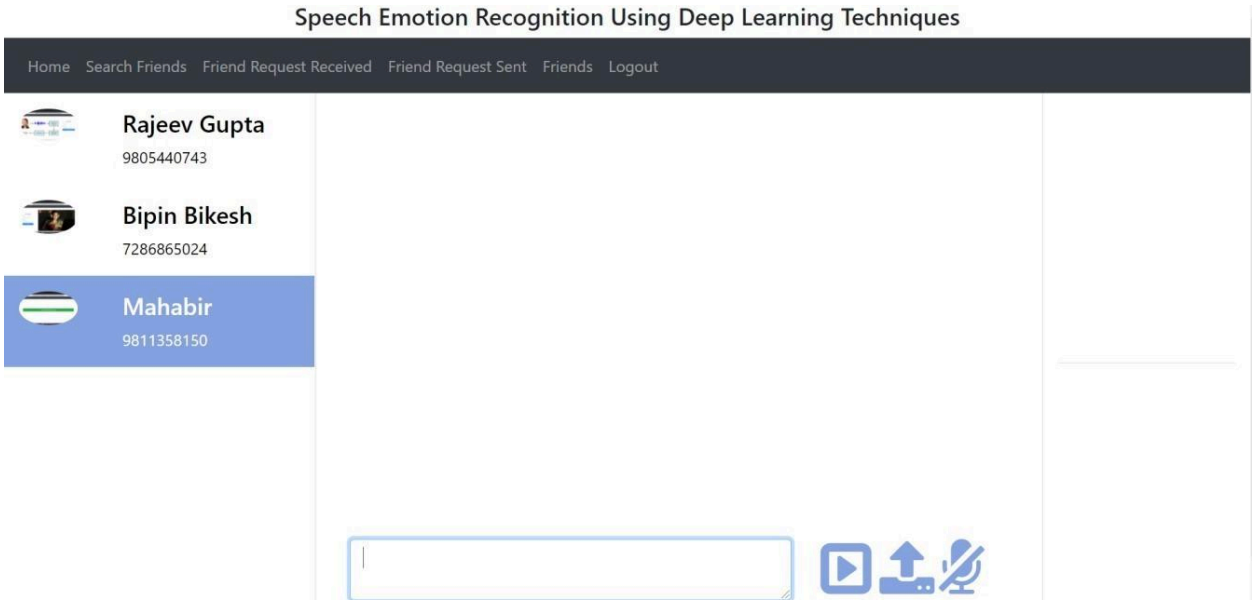

Figure: 9.9: Chat Window

Message Sent:

Message sent & sentiment shown for the text message (i.e., Neutral).

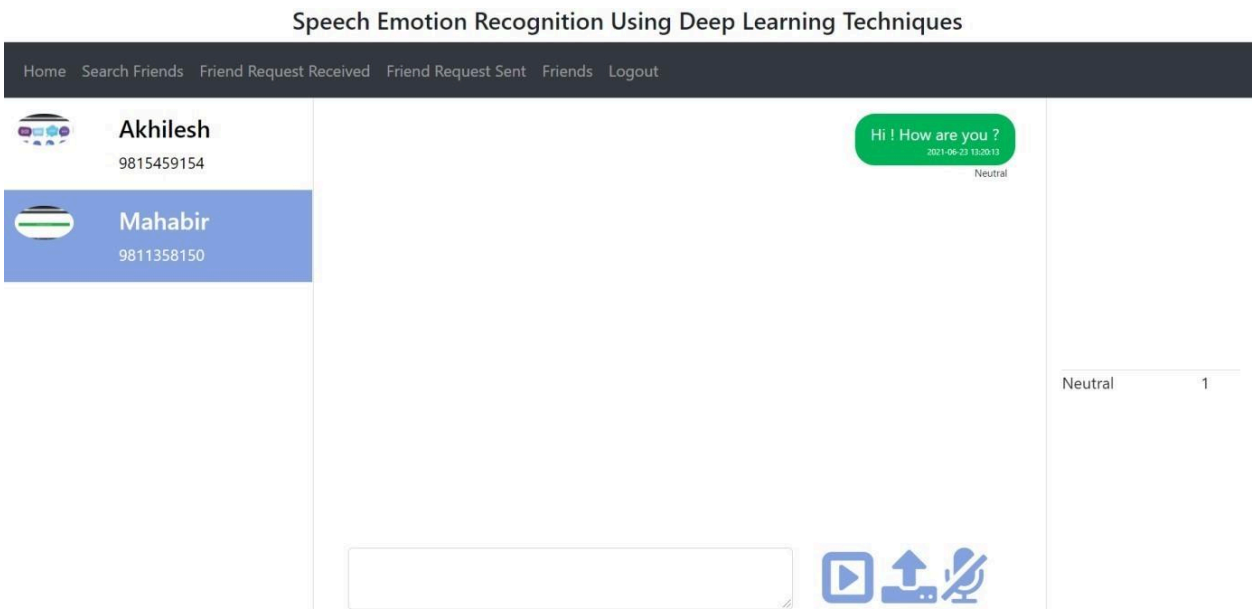

Figure: 9.10:Message Sent & Sentiment Shown for the text message (i.e., Neutral)

Message Received & Replying:

Message received with neutral sentiments.

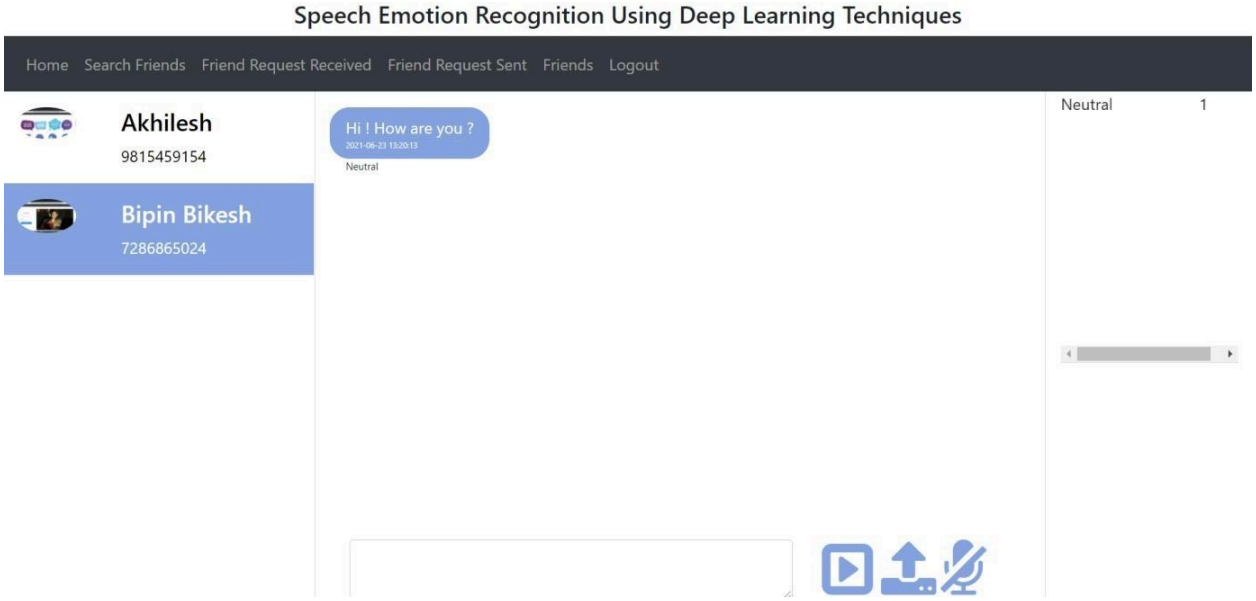

Figure 9.11: Message Received with Neutral Sentiments.

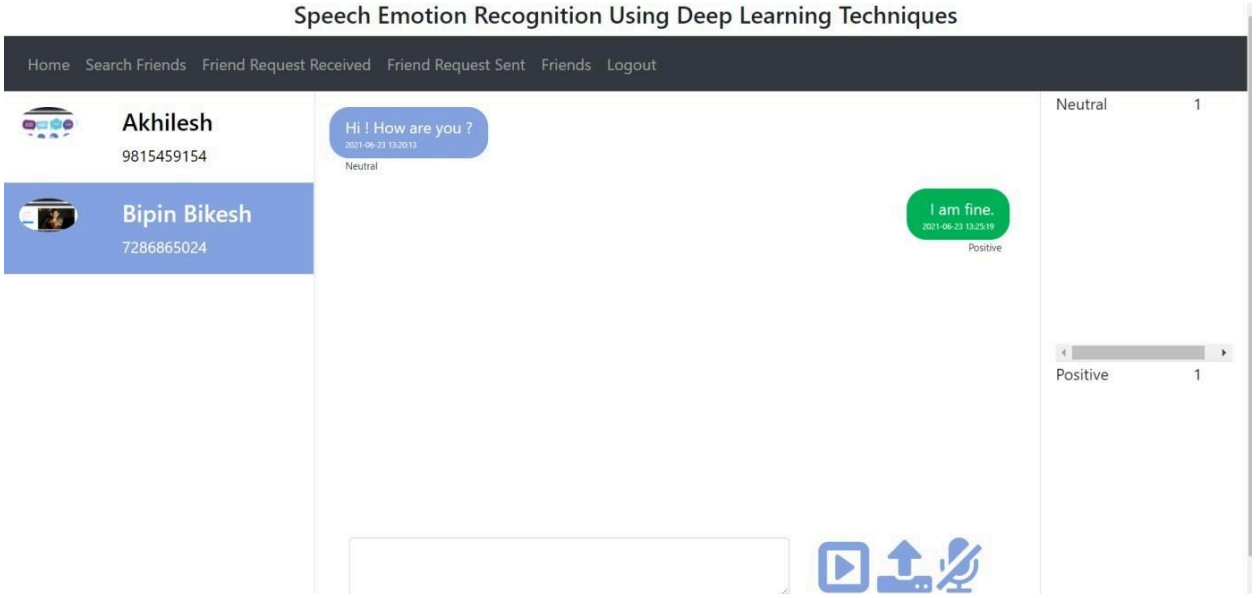

Figure 9.12:Replying to the message and the sentiment is showing Positive

Sending an audio file:

The result of the Audio file is predicted with an Emotion (i.e.,male_angry)

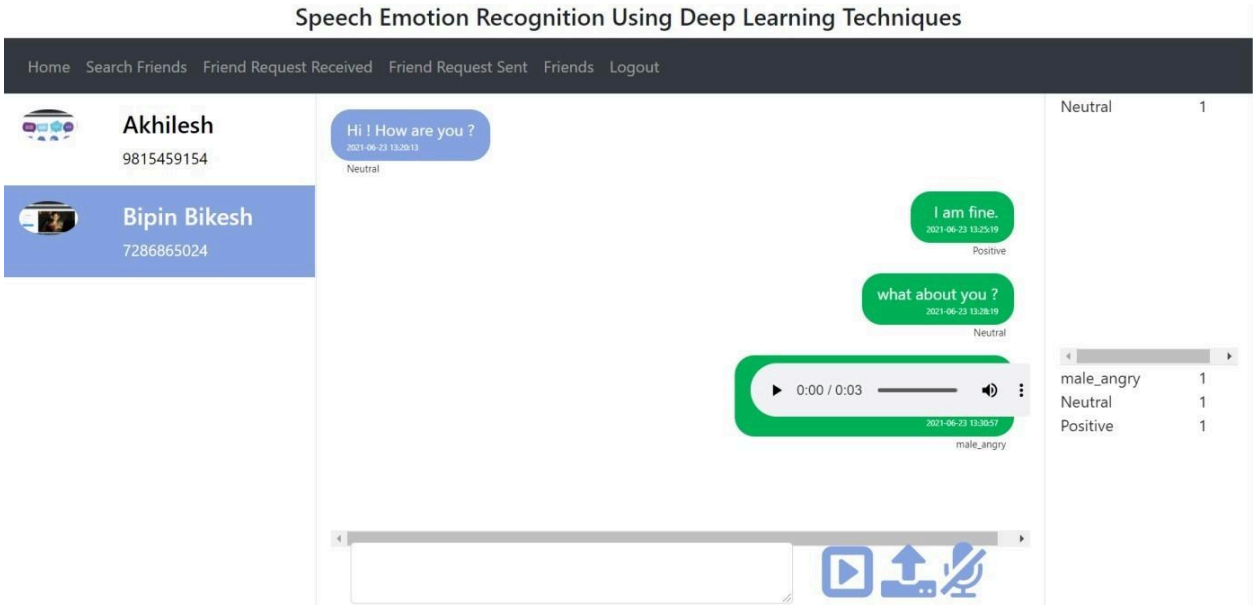

Figure: 9.13: Sending an Audio file

Logout Page:

While clicking the logout page return to the main page called the home page.

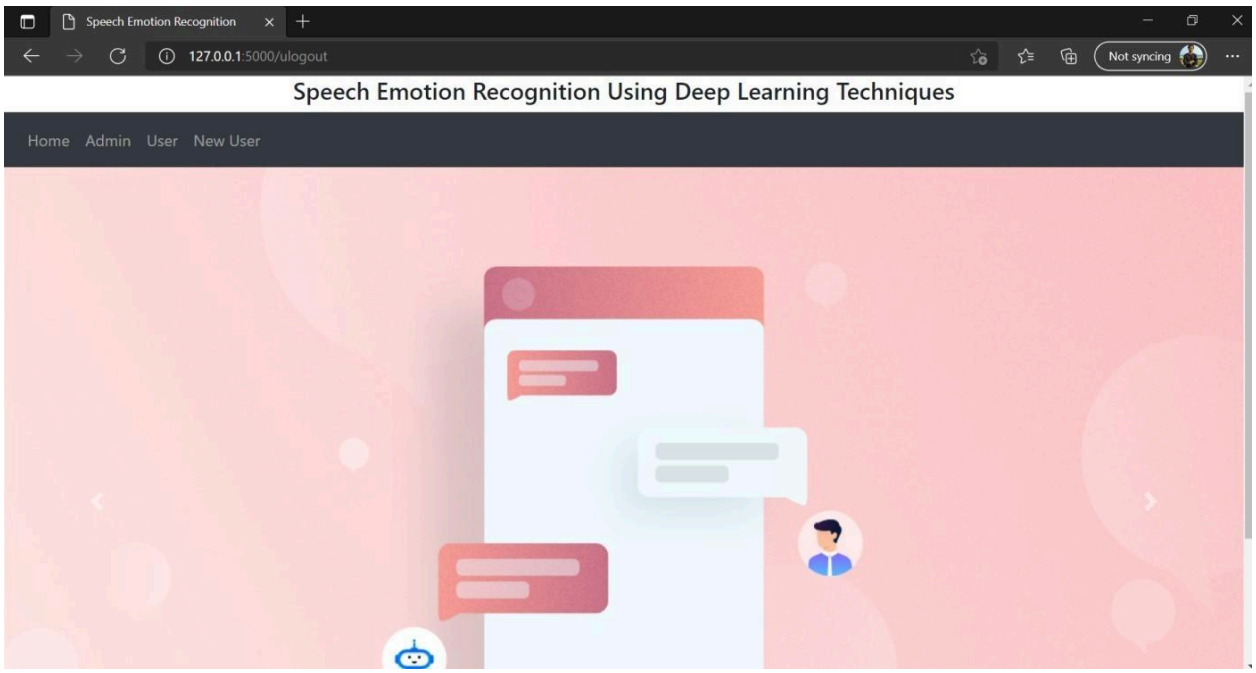

Figure: 9.14: Logout page

10. Conclusion & Future Scope

10.1 Conclusion

We observed that the model is more accurate in predicting surprise, angry emotions and sentiments and it also makes sense because audio files of these emotions differ from other audio files in a lot of ways like pitch, speed, and loudness.
We overall achieved 71% accuracy on our test data and it's decent but we can improve it more by applying more augmentation techniques and using other feature extraction methods.

10.2 Future Scope

In this project, we present only the prediction of eight human emotions using text and speech. It can be expanded to predict more human emotions. The CNN classification algorithms wrongly predicted some of the samples belonging to a happy class, and in SVM, RF belonging to the fear class. This can be rectified by extracting more features to better distinguish between these two classes. The plan to further make the Speech Emotion Recognition system more robust & real-time analysis would be done.

REFERENCES

[1] LIVINGSTONE, S. R., AND RUSSO, F. A. : The Ryerson audio-visual database of emotional speech and song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in north american english. PloS one 13, 5 (2018), e0196391.

[2] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh, and K. Shaalan, ‘‘Speech recognition using deep neural networks: A systematic review,’’ IEEE Access, vol. 7, pp. 19143–19165, 2019.

[3] Khalil, Ruhul Amin, et al. "Speech emotion recognition using deep learning techniques: A review." IEEE Access, vol. 7, pp. 117327- 117345, 2019.

[4] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, “Multimodal Deep Learning,” Proc. 28th Int. Conf. Mach. Learn., pp. 689–696, 2011.

[5] J. Rong, G. Li, and Y. P. P. Chen, “Acoustic feature selection for automatic emotion recognition from speech,” Inf. Process. Manag., vol. 45, no. 3, pp. 315–328, 2009.

[6] C.-W. Huang and S. S. Narayanan, “Characterizing Types of Convolution in Deep Convolutional Recurrent Neural Networks for Robust Speech Emotion Recognition,” pp. 1–19, 2017.

[7] H. M. Fayek, M. Lech, and L. Cavedon, “Evaluating deep learning architectures for Speech Emotion Recognition,” Neural Networks, vol. 92, pp. 60–68, 2017.