**GUI Main Page Code:**

import tkinter as tk

from tkinter import ttk, LEFT, END from PIL import Image, ImageTk

##############################################+==============================

===============================

root = tk.Tk() root.configure(background="pink") root.geometry("300x200")

w, h = root.winfo\_screenwidth(), root.winfo\_screenheight() root.geometry("%dx%d+0+0" % (w, h))

root.title("Voice Based Medical Assistant Chatbot")

# ++++++++++++++++++++++++++++++++++++++++++++

#####For background Image image2 = Image.open('b1.jpg')

image2 = image2.resize((w,h), Image.ANTIALIAS)

background\_image = ImageTk.PhotoImage(image2) background\_label = tk.Label(root, image=background\_image) background\_label.image = background\_image background\_label.place(x=0, y=0) # , relwidth=1, relheight=1) #

label\_l2 = tk.Label(root, text="Voice Based Medical Assistant Chatbot",font=("times", 30, 'bold','italic'),

background="green", fg="white", width=70, height=2) label\_l2.place(x=0, y=0)

#logo

img = Image.open('logo.jpg')

img = img.resize((100,75), Image.ANTIALIAS) logo\_image = ImageTk.PhotoImage(img) logo\_label = tk.Label(root, image=logo\_image) logo\_label.image = logo\_image logo\_label.place(x=21, y=10)

**MEDICAL CHATBOT**

# label image

img1 = Image.open('r2.jpg')

img1 = img1.resize((600,645), Image.ANTIALIAS) logo\_image1 = ImageTk.PhotoImage(img1)

logo\_label1 = tk.Label(root, image=logo\_image1,bd=5 ) logo\_label1.image = logo\_image1 logo\_label1.place(x=40, y=100)

frame\_alpr = tk.LabelFrame(root, text=" --Details-- ", width=680, height=625, bd=5, font=('times', 14, ' bold '),bg="pink")

frame\_alpr.place(x=640, y=100)

# label\_l2 = tk.Text(root,font=("Times New Roman", 15, 'italic'),

# background="#220A29", fg="white", width=72, height=29) # label\_l2.place(x=840, y=130)

img = Image.open('r1.jpg')

img = img.resize((200,150), Image.ANTIALIAS) logo\_image = ImageTk.PhotoImage(img)

logo\_label = tk.Label(frame\_alpr, image=logo\_image) logo\_label.image = logo\_image logo\_label.place(x=220,y=50)

label\_l2 = tk.Label(root,text="Voice Based Medical Assistant Chatbot",font=("Times New Roman",20, 'bold','italic'),

background="pink", fg="black") label\_l2.place(x=750, y=360)

def window():

root.destroy() def log():

from subprocess import call call(["python","login.py"])

def register():

from subprocess import call call(["python","register.py"])

button1 = tk.Button(frame\_alpr, text="LOGIN", command=log, width=12, height=1,font=('times 15 bold italic '),bd=5, bg="blue", fg="white")

button1.place(x=250, y=300)

**MEDICAL CHATBOT**

button2 = tk.Button(frame\_alpr, text="REGISTER",command=register,width=12, height=1,font=('times 15 bold italic'), bd=5,bg="blue", fg="white") button2.place(x=250, y=400)

button4 = tk.Button(frame\_alpr, text="EXIT", command=window, width=12, height=1,font=('times 15 bold italic'),bd=5,bg="red", fg="white")

button4.place(x=250, y=500)

# label\_l1 = tk.Label(root, text="\*\* Healthcare Chatbot System @ 2021 by \*\*",font=("Times New Roman", 10, 'bold'),

# background="black", fg="white", width=250, height=2) # label\_l1.place(x=0, y=798)

root.mainloop()

**Registration Page:**

import tkinter as tk

#from tkinter import ttk, LEFT, END from tkinter import messagebox as ms import sqlite3

from PIL import Image, ImageTk import re

##############################################+==============================

===============================

root = tk.Tk() root.configure(background="white") # root.geometry("1300x700")

w, h = root.winfo\_screenwidth(), root.winfo\_screenheight() root.geometry("800x700+200+100" ) root.title("Registration Form")

# 43

# ++++++++++++++++++++++++++++++++++++++++++++

#####For background Image image2 = Image.open('m.jpg')

image2 = image2.resize((w,h), Image.ANTIALIAS)

**MEDICAL CHATBOT**

background\_image = ImageTk.PhotoImage(image2) background\_label = tk.Label(root, image=background\_image) background\_label.image = background\_image background\_label.place(x=0, y=0) # , relwidth=1, relheight=1)

# frame\_alpr = tk.LabelFrame(root, text=" --About us-- ", width=550, height=500, bd=5, font=('times', 14, ' bold '),bg="#7CCD7C")

# frame\_alpr.grid(row=0, column=0, sticky='nw') # frame\_alpr.place(x=550, y=200)

# label\_l2 = tk.Label(root, text=" Registration Form ",font=("Times New Roman", 30, 'bold'), # background="black", fg="white", width=67, height=2)

# label\_l2.place(x=0, y=90)

frame\_alpr = tk.LabelFrame(root, text=" --Register-- ", width=600, height=600, bd=5, font=('times', 14, ' bold '),fg="black",bg="#CD6090")

frame\_alpr.grid(row=0, column=0, sticky='nw') frame\_alpr.place(x=100, y=50) ######################### Registration form

#####################################################################

Fullname = tk.StringVar() address = tk.StringVar() username = tk.StringVar() Email = tk.StringVar() PhoneNo = tk.IntVar() AdharNo = tk.IntVar()

# var = tk.IntVar() # age = tk.IntVar()

password = tk.StringVar() password1 = tk.StringVar()

# database code

db = sqlite3.connect('medical chatbot db.db') cursor = db.cursor()

cursor.execute("CREATE TABLE IF NOT EXISTS registration"

**MEDICAL CHATBOT**

"(Fullname TEXT, address TEXT, username TEXT, Email TEXT, Phoneno TEXT, AdharNo TEXT, password TEXT)")

db.commit()

def password\_check(passwd):

SpecialSym =['$', '@', '#', '%'] val = True

if len(passwd) < 6:

print('length should be at least 6') val = False

if len(passwd) > 20:

print('length should be not be greater than 8') val = False

if not any(char.isdigit() for char in passwd): print('Password should have at least one numeral') val = False

if not any(char.isupper() for char in passwd):

print('Password should have at least one uppercase letter') val = False

if not any(char.islower() for char in passwd):

print('Password should have at least one lowercase letter') val = False

if not any(char in SpecialSym for char in passwd):

print('Password should have at least one of the symbols $@#') val = False

if val:

return val

def insert():

fname = Fullname.get() addr = address.get()

un = username.get() email = Email.get()

**MEDICAL CHATBOT**

mobile = PhoneNo.get() adhar = AdharNo.get() # gender = var.get()

# time = age.get() pwd = password.get()

cnpwd = password1.get()

with sqlite3.connect('medical chatbot db.db') as db: c = db.cursor()

# Find Existing username if any take proper action

find\_user = ('SELECT \* FROM registration WHERE username = ?') c.execute(find\_user, [(username.get())])

# else:

# ms.showinfo('Success!', 'Account Created Successfully !')

# to check mail

#regex = '^\w+([\.-]?\w+)\*@\w+([\.-]?\w+)\*(\.\w{2,3})+$'

regex='^[a-z0-9]+[\.\_]?[a-z0-9]+[@]\w+[.]\w{2,3}$' if (re.search(regex, email)):

a = True else:

a = False # validation

if (fname.isdigit() or (fname == "")): ms.showinfo("Message", "please enter valid name")

elif (addr == ""):

ms.showinfo("Message", "Please Enter Address")

elif (email == "") or (a == False): ms.showinfo("Message", "Please Enter valid email")

elif((len(str(mobile)))<10 or len(str((mobile)))>10): ms.showinfo("Message", "Please Enter 10 digit mobile number")

**MEDICAL CHATBOT**

elif((len(str(adhar)))<12 or len(str((adhar)))>12): ms.showinfo("Message", "Please Enter 12 digit adhar number")

# elif ((time > 100) or (time == 0)):

# ms.showinfo("Message", "Please Enter valid age") elif (c.fetchall()):

ms.showerror('Error!', 'Username Taken Try a Diffrent One.') elif (pwd == ""):

ms.showinfo("Message", "Please Enter valid password") # elif (var == False):

# ms.showinfo("Message", "Please Enter gender") elif(pwd=="")or(password\_check(pwd))!=True:

ms.showinfo("Message", "password must contain atleast 1 Uppercase letter,1 symbol,1 number") elif (pwd != cnpwd):

ms.showinfo("Message", "Password Confirm password must be same") else:

conn = sqlite3.connect('medical chatbot db.db') with conn:

cursor = conn.cursor() cursor.execute(

'INSERT INTO registration(Fullname, address, username, Email, AdharNo, PhoneNo, password) VALUES(?,?,?,?,?,?,?)',

(fname, addr, un, adhar, email, mobile, pwd))

conn.commit() db.close()

ms.showinfo('Success!', 'Account Created Successfully !') # window.destroy()

root.destroy()

from subprocess import call call(["python", "login.py"])

################################################################################## ###################################################################

l1 = tk.Label(frame\_alpr, text=" Registration Form ", font=("Times new roman", 30, "bold","italic"),bd=5, bg="#CD6090", fg="black")

**MEDICAL CHATBOT**

l1.place(x=120, y=10)

# that is for label1 registration

l2 = tk.Label(frame\_alpr, text="Full Name :", width=12, font=("Times new roman", 15, "bold"),bd=5, fg="black")

l2.place(x=100, y=100)

t1 = tk.Entry(frame\_alpr, textvar=Fullname, width=20, font=('', 15)) t1.place(x=300, y=100)

# that is for label 2 (full name)

l3 = tk.Label(frame\_alpr, text="Address :", width=12, font=("Times new roman", 15, "bold"),bd=5, fg="black")

l3.place(x=100, y=150)

t2 = tk.Entry(frame\_alpr, textvar=address, width=20, font=('', 15)) t2.place(x=300, y=150)

# that is for label 3(address)

# that is for label 4(blood group)

l4 = tk.Label(frame\_alpr, text="E-mail :", width=12, font=("Times new roman", 15, "bold"), bd=5,fg="black")

l4.place(x=100, y=200)

t3 = tk.Entry(frame\_alpr, textvar=Email, width=20, font=('', 15)) t3.place(x=300, y=200)

# that is for email address

l5 = tk.Label(frame\_alpr, text="Phone number :", width=12, font=("Times new roman", 15, "bold"),bd=5, fg="black")

l5.place(x=100, y=250)

t4 = tk.Entry(frame\_alpr, textvar=PhoneNo, width=20, font=('', 15)) t4.place(x=300, y=250)

# phone number

l6 = tk.Label(frame\_alpr, text="Adhar number :", width=12, font=("Times new roman", 15, "bold"),bd=5, fg="black")

l6.place(x=100, y=300)

t5 = tk.Entry(frame\_alpr, textvar=AdharNo, width=20, font=('', 15)) t5.place(x=300, y=300)

**MEDICAL CHATBOT**

# l7 = tk.Label(frame\_alpr, text="Gender :", width=12, font=("Times new roman", 15, "bold"), bg="snow")

# l7.place(x=10, y=250) # # gender

# tk.Radiobutton(frame\_alpr, text="Male", padx=5, width=5, bg="snow", font=("bold", 15), variable=var, value=1).place(x=80,

# y=250)

# tk.Radiobutton(frame\_alpr, text="Female", padx=20, width=4, bg="snow", font=("bold", 15), variable=var, value=2).place(

# x=120, y=250)

# l8 = tk.Label(frame\_alpr, text="Age :", width=12, font=("Times new roman", 15, "bold"), bg="snow")

# l8.place(x=10, y=400)

# t6 = tk.Entry(frame\_alpr, textvar=age, width=20, font=('', 15)) # t6.place(x=80, y=400)

l7 = tk.Label(frame\_alpr, text="User Name :", width=12, font=("Times new roman", 15, "bold"), bd=5,fg="black")

l7.place(x=100, y=350)

t6 = tk.Entry(frame\_alpr, textvar=username, width=20, font=('', 15)) t6.place(x=300, y=350)

l8 = tk.Label(frame\_alpr, text="Password :", width=12, font=("Times new roman", 15, "bold"),bd=5, fg="black")

l8.place(x=100, y=400)

t7 = tk.Entry(frame\_alpr, textvar=password, width=20, font=('', 15), show="\*") t7.place(x=300, y=400)

l9 = tk.Label(frame\_alpr, text="Confirm Password:", width=13, font=("Times new roman", 15, "bold"),bd=5, fg="black")

l9.place(x=100, y=450)

t8 = tk.Entry(frame\_alpr, textvar=password1, width=20, font=('', 15), show="\*") t8.place(x=300, y=450)

**MEDICAL CHATBOT**

btn = tk.Button(frame\_alpr, text="Register", bg="#FAEBD7",font=("times new roman",20,"bold"),fg="black", width=9, height=1, bd=5,command=insert) btn.place(x=220, y=500)

def log1():

from subprocess import call call(["python","login.py"]) root.destroy()

def window():

root.destroy() # def register():

# from subprocess import call # call(["python","register.py"])

# label\_l1 = tk.Label(root, text="\*\* Music Recommendation System @ 2021 by

\*\*",font=("Times New Roman", 10, 'bold'),

# background="black", fg="white", width=250, height=2) # label\_l1.place(x=0, y=800)

root.mainloop()

Login Page:

import tkinter as tk

from tkinter import ttk, LEFT, END from tkinter import messagebox as ms import sqlite3

from PIL import Image, ImageTk import re

##############################################+==============================

===============================

root = tk.Tk() root.configure(background="black") # root.geometry("1300x700")

w, h = root.winfo\_screenwidth(), root.winfo\_screenheight() root.geometry("700x700+200+100")

root.title("Login Form") username = tk.StringVar() password = tk.StringVar()

**MEDICAL CHATBOT**

# ++++++++++++++++++++++++++++++++++++++++++++

#####For background Image image2 = Image.open('b.jpg')

image2 = image2.resize((w,h), Image.ANTIALIAS)

background\_image = ImageTk.PhotoImage(image2) background\_label = tk.Label(root, image=background\_image) background\_label.image = background\_image background\_label.place(x=0, y=0) # , relwidth=1, relheight=1) def registration():

from subprocess import call call(["python","register.py"]) root.destroy()

def login():

##from subprocess import call ###call(["python","login.py"])

# Establish Connection

with sqlite3.connect('medical chatbot db.db') as db: c = db.cursor()

# Find user If there is any take proper action db = sqlite3.connect('medical chatbot db.db') cursor = db.cursor()

cursor.execute("CREATE TABLE IF NOT EXISTS registration"

"(Fullname TEXT, address TEXT, username TEXT, Email TEXT, Phoneno TEXT,Gender TEXT,age TEXT , password TEXT)")

db.commit()

find\_entry = ('SELECT \* FROM registration WHERE username = ? and password = ?') c.execute(find\_entry, [(username.get()), (password.get())])

result = c.fetchall()

if result:

msg = ""

# self.logf.pack\_forget()

# self.head['text'] = self.username.get() + '\n Loged In'

**MEDICAL CHATBOT**

# msg = self.head['text']

# self.head['pady'] = 150 print(msg)

ms.showinfo("messege", "LogIn sucessfully")

# ===========================================

root.destroy()

from subprocess import call call(['python','CHAT\_1 .py'])

# ================================================

else:

ms.showerror('Oops!', 'Username Or Password Did Not Found/Match.')

# frame\_alpr = tk.LabelFrame(root, text=" --About us-- ", width=550, height=500, bd=5, font=('times', 14, ' bold '),bg="#7CCD7C")

# frame\_alpr.grid(row=0, column=0, sticky='nw') # frame\_alpr.place(x=550, y=200)

# label\_l2 = tk.Label(root, text=" Login Form ",font=("Times New Roman", 30, 'bold'), # background="#EEEE00", fg="black", width=67, height=3)

# label\_l2.place(x=0, y=90) #bg1\_icon=ImageTk.PhotoImage(file="L.jpg")

bg\_icon=ImageTk.PhotoImage(file="L.jpg") user\_icon=ImageTk.PhotoImage(file="l1.png") pass\_icon=ImageTk.PhotoImage(file="p1.jpg")

# bg\_lbl=tk.Label(root,image=bg1\_icon, width=600,height=600) # bg\_lbl.place(x=50,y=50)

title=tk.Label(root, text="Login Here", font=("Algerian", 30, "bold","italic"),bd=5,bg="black",fg="white") title.place(x=200,y=150,width=250)

Login\_frame=tk.Frame(root,bg="white") Login\_frame.place(x=100,y=300)

**MEDICAL CHATBOT**

logolbl=tk.Label(Login\_frame,image=bg\_icon,bd=0).grid(row=0,columnspan=2,pady=20)

lbluser=tk.Label(Login\_frame,text="Username",image=user\_icon,compound=LEFT,font=("Times new roman", 20, "bold"),bg="white").grid(row=1,column=0,padx=20,pady=10) txtuser=tk.Entry(Login\_frame,bd=5,textvariable=username,font=("",15)) txtuser.grid(row=1,column=1,padx=20)

lblpass=tk.Label(Login\_frame,text="Password",image=pass\_icon,compound=LEFT,font=("Times new roman", 20, "bold"),bg="white").grid(row=2,column=0,padx=50,pady=10) txtpass=tk.Entry(Login\_frame,bd=5,textvariable=password,show="\*",font=("",15)) txtpass.grid(row=2,column=1,padx=20)

btn\_log=tk.Button(Login\_frame,text="Login",command=login,width=15,font=("Times new roman", 14, "bold"),bg="Green",fg="black")

btn\_log.grid(row=3,column=1,pady=10) btn\_reg=tk.Button(Login\_frame,text="Create Account",command=registration,width=15,font=("Times new roman", 14, "bold"),bg="red",fg="black")

btn\_reg.grid(row=3,column=0,pady=10 # Login Function

# # def log1():

# # from subprocess import call

# # call(["python","GUI\_main.py"]) # # root.destroy()

# # def window():

# # root.destroy() #def log():

# from subprocess import call # call(["python","login.py"])

# def register():

# from subprocess import call # call(["python","register.py"])

# button1 = tk.Button(label\_l1, text="HOME", command=log1, width=12, height=1,font=('times 15 bold italic underline'),bd=0, bg="#610B4B", fg="white")

# button1.place(x=1190, y=50)

**MEDICAL CHATBOT**

# button2 = tk.Button(label\_l1, text="LOGIN",command=log,width=12, height=1,font=('times 15 bold italic underline'), bd=0,bg="#610B4B", fg="white")

# button2.place(x=1310, y=50)

# button4 = tk.Button(label\_l1, text="REGISTER", command=register, width=12, height=1,font=('times 15 bold italic underline'),bd=0,bg="#610B4B", fg="white") # button4.place(x=1430, y=50)

# label\_l1 = tk.Label(root, text="\*\* Music Recommendation System @ 2021 by

\*\*",font=("Times New Roman", 10, 'bold'),

# background="black", fg="white", width=250, height=2) # label\_l1.place(x=0, y=800)

root.mainloop()

Chat Page Code:

import nltk

from textblob import TextBlob

from nltk.stem import WordNetLemmatizer lemmatizer = WordNetLemmatizer() import pickle

import numpy as np

from PIL import Image , ImageTk from keras.models import load\_model

model = load\_model('TRIAL\_MODEL.h5') import json

import random import sqlite3 import tkinter as tk import pyttsx3 import PIL.Image

import speech\_recognition as sr

converter = pyttsx3.init() #text to spech converter.setProperty('rate', 150)### voice rate converter.setProperty('volume', 1.0)

intents = json.loads(open('intents2.json').read())### q & ans file words = pickle.load(open('WORDS.pkl','rb'))##treaning model

**MEDICAL CHATBOT**

classes = pickle.load(open('CLASSES.pkl','rb')) def clean\_up\_sentence(sentence):

# tokenize the pattern - split words into array sentence\_words = nltk.word\_tokenize(sentence) # stem each word - create short form for word

sentence\_words = [lemmatizer.lemmatize(word.lower()) for word in sentence\_words] return sentence\_words

# return bag of words array: 0 or 1 for each word in the bag that exists in the sentence

def bow(sentence, words, show\_details=True):###words or features check # tokenize the pattern

sentence\_words = clean\_up\_sentence(sentence)

# bag of words - matrix of N words, vocabulary matrix bag = [0]\*len(words)

for s in sentence\_words:

for i,w in enumerate(words): if w == s:

# assign 1 if current word is in the vocabulary position bag[i] = 1

if show\_details:

print ("found in bag: %s" % w) return(np.array(bag))

def predict\_class(sentence, model):

# filter out predictions below a threshold

p = bow(sentence, words,show\_details=False)

res = model.predict(np.array([p]))[0] ## check probablity of that words ERROR\_THRESHOLD = 0.25

results = [[i,r] for i,r in enumerate(res) if r>ERROR\_THRESHOLD] # sort by strength of probability

results.sort(key=lambda x: x[1], reverse=True) return\_list = []

for r in results:

return\_list.append({"intent": classes[r[0]], "probability": str(r[1])}) return return\_list

**MEDICAL CHATBOT**

def getResponse(ints, intents\_json):

tag = ints[0]['intent'] ####for gating replay list\_of\_intents = intents\_json['intents']

for i in list\_of\_intents: if(i['tag']== tag):

result = random.choice(i['responses']) break

return result

def chatbot\_response(msg):

ints = predict\_class(msg, model) res = getResponse(ints, intents) if 'CHICKEN POX' in res:

db = sqlite3.connect(r'Doctors.db') c = db.cursor()

find = ("SELECT Name,Contact,Address FROM List WHERE Disease='CHICKEN POX'") c.execute(find)

d = c.fetchall() print(res , d) return res , d

if 'FEVER' in res:

db = sqlite3.connect(r'Doctors.db') c = db.cursor()

find = ("SELECT Name,Contact,Address FROM List WHERE Disease='FEVER'") c.execute(find)

d = c.fetchall() print(res , d) return res , d

if 'COUGH' in res:

db = sqlite3.connect(r'Doctors.db') c = db.cursor()

find = ("SELECT Name,Contact,Address FROM List WHERE Disease='COUGH'") c.execute(find)

d = c.fetchall() print(res , d) return res , d

**MEDICAL CHATBOT**

if 'COLD' in res:

db = sqlite3.connect(r'Doctors.db') c = db.cursor()

find = ("SELECT Name,Contact,Address FROM List WHERE Disease='COLD'") c.execute(find)

d = c.fetchall() print(res , d) return res , d

if 'Contact Dermatitis' in res:

db = sqlite3.connect(r'Doctors.db') c = db.cursor()

find = ("SELECT Name,Contact,Address FROM List WHERE Disease='Contact Dermatitis'") c.execute(find)

d = c.fetchall() print(res , d) return res , d

if 'Eye Allergies' in res:

db = sqlite3.connect(r'Doctors.db') c = db.cursor()

find = ("SELECT Name,Contact,Address FROM List WHERE Disease='Eye Allergies'") c.execute(find)

d = c.fetchall() print(res , d) return res , d

if 'Sinus Infection' in res:

db = sqlite3.connect(r'Doctors.db') c = db.cursor()

find = ("SELECT Name,Contact,Address FROM List WHERE Disease='Sinus Infection'") c.execute(find)

d = c.fetchall() print(res , d) return res , d

if 'Allergic Rhinitis' in res:

db = sqlite3.connect(r'Doctors.db') c = db.cursor()

find = ("SELECT Name,Contact,Address FROM List WHERE Disease='Allergic Rhinitis'")

**MEDICAL CHATBOT**

c.execute(find) d = c.fetchall() print(res , d) return res , d

if 'Food Allergy' in res:

db = sqlite3.connect(r'Doctors.db') c = db.cursor()

find = ("SELECT Name,Contact,Address FROM List WHERE Disease='Food Allergy'") c.execute(find)

d = c.fetchall() print(res , d) return res , d

if 'Anaphylaxis' in res:

db = sqlite3.connect(r'Doctors.db') c = db.cursor()

find = ("SELECT Name,Contact,Address FROM List WHERE Disease='Anaphylaxis'") c.execute(find)

d = c.fetchall() print(res , d) return res , d

if 'Acne' in res:

db = sqlite3.connect(r'Doctors.db') c = db.cursor()

find = ("SELECT Name,Contact,Address FROM List WHERE Disease='Acne'") c.execute(find)

d = c.fetchall() print(res , d) return res , d

if 'Eczema' in res:

db = sqlite3.connect(r'Doctors.db') c = db.cursor()

find = ("SELECT Name,Contact,Address FROM List WHERE Disease='Eczema'") c.execute(find)

d = c.fetchall()

**MEDICAL CHATBOT**

print(res , d) return res , d

if 'hives' in res:

db = sqlite3.connect(r'Doctors.db') c = db.cursor()

find = ("SELECT Name,Contact,Address FROM List WHERE Disease='hives'") c.execute(find)

d = c.fetchall() print(res , d) return res , d

if 'dark circles' in res:

db = sqlite3.connect(r'Doctors.db') c = db.cursor()

find = ("SELECT Name,Contact,Address FROM List WHERE Disease='dark circles'") c.execute(find)

d = c.fetchall() print(res , d) return res , d

if 'Blackheads' in res:

db = sqlite3.connect(r'Doctors.db') c = db.cursor()

find = ("SELECT Name,Contact,Address FROM List WHERE Disease='Blackheads'") c.execute(find)

d = c.fetchall() print(res , d) return res , d

if 'psoriasis' in res:

db = sqlite3.connect(r'Doctors.db') c = db.cursor()

find = ("SELECT Name,Contact,Address FROM List WHERE Disease='psoriasis'") c.execute(find)

d = c.fetchall()

**MEDICAL CHATBOT**

print(res , d) return res , d

if 'dry, cracked skin' in res:

db = sqlite3.connect(r'Doctors.db') c = db.cursor()

find = ("SELECT Name,Contact,Address FROM List WHERE Disease='dry, cracked skin'") c.execute(find)

d = c.fetchall() print(res , d) return res , d

if 'ulcers' in res:

db = sqlite3.connect(r'Doctors.db') c = db.cursor()

find = ("SELECT Name,Contact,Address FROM List WHERE Disease='ulcers'") c.execute(find)

d = c.fetchall() print(res , d) return res , d

if 'rosacea' in res:

db = sqlite3.connect(r'Doctors.db') c = db.cursor()

find = ("SELECT Name,Contact,Address FROM List WHERE Disease='rosacea'") c.execute(find)

d = c.fetchall() print(res , d) return res , d

if 'open sores or lesions' in res:

db = sqlite3.connect(r'Doctors.db') c = db.cursor()

find = ("SELECT Name,Contact,Address FROM List WHERE Disease='open sores or lesions'") c.execute(find)

d = c.fetchall() print(res , d)

**MEDICAL CHATBOT**

return res , d

if 'ringworm' in res:

db = sqlite3.connect(r'Doctors.db') c = db.cursor()

find = ("SELECT Name,Contact,Address FROM List WHERE Disease='ringworm'") c.execute(find)

d = c.fetchall() print(res , d) return res , d

else:

print(res) return res

#Creating GUI with tkinter import tkinter

from tkinter import \*

def send():

msg = EntryBox.get("1.0",'end-1c').strip() EntryBox.delete("0.0",END)

if msg != '': ChatLog.configure(state=NORMAL) ChatLog.insert(END, "You: " + msg + '\n\n')

ChatLog.config(foreground="#442265", font=("Verdana", 12 ))

res = chatbot\_response(msg) ChatLog.insert(END, "Bot: \n" + str(res) + '\n')

ChatLog.config(state=DISABLED) ChatLog.yview(END)

from translate import Translator def To\_English():

global file

pass\_text = detect\_text()

**MEDICAL CHATBOT**

en\_blob = TextBlob(str(pass\_text)) translated = (en\_blob.translate(to='en')) print(translated)

To\_English\_label = tk.Label(root,text=str(translated),font=('Times New Roman',12,'italic'),width=47,height=30,bg='green4',fg='white')

To\_English\_label.place(x=905,y=50)

def SPEECH():

r = sr.Recognizer()

with sr.Microphone() as source: print("Say Something...") audio = r.listen(source)

text = r.recognize\_google(audio)

import time #language = 'mr' #Lang1=c.get()

#translator= Translator(from\_lang=Lang1,to\_lang=language) #translation = translator.translate(text)

global file

#en\_blob = TextBlob(str(text)) #translated = (en\_blob.translate(to='en')) #print(translated)

time.sleep(1)

translator= Translator(from\_lang="English",to\_lang="English") translation = translator.translate(text)

print(translation) print(type(translation))

print('You Said : {}'.format(translation)) # en\_blob = TextBlob(str(translation))

# translated = (en\_blob.translate(to='en')) # translated = str(translated)

# print(str(translated)) # print(type(translated))

**MEDICAL CHATBOT**

#msg = listen() # print(msg)

# en\_blob = TextBlob(str(msg))

# translated = (en\_blob.translate(to='en')) # print(translated)

language = 'mr' # Lang1=c.get()

# translator= Translator(from\_lang=Lang1,to\_lang='en-US') # translation = translator.translate(msg)

# print(translation) EntryBox.delete("0.0",END)

if translation != '': ChatLog.configure(state=NORMAL) ChatLog.insert(END, "You: " + translation + '\n\n')

ChatLog.config(foreground="#442265", font=("Verdana", 12 ))

res = chatbot\_response(translation) ChatLog.insert(END, "Bot:\n " + "\n"+ str(res)+ '\n\n') converter.say(str(res))

converter.runAndWait() ChatLog.config(state=DISABLED) ChatLog.yview(END)

def eng\_mr():

import speech\_recognition as sr from googletrans import Translator

from translate import Translator as trans

#from customer.chat import get\_response,bot\_name from gtts import gTTS

import os

r = sr.Recognizer() print("Please talk")

with sr.Microphone() as source:

**MEDICAL CHATBOT**

# read the audio data from the default microphone audio\_data = r.record(source, duration=10) print("Recognizing...")

# convert speech to text text = r.recognize\_google(audio\_data) print("Recognised Speech:" + text)

a=text

translator = Translator() source\_lan = "mr" translated\_to= "hi"

translated\_text = translator.translate(text, src=source\_lan, dest = translated\_to) res1=translated\_text.text

print(translated\_text.text) translator1 = Translator() source\_lan1 = "hi" translated\_to1= "en"

translated\_text1 = translator1.translate(res1, src=source\_lan1, dest = translated\_to1)

print(translated\_text1.text) a\_res=translated\_text1.text

#translator5 = trans(from\_lang="en", to\_lang="hi") #data3 = translator5.translate(text)

result=chatbot\_response(a\_res) #result="hello i am robot" print(result)

translator2 = Translator() source\_lan2 = "en" translated\_to2= "mr"

translated\_text2 = translator2.translate(result, src=source\_lan2, dest = translated\_to2)

print(translated\_text2.text) final=translated\_text2.text

#print(r) context={"user":res1,"bot":final} print(context)

**MEDICAL CHATBOT**

TTS = gTTS(text=str(final)) TTS.save("voice.mp4") os.system("voice.mp4") EntryBox.delete("0.0",END)

if res1 != '': ChatLog.configure(state=NORMAL) ChatLog.insert(END, "You: " + res1 + '\n\n')

ChatLog.config(foreground="#442265", font=("Verdana", 12 ))

#res = chatbot\_response(a\_res)

ChatLog.insert(END, "Bot:\n " + "\n"+ str(final)+ '\n\n') converter.say(str(final))

converter.runAndWait() ChatLog.config(state=DISABLED) ChatLog.yview(END)

def eng\_hi():

import speech\_recognition as sr from googletrans import Translator

from translate import Translator as trans

#from customer.chat import get\_response,bot\_name from gtts import gTTS

import os

r = sr.Recognizer() print("Please talk")

with sr.Microphone() as source:

# read the audio data from the default microphone audio\_data = r.record(source, duration=10) print("Recognizing...")

# convert speech to text text = r.recognize\_google(audio\_data) print("Recognised Speech:" + text)

a=text

**MEDICAL CHATBOT**

# translator = Translator() # source\_lan = "mr"

# translated\_to= "hi"

# translated\_text = translator.translate(text, src=source\_lan, dest = translated\_to) # res=translated\_text.text

# print(translated\_text.text)

translator1 = Translator() source\_lan1 = "hi" translated\_to1= "en"

translated\_text1 = translator1.translate(text, src=source\_lan1, dest = translated\_to1) print(translated\_text1.text)

a\_res=translated\_text1.text

translator5 = trans(from\_lang="en", to\_lang="hi") data3 = translator5.translate(text) result=chatbot\_response(a\_res)

#result="hello i am robot" print(result)

translator2 = Translator() source\_lan2 = "en" translated\_to2= "hi"

translated\_text2 = translator2.translate(result, src=source\_lan2, dest = translated\_to2) print(translated\_text2.text)

final=translated\_text2.text

#print(r) context={"user":data3,"bot":final} print(context)

TTS = gTTS(text=str(final)) TTS.save("voice.mp4") os.system("voice.mp4")

if data3 != '': ChatLog.configure(state=NORMAL) ChatLog.insert(END, "You: " + data3 + '\n\n')

ChatLog.config(foreground="#442265", font=("Verdana", 12 ))

**MEDICAL CHATBOT**

#res = chatbot\_response(a\_res)

ChatLog.insert(END, "Bot:\n " + "\n"+ str(final)+ '\n\n') converter.say(str(final))

converter.runAndWait() ChatLog.config(state=DISABLED) ChatLog.yview(END)

base = tk.Toplevel() base.title("Medical Chat bot")

base.configure(background="#A9A9A9") #base.geometry("600x600") base.resizable(width=tk.TRUE, height=tk.TRUE)

w, h = base.winfo\_screenwidth(), base.winfo\_screenheight() base.geometry("%dx%d+0+0" % (w, h))

load\_bg = PIL.Image.open(r"img1.jpg") load\_bg = load\_bg.resize((230,200)) render\_bg = ImageTk.PhotoImage(load\_bg) bg = tk.Label(base, image = render\_bg) bg.image = render\_bg bg.place(x=1000,y=125)

##head\_icon=ImageTk.PhotoImage(file="icon1.jpg") ###########Create Chat window

ChatLog = tk.Text(base, bd=2, bg="white",fg="black", height="10", width="150", font="Arial",) #ChatLog.config(state=DISABLED)

#Bind scrollbar to Chat window

scrollbar = tk.Scrollbar(base, command=ChatLog.yview, cursor="heart")

ChatLog['yscrollcommand'] = scrollbar.set

#Create Button to send message

# SendButton = tk.Button(base,text="Send",command=send,font=('Times New Roman',15,'bold'),width=20,bg='green',fg='white')

**MEDICAL CHATBOT**

# SendButton.place(x=550,y=500) #Create the box to enter message

EntryBox = tk.Text(base, bd=1, bg="pink",width="40", height="1", font=("Arial",15)) #EntryBox.bind("<Return>", send)

SendButton = tk.Button(base,text="Send",command=send,font=('Times New Roman',15,'bold'),width=10,bg='green',fg='white', bd=4) SendButton.place(x=335,y=636)

c=StringVar()

#Place all components on the screen

head = tk.Label(base,width=70,text="Voice Based Medical Assistant ChatBot",font=("times",30,'bold',"italic"),bg="#8B0A50",fg="White",height=2) head.place(x=0,y=0)

scrollbar.place(x=866,y=99, height=435) ChatLog.place(x=0,y=99, height=435, width=866) #SendButton.place(x=730, y=500, height=45) EntryBox.place(x=0, y=536, height=145, width=883) head = tk.Label(base,width=19,text="Voice

Communication",font=("times",23,"italic"),bg="black",fg="white") head.place(x=935,y=352)

label\_1 = Label(base, text="Select Language",height=2,width=20,font=("bold", 10),bg='Olive',fg='black')

label\_1.place(x=924,y=428)

###list1 = ['English','Marathi','Hindi'] list1 = ['English']; droplist=OptionMenu(base,c, \*list1) droplist.config(height=2,width=20) c.set('Select language') droplist.place(x=1120,y=424)

#Button(frame\_alpr1, text='After Selecting Language... Press Button and Talk. ',height=5,width=50,font=('times', 14, ' bold

'),bg='brown',fg='white',command=translate\_text).place(x=30,y=180)

button2 = tk.Button(base,text="... Press Button and Talk. ",command=SPEECH,font=('Times New

Roman',15,'bold'),width=20,bg='green',fg='white', bd=4) button2.place(x=980,y=522)

# button2 = tk.Button(base,text=" Press Button and Talk in

Hindi ",command=eng\_hi,font=('Times New Roman',15,'bold'),width=20,bg='green',fg='white')

**MEDICAL CHATBOT**

# button2.place(x=950,y=625) import tkinter as tk

from tkinter import ttk, LEFT, END from PIL import Image, ImageTk

from tkinter.filedialog import askopenfilename from tkinter import messagebox as ms

import cv2 import sqlite3 import os

import numpy as np import time

# def prediction\_emotion():

# #clear\_img()

# #update\_label("Model Training Start ")

# start = time.time()

# result = validate.files\_count() # #validate.files\_count()

# end = time.time()

# #print(" " + result)

# msg = '\n' + str(result) + '\n'

# update\_label(msg) ################################################################################## ###############################

def window(): root.destroy()

# button3 = tk.Button(base, text="Face Recognise",command=evaluation,width=12, height=1,font=('times', 15, ' bold '), bg="brown4", fg="white")

# button3.place(x=1300, y=450)

# button4 = tk.Button(base, text="Prediction",command=prediction\_emotion, width=12, height=1, bg="brown4", fg="white",font=('times', 15, ' bold '))

# button4.place(x=1400, y=400) base.mainloop()