

```

import speech_recognition as sr
import webbrowser
import os
import tkinter as tk
from tkinter import Label, Button, PhotoImage
import threading
from gtts import gTTS
import pygame
import datetime
import requests
import pywhatkit # For YouTube playback

# Initialize pygame for playing audio
pygame.mixer.init()

# Dictionary for websites
websites = {
    "youtube": "https://www.youtube.com",
    "google": "https://www.google.com",
    "github": "https://www.github.com",
    "facebook": "https://www.facebook.com",
    "linkedin": "https://www.linkedin.com"
}

# Dictionary for file opening (Change paths as per your system)
files = {
    "resume": "C:/Users/gaura/Documents/resume.pdf",
    "project": "C:/Users/gaura/Documents/project.docx"
}

# Function to make the bot speak
def speak(text):
    """Speaks the given text using gTTS and plays the audio."""
    try:
        pygame.mixer.music.stop()
        pygame.mixer.quit()

        if os.path.exists("speech.mp3"):
            os.remove("speech.mp3")

        tts = gTTS(text=text, lang='en')
        tts.save("speech.mp3")

        pygame.mixer.init()
        pygame.mixer.music.load("speech.mp3")
        pygame.mixer.music.play()

    except Exception as e:
        print(f"❌ Error in speaking: {e}")

```

```
# Function to animate mic while recording
def start_animation():
    """Show mic recording animation."""
    mic_label.config(image=mic_gif[1])
    root.update()

# Function to stop animation
def stop_animation():
    """Stop mic recording animation."""
    mic_label.config(image=mic_gif[0])
    root.update()

# Function to get current time
def get_time():
    now = datetime.datetime.now().strftime("%I:%M %p")
    return f"The current time is {now}"

# Function to get today's date
def get_date():
    today = datetime.date.today().strftime("%B %d, %Y")
    return f"Today's date is {today}"

# Function to get weather info
def get_weather(city):
    """Fetches weather details for the given city."""
    api_key = "your_api_key" # Replace with your actual OpenWeatherMap API key
    url = f"http://api.openweathermap.org/data/2.5/weather?q={city}&appid={api_key}&units=metric"

    try:
        response = requests.get(url)
        data = response.json()

        if response.status_code == 200:
            temp = data["main"]["temp"]
            desc = data["weather"][0]["description"]
            return f"The weather in {city} is {temp}°C with {desc}."
        else:
            return "Sorry, I couldn't fetch the weather details. Please try again."

    except requests.exceptions.RequestException:
        return "There was a network issue. Please check your internet."

# Function to greet the user
def greet_user():
    """Greet the user on startup."""
    output_label.config(text="👋 Welcome to BotChat!\nHow can I help you?")
    speak("Welcome to BotChat. How can I help you?")

# Function to recognize and process speech
def recognize_speech():
```

```

"""Capture and process voice commands with high accuracy."""
start_animation()
recognizer = sr.Recognizer()

with sr.Microphone() as source:
    status_label.config(text="Listening... Speak now!")
    root.update()
    speak("Listening...")

    recognizer.adjust_for_ambient_noise(source, duration=1)

    try:
        audio = recognizer.listen(source, timeout=5, phrase_time_limit=10)
    except sr.WaitTimeoutError:
        stop_animation()
        output_label.config(text="⚠ No speech detected.")
        speak("I didn't hear anything. Please try again.")
        return

stop_animation()

try:
    command = recognizer.recognize_google(audio, language="en-IN").lower()
    output_label.config(text=f"✅ Task Allocated: {command}")
    speak(f"Task Allocated: {command}")

    # 🎵 Play Song on YouTube
    if "play" in command:
        song = command.replace("play ", "").strip()
        speak(f"Playing {song} on YouTube...")
        pywhatkit.playonyt(song)

    # Open websites dynamically
    elif "open" in command:
        site = command.replace("open ", "").strip()
        if site in websites:
            webbrowser.open(websites[site])
            speak(f"Opening {site}...")
        else:
            webbrowser.open(f"https://www.{site}.com")
            speak(f"Trying to open {site}...")

    # Get current time
    elif "what time is it" in command:
        speak(get_time())

    # Get today's date
    elif "what's today's date" in command:
        speak(get_date())

```

```

# Get current time
elif "what time is it" in command:
    speak(get_time())

# Get today's date
elif "what's today's date" in command:
    speak(get_date())

# Get weather
elif "what's the weather in" in command:
    city = command.replace("what's the weather in ", "").strip()
    speak(get_weather(city))

else:
    speak("Command not recognized. Please try again.")
    output_label.config(text="⚠ Command not recognized!")

except sr.UnknownValueError:
    speak("Sorry, I didn't catch that.")
except sr.RequestError:
    speak("Please check your internet connection.")

# GUI Setup
root = tk.Tk()
root.title("BotChat - Voice Assistant")
root.geometry("400x500")
root.config(bg="#282c34")

mic_gif = [PhotoImage(file="micoff.png"), PhotoImage(file="miconn.png")]

title_label = Label(root, text="🤖 BotChat Voice Assistant", font=("Arial", 16, "bold"), bg="#282c34", fg="white")
title_label.pack(pady=10)

mic_label = Label(root, image=mic_gif[0], bg="#282c34")
mic_label.pack(pady=20)

status_label = Label(root, text="Press the button and start speaking!", font=("Arial", 12), bg="#282c34", fg="lightgrey")
status_label.pack()

record_button = Button(root, text="🎤 Start Recording", font=("Arial", 14, "bold"), command=lambda: threading.Thread(target=recognize_speech, daemon=True).start)
record_button.pack(pady=20)

output_label = Label(root, text="", font=("Arial", 14, "bold"), bg="#282c34", fg="#98c379", wraplength=350, justify="center")
output_label.pack(pady=20)

greet_user()
root.mainloop()

```