```
import os
import datetime
import webbrowser
import pyttsx3
import speech_recognition as sr
import openai
from tkinter import *

openai.api_key = "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"

def ai(prompt):
    response = openai.ChatCompletion.create(
        model="gpt-3.5-turbo",
        messages=[
            {"role": "system", "content": "You are a helpful assistant."},
            {"role": "user", "content": prompt},
        ],
        max_tokens=256,
        n=1,
        stop=None,
        temperature=0.7,
        top_p=1,
        frequency_penalty=0,
        presence_penalty=0
    )
    generated_text = response["choices"][0]["message"]["content"]
    generated_text_label.config(text=generated_text)
    root.update()
    speak(generated_text)
    return generated_text
```

```
def speak(text):
    try:
        engine = pyttsx3.init()
        engine.say(text)
        engine.runAndWait()
    except pyttsx3.EngineError as e:
        print(f"Error in speak function: {e}")
    except pyttsx3.UnknownValueError:
        print("Text-to-speech could not understand the speech.")
    except pyttsx3.RequestError as e:
        print(f"Error in text-to-speech request: {e}")

def take_command():
    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("Listening...")
        r.pause_threshold = 0.5
        audio = r.listen(source)
        try:
            command = r.recognize_google(audio, language="en-in")
            print(f"User  said: {command}")
            return command.lower()
        except sr.UnknownValueError:
            print("Sorry, I didn't understand that.")
            return None
        except sr.RequestError as e:
            print(f"Could not request results from Google Speech Recognition service; {e}")
            return None

def search():
    search_query = search_entry.get()
```

```
    if search_query:
        response = ai(prompt=search_query)
        search_entry.delete(0, END)
    else:
        speak("Please provide a search query.")

def start_listening():
    microphone_button.config(bg="red")
    command = take_command()
    if command:
        response = ai(prompt=command)
        generated_text_label.config(text=response)
    microphone_button.config(bg="SystemButtonFace")

def create_start_button():
    global generated_text_label, search_entry, microphone_button
    search_frame = Frame(root)
    search_frame.pack(pady=80)
    search_entry = Entry(search_frame, width=40)
    search_entry.pack(side=LEFT, padx=5)
    search_entry.bind("<Return>", lambda event: search())
    search_button = Button(search_frame, text="Search", command=search)
    search_button.pack(side=LEFT, padx=5)

    microphone_icon = PhotoImage(file=r"C:\Users\rishi\Downloads\free-microphone-icon-342-thumb.png")
    microphone_icon = microphone_icon.subsample(14)
    microphone_button = Button(search_frame, image=microphone_icon, command=start_listening, bd=0)
    microphone_button.image = microphone_icon
    microphone_button.pack(side=LEFT, padx=5)
    microphone_button.bind("<Enter>", on_enter)
```

```
    microphone_button.bind("<Leave>", on_leave)

    generated_text_label = Label(root, text="")
    generated_text_label.pack()

def on_enter(event):
    microphone_button.config(bg="lightblue")

def on_leave(event):
    microphone_button.config(bg="SystemButtonFace")

def start_program():
    while True:
        user_command = take_command()
        if user_command:
            print(f"User  command recognized: {user_command}")
            if "using ai" in user_command:
                ai(prompt="")
            elif "open google" in user_command:
                root.update()
                speak("Opening Google.")
                webbrowser.open("https://www.google.com")
            elif "open youtube" in user_command:
                root.update()
                speak("Opening YouTube.")
                webbrowser.open("https://www.youtube.com")
            elif "play music" in user_command:
                root.update()
                speak("Playing music.")
                music_path = "C:\\Users\\rishi\\Downloads\\Neeve-SenSongsMp3.Co.mp3"
                if os.path.exists(music_path):
```

```
            os.startfile(music_path)
        else:
            root.update()
            speak("Sorry, the specified music file was not found.")
    elif "what is the time" in user_command:
        current_time = datetime.datetime.now().strftime("%H:%M")
        root.update()
```