**TRƯỜNG ĐẠI HỌC SÀI GÒN**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO**

**ĐỒ ÁN MÔN LẬP TRÌNH PYTHON**

**ĐỀ TÀI:**

**XÂY DỰNG PHẦN MỀM TRỢ LÝ ẢO**

Nhóm 13:

Bùi Nguyễn Trung Kiên 3120560048

Nguyễn Tuấn Vũ 3120410624

*Giảng viên hướng dẫn*: Trịnh Tấn Đạt

*TP. Hồ Chí Minh, ngày 8 tháng 12 năm 2022*

# **NHẬN XÉT CỦA GIẢNG VIÊN**

TP. HCM, ngày… tháng… năm 2022

Giảng viên hướng dẫn

**LỜI MỞ ĐẦU**

Ngày nay, ứng dụng công nghệ thông tin và việc tin học hóa được xem là một trong những yếu tố mang tính quyết định trong hoạt động của các chính phủ, tổ chức, cũng như của các công ty, nó đóng vai trò hết sức quan trọng, có thể tạo ra những bước đột phá mạnh mẽ. Đặt biệt là ngành AI, sự phát triển nhanh chóng của AI và việc phát triển công nghệ nhận dạng giọng nói của công nghệ này đã và đang ngày càng đi sâu vào mọi lĩnh vực trong cuộc sống. Từ những điều trên, cùng với những phân tích bên dưới, ta sẽ chứng minh một cách sinh động rằng tại sao công nghệ điều khiển giọng nói nên được coi là xu thế công nghệ hiện tại không chỉ bới những ông lớn về IT hay những người có niềm đam mê với AI mà các doanh nghiệp cũng đang ráo riết nâng cấp, tối ưu hệ thống bằng công nghệ giọng nói hay còn gọi là trợ lý ảo. Bắt nguồn với ý tưởng này, Bọn em đã thực hiện đồ án:

“XÂY DỰNG PHẦN MỀM TRỢ LÝ ẢO ”

# 

# **MỤC LỤC**

[NHẬN XÉT CỦA GIẢNG VIÊN](#_gjdgxs) 2

[MỤC LỤC](#_23ckvvd) 3

**[CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI](#_sjdwjrwa7xj)** 4

[I. Lý do chọn đề tài:](#_lerp3mc35lrj) 4

[II. Đối tượng hướng tới:](#_cc9fzgkbwbtf) 5

**[CHƯƠNG 2: LÝ THUYẾT CƠ BẢN](#_y6hlw5kz4q99)** 6

[I. Giới thiệu ngôn ngữ lập trình Python](#_fwnw5t3pt52w) 6

[II. Giới thiệu môi trường Visual Studio Code](#_4i1mulquovcp) 6

**[CHƯƠNG 3: PHÂN TÍCH CHỨC NĂNG](#_1bfjrhgun7ds)** 8

**[CHƯƠNG 4: GIAO DIỆN VÀ DEMO](#_oi4hewiax5e2)** 22

**[CHƯƠNG 5:KẾT LUẬN](#_x6zqe1ngcyho)** 28

[I. Những điều đã đạt được](#_7kbiyl4tsnl5) 28

[II. Hướng phát triển cho tương lai](#_ssedenhxmg71) 28

**[TÀI LIỆU THAM KHẢO](#_4cy2iym3j1x7)** 28

# [CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI](https://docs.google.com/document/d/1pBMq_OnytAk68APTzbhrs0Yx5Kck2ZCH/edit" \l "heading=h.1fob9te)

## I. Lý do chọn đề tài:

Sự phát triển nhanh chóng của AI đã giúp phát triển công nghệ nhận dạng giọng nói, công nghệ này ngày càng đi sâu vào mọi lĩnh vực trong cuộc sống. Từ những điều trên, cũng với những phân tích bên dưới, ta sẽ chứng minh một cách sinh động rằng tại sao công nghệ điều khiển giọng nói nên được coi là xu thế công nghệ hiện tại không chỉ bới những ông lớn về IT hay những người có niềm đam mê với AI mà các doanh nghiệp cũng đang ráo riết nâng cấp, tối ưu hệ thống bằng công nghệ giọng nói hay còn gọi là trợ lý ảo. Những năm gần đây, Trợ lý ảo đã dần trở thành xu hướng công nghệ được đón đầu. Các “ông lớn” làng công nghệ thế giới đã cho ra đời các trợ lý ảo, được nhiều người biết đến như: Apple Siri, Google Assistant, Amazon Alexa, Microsoft Cortana.

Phần mềm AI nhận dạng giọng nói được phát triển nhằm giúp cho nhu cầu đời sống trở nên dễ dàng, thuận tiện hơn. Đối với đời sống thực tế giúp đỡ những người bất tiện trong hoạt động chân tay, có thể dễ dàng sử dụng thiết bị điện tử mà không cần phải thao tác nhiều. Từ đó, giúp cho cuộc sống của họ trở nên tốt đẹp hơn. Đối với ngành công nghệ, sự phát triển của công nghệ nhận dạng giọng nói đã góp phần không nhỏ cho ra đời của các công nghệ sau này. Đặc biệt nó là cơ sở để phát triển công nghệ internet of things - thứ công nghệ có tiềm năng đột phá mạnh mẽ trong tương lai.

## II. Đối tượng hướng tới:

Thời đại công nghệ bùng nổ hiện nay, đã giải quyết được nhiều vấn đề tồn đọng trong quá khứ và đặt ra nhiều thách thức mới cho con người. Đặc biệt trong lĩnh vực công nghệ thông tin, các phần mềm được tạo ra đã và đang đáp ứng đúng nhu cầu cần thiết trong cuộc sống của con người. Bên cạnh đó, đối mặt với nhu cầu ngày càng tăng cao của con người đã làm nảy sinh ra các vấn đề mới đòi hỏi các kỹ sư phải sáng tạo, logic và thẩm mỹ. Từ đó, trí tuệ nhân tạo ra đời. Trí tuệ nhân tạo xuất hiện đã làm cho cuộc sống ngày càng thuận tiện hơn. Nó đã thay đổi mọi mặt trong đời sống con người, giúp con người trong hầu hết mọi việc trong cuộc sống. Vì vậy bọn em đối tượng bọn em hướng tới là tất cả mọi người. Ai cũng có thể sử dụng và đặc biệt là những người khuyết tật.

# [CHƯƠNG 2: LÝ THUYẾT CƠ BẢN](https://docs.google.com/document/d/1pBMq_OnytAk68APTzbhrs0Yx5Kck2ZCH/edit" \l "heading=h.nmf14n)

## I. Giới thiệu ngôn ngữ lập trình Python

Pythonđược phát triển để chạy trên nền **Unix**. Nhưng theo thời gian, nó đã "bành trướng" sang mọi hệ điều hành từ **MS-DOS** đến **MAC OS, OS/2, Windows, Linux** và một số điều hành khác thuộc họ Unix.

**Python** cũng là một trong những ngôn ngữ phổ biến nhất thế giới.

Các tính năng của Python bao gồm:

* **Dễ học**: Python có ít từ khóa, cấu trúc đơn giản và cú pháp được định nghĩa rõ ràng. Điều này cho phép người mới học tiếp cận ngôn ngữ một cách nhanh chóng.
* **Dễ đọc**: Mã Python được định nghĩa rõ ràng hơn và có thể nhìn thấy bằng mắt.
* **Dễ bảo trì**: Mã nguồn của Python khá dễ bảo trì.
* **Một thư viện tiêu chuẩn rộng**: Phần lớn thư viện của Python rất dễ đính kèm và đa nền tảng tương thích trên UNIX, Windows và Macintosh.
* **Chế độ tương tác**: Python có hỗ trợ cho chế độ tương tác cho phép kiểm tra tương tác và debug.
* **Portable**: Python có thể chạy trên nhiều nền tảng phần cứng khác nhau và có cùng giao diện trên tất cả các nền tảng.
* **Có thể mở rộng**: Bạn có thể thêm các module cấp thấp vào trình thông dịch Python. Các module này cho phép các lập trình viên thêm hoặc tùy chỉnh các công cụ của mình để hiệu quả hơn.
* **Cơ sở dữ liệu**: Python cung cấp phương thức giao tiếp cho tất cả các cơ sở dữ liệu.
* **Lập trình GUI**: Python hỗ trợ các ứng dụng GUI có thể được tạo và chuyển sang nhiều cuộc gọi hệ thống, thư viện và hệ thống cửa sổ, như Windows MFC, Macintosh và hệ thống X Window của Unix.
* **Khả năng mở rộng**: Python cung cấp cấu trúc và hỗ trợ tốt hơn cho các chương trình lớn hơn so với kịch bản lệnh shell.

## II. Giới thiệu môi trường Visual Studio Code

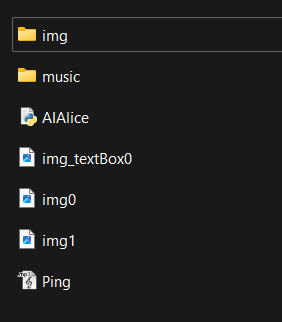
Visual Studio Code là một trong những trình soạn thảo mã nguồn rất phổ biến được các lập trình viên sử dụng. Với các ưu điểm nổi bật là sự nhanh chóng, nhẹ, hỗ trợ đa nền tảng cùng nhiều tính năng và là mã nguồn mở chính. Visual Studio Code ngày càng được ưa chuộng sử dụng, là lựa chọn hàng đầu của các lập trình viên.

Ưu điểm nổi bật:

* Đa dạng ngôn ngữ lập trình giúp người dùng thỏa sức sáng tạo và sử dụng như HTML, CSS, JavaScript, C++,…
* Ngôn ngữ, giao diện tối giản, thân thiện, giúp các lập trình viên dễ dàng định hình nội dung.
* Các tiện ích mở rộng rất đa dạng và phong phú.
* Tích hợp các tính năng quan trọng như tính năng bảo mật (Git), khả năng tăng tốc xử lý vòng lặp (Debug),…
* Đơn giản hóa việc tìm quản lý hết tất cả các Code có trên hệ thống.

# [CHƯƠNG 3:](https://docs.google.com/document/d/1pBMq_OnytAk68APTzbhrs0Yx5Kck2ZCH/edit" \l "heading=h.1rvwp1q) PHÂN TÍCH [CHỨC NĂNG](https://docs.google.com/document/d/1pBMq_OnytAk68APTzbhrs0Yx5Kck2ZCH/edit" \l "heading=h.43ky6rz)

**Cấu trúc thư mục đồ án:**

****

*Hình 1: Cấu trúc thư mục của đồ án*

* Cây thư mục đồ án gồm:
* Thư mục img để chứa hình ảnh thay đổi hình nền
* Thư mục music để chứa nhạc của phần mềm
* file AIAlice.py chứ các hàm quan trọng
* file img\_textBox0.png, img0.png, img1.png là các file ảnh chạy trong giao diện
* file Ping.mp3 chứa âm thanh phát mỗi khi AI nghe lệnh

## Khai báo thư viện:

#Thư viện

#Truy cập web và trình duyệt

from re import search

import webbrowser as wb

#Chuyển âm thanh thành văn bản

import speech\_recognition as sr

import pyttsx3

#Xử lí thởi gian

import time

from datetime import date, datetime

#Lấy thông tin từ web

import requests

import ctypes

import json

import urllib

import urllib.request as urllib2

#Chuyển văn bản thành âm thanh

from gtts import gTTS

#Mở âm thanh

import playsound

#truy cập, xử lí file hệ thống

import os

#Thư viện Tkinter hỗ trợ giao diện

from tkinter.ttk import Frame, Button, Style

from tkinter import \*

from PIL import Image, ImageTk

import tkinter.messagebox as mbox

#Truy cập web, trình duyệt, hỗ trợ tìm kiếm

from selenium import webdriver

from selenium.webdriver.common.keys import Keys

from webdriver\_manager.chrome import ChromeDriverManager

from youtube\_search import YoutubeSearch

from youtubesearchpython import SearchVideos

import wikipedia

#Chọn ngẫu nhiên

import random

#chuyển chữ số sang số

from word2number import w2n #tiếng anh

from vietnam\_number import w2n # tiếng việt

* Thêm các thư viện cần thiết để chạy chương trình trợ lý ảo.
* Ví dụ như thư viện speech\_recognition giúp chuyển âm thanh từ micro thành văn bản để nhận lệnh từ người sử dụng và dễ dàng thực hiện các hàm chức năng và thư viện gTTS, pyttsx3 chuyển văn bản thành âm thanh giúp trợ lý ảo có thể nói những gì mình nghĩ. Các thư viện trên là các thư viện cơ bản để giúp một trợ lý ảo có thể nghe, nói như một con người. Hoặc là thư viện os giúp xử lý các file có sẵn trong hệ thống như mở file,….. Hoặc là thư viện tkinter giúp tạo giao diện đẹp mắt của python.

1. **Khai báo biến:**

wikipedia.set\_lang('vi')

language = 'vi'

robot\_ear = sr.Recognizer()

robot\_speak = pyttsx3.init()

robot\_brain = "Xin chào master"

master = ""

* Khai báo wikipedia với kiểu ngôn ngữ tiếng Việt.
* Khai báo biến robot\_ear nhận dạng giọng nói từ micro máy tính. Biến robot\_ear khởi tạo từ thư viện speech\_recognition
* Khai báo biến robot\_speak khởi tạo trong thư viện pyttsx3. Biến robot\_speak được khởi tạo để chuyển văn bản thành âm thanh.
* Khai báo biến robot\_brain kiểu chuỗi để chứa suy nghĩ robot
* Khai báo biến master kiểu chuoõi để chứa các lệnh của người dùng

1. **Hàm nghe lệnh:**

def get\_audio():

playsound.playsound("Ping.mp3")

time.sleep(2)

print("\nALice: Đang nghe ...")

robot\_ear = sr.Recognizer()

with sr.Microphone() as source:

robot\_ear.pause\_threshold = 2

audio = robot\_ear.listen(source, phrase\_time\_limit=6)

try:

master = robot\_ear.recognize\_google(audio, language="vi-VN")

print("Master: " + master)

text\_area.insert(INSERT,"Master: "+master+"\n")

return master.lower()

except:

print("\n")

return ""

* Hàm get\_audio() giúp trợ lý ảo nghe lệnh qua mic trên máy tính.
* **Ý tưởng:** Mỗi khi nghe lệnh sẽ có tiếng Ping để nhận biết. Sau tiếng Ping, sẽ bật micro trong máy để nghe lệnh từ người dùng (giới hạn khoảng thời gian nghe lệnh là 6 giây). Sau khi nghe lệnh sẽ chuyển âm thanh thành văn bản để thực hiện các hàm chức năng theo yêu cầu. Không nhận dạng được âm thanh thành văn bản sẽ trả về chuỗi rỗng ban đầu.
* **Thực hiện:** Bật file Ping.mp3 có sẵn bằng thư viện playsound. Biến robot\_ear nhận được lệnh chạy thư viện speech\_recognition cùng micro. Tạo biến audio chứa âm thanh vừa nghe được. Sau đó, tiến hành xử lý ngoại lệ cho audio. Lệnh chạy chương trình nhận dạng âm thanh của google, nếu nhận dạng thành công sẽ biến âm thanh thành văn bản và gán vào biến master(biến master là biến chứa lệnh người dùng), không thành công sẽ trả về chuỗi rỗng.

1. **Hàm giúp Trợ lý ảo nói:**

def speak(robot\_brain):

print("Alice: " + robot\_brain)

tts = gTTS(text=robot\_brain, lang='vi', slow=False)

tts.save("sound.mp3")

text\_area.insert(INSERT,"ALice: "+robot\_brain+"\n")

playsound.playsound("sound.mp3")

os.remove("sound.mp3")

robot\_speak.runAndWait()

window.update()

* Hàm giúp robot nói những gì nó suy nghĩ (robot\_brain).
* **Ý tưởng:** Chuyển văn bản thành file âm thanh sound.mp3 bằng thư viện gTTS. Mỗi khi nói sẽ chạy file sound.mp3, sau khi chạy xong sẽ xóa luôn file sound.mp3 khỏi hệ thống.
* **Thực hiện:** Nhận văn bản từ biến robot\_brain. Gán file âm thanh sound.mp3 được chuyển từ robot\_brain thành âm thanh tiếng việt bằng thư viện gTTS với biết tts. Lưu file sound.mp3 vào hệ thống. Sau khi chạy file sound.mp3, sound.mp3 sẽ được xóa khỏi hệ thống thông qua thư viện os. Tạo quãng nghỉ sau khi nói.

1. **Hàm tính thời gian:**

def time\_message():

robot\_brain = "Bạn muốn xem thời gian cụ thể nào?"

speak(robot\_brain)

master = get\_audio()

now = datetime.now()

time0 = now.strftime("%w")

time1 = int(time0)

time2 = "Chủ nhật"

if "giờ" in master:

robot\_brain = "Bây giờ là %d giờ %d phút %d giây" % (now.hour, now.minute, now.second)

elif "ngày" in master:

robot\_brain = "Hôm nay là ngày %d tháng %d năm %d" % (now.day, now.month, now.year)

elif "thứ" in master and time1!=0:

robot\_brain = "Hôm nay là thứ %s" % (time1+1)

elif "thứ" in master and time1==0:

robot\_brain = "Hôm nay là thứ %s" % (time2)

elif "hiện tại" in master and time1!=0:

robot\_brain = "Hôm nay là thứ %s ngày %d tháng %d năm %d và bây giờ là %d giờ %d phút %d giây" % (time1+1,now.day, now.month, now.year,now.hour, now.minute, now.second)

elif "hiện tại" in master and time1==0:

robot\_brain = "Hôm nay là thứ %s ngày %d tháng %d năm %d và bây giờ là %d giờ %d phút %d giây" % (time2,now.day, now.month, now.year,now.hour, now.minute, now.second)

else:

speak("Tôi chưa hiểu ý của bạn. Bạn nói lại được không?")

time.sleep(6)

speak(robot\_brain)

* Hàm giúp trợ lý ảo xem thời gian hiện tại
* **Ý tưởng:** Lấy thời gian hiện tại: thứ, ngày tháng năm, giờ phút giây thông qua thư viện time và datetime. Sau đó, gán biến robot\_brain với kết quả lấy được để có được suy nghĩ của trợ lý ảo
* **Thực hiện:** Hỏi người dùng muốn xem thời gian như thế nào bằng robot\_brain. Sau khi hỏi sẽ lấy lệnh người dùng thông qua hàm get\_audio(). Biến now là đối tượng datetime chứa ngày và giờ hiện tại. Hàm strftime(“%w”) có thể chứa code định dạng ngày trong tuần, dạng giá trị số (từ 0 đến 6). Biến time1 ép kiểu dạng giá trị số thành kiểu int. Truyền vào biến time2 một chuỗi tên là Chủ nhật. Nếu lệnh lấy được có “giờ” thì truyền vào chuỗi trong robot\_brain giờ, phút, giây hiện tại. Nếu có “ngày” trong lệnh thì truyền vào chuỗi là ngày tháng năm. Nếu có “hiện tại” trong lệnh mà định dạng ngày trong tuần khác 0 thì +1 vào time1 để làm thứ trong tuần và xuất ra thời gian ngày tháng năm, giờ phút giây, định dạng ngày bằng 0 thì thứ sẽ là biến time2(“Chủ nhật”). Nếu trong lệnh không có các chuỗi trên thì xuất ra chuỗi không hiểu ý.

1. **Hàm chào hỏi:**

def Hello():

hour = datetime.now().hour

if hour >= 1 and hour < 12:

robot\_brain = "Chào buổi sáng tốt lành, master"

elif hour >= 12 and hour < 18:

robot\_brain = "Chào buổi chiều tốt lành, master"

elif hour >= 18 and hour < 22:

robot\_brain = "Chào buổi tối tốt lành, master"

else:

robot\_brain = "Buổi đêm vui vẻ, master"

speak(robot\_brain)

robot\_brain = "Alice đã sẵn sàng, master?"

speak(robot\_brain)

window.update()

* Trợ lý ảo sẽ luôn chào bạn mỗi khi chương trình được khởi chạy
* **Ý tưởng:** Dựa theo thời gian hiện tại lấy được mà xuất ra các câu chào hỏi dành cho người dùng khi khởi động hệ thống.
* **Thực hiện:** Biến hour là đối tượng datetime chứa giờ hiện tại. Dựa điều kiện là từng khoảng giờ đã như trên mà gán biến robot\_brain với từng chuỗi chào phù hợp với giờ hiện tại.

1. **Hàm tìm kiếm google:**

def findgoogle():

robot\_brain = "Ngài muốn tìm gì trên google, master?"

speak(robot\_brain)

search = get\_audio()

url = f"http://www.google.com/search?q={search}"

wb.get().open(url)

robot\_brain = f"Đây là kết quả của tìm kiếm {search} trên google, thưa ngài!!"

speak(robot\_brain)

* Hàm giúp mở web và tìm kiếm thông tin yêu cầu trên google
* **Ý tưởng:** Tìm kiếm trên google theo từ khóa có được từ lệnh của dùng
* **Thực hiện:** Hỏi người dùng muốn tìm gì trên google. Lấy từ khóa qua hàm get\_audio(). Thêm từ khóa vào cấu trúc url tìm kiếm của google. Thực hiện mở trang web tìm kiếm trên google theo url đã có bằng thư viện webbrowser

1. **Hàm tìm kiếm trên youtube và mở video:**

def findyoutube():

robot\_brain = "Ngài muốn tìm gì trên youtube, master?"

speak(robot\_brain)

search = get\_audio()

url = f"http://www.youtube.com/search?q={search}"

wb.get().open(url)

robot\_brain = f"Đây là kết quả của tìm kiếm {search} trên Youtube, thưa ngài!!"

speak(robot\_brain)

speak("Ngài muốn mở video luôn không?")

answer = get\_audio()

if "có" in answer:

while True:

result = YoutubeSearch(search, max\_results=10).to\_dict()

if result:

break

speak("Ngài muốn mở video số mấy vậy?chỉ nói số thôi nha")

number = get\_audio()

for i in range(len(result)):

i = w2n(number)

url = f"http://www.youtube.com" + result[i]['url\_suffix']

wb.get().open(url)

speak("Mời ngài thưởng thức video")

elif "không" in answer:

speak("Ngài còn yêu cầu gì nữa, master?")

pass

* Tìm kiếm theo yêu cầu trên youtube và mở video theo số được yêu cầu từ người sử dụng (giới hạn 10 video đầu)
* **Ý tưởng:** Tìm kiếm trên youtube cũng tương tự như tìm kiếm trên google nhưng sử dụng cấu trúc url của youtube. Lấy danh sách nội dung các video đang tìm kiếm trên youtube(tối đa là 10). Phát video đã tìm kiếm trên youtube theo yêu cầu của người dùng.
* **Thực hiện:** Tìm kiếm trên youtube giống như tìm kiếm trên google nhưng lại sử dụng cấu trúc url của youtube. Sau khi tìm kiếm, sử dụng model YoutubeSearch để lấy nội dung của 10 video tìm kiếm trên youtube và trả về dưới dạng to\_dict(). Gán kết quả cho biết result. Kiểm tra result, nếu có dữ liệu thì break. Lấy số của video muốn mở từ người dùng. Vì trợ lý ảo sử dụng tiếng việt nên số lấy được từ lệnh là chữ số nên phải dùng model w2n để chuyển chữ số thành kiểu int. Thiết lập đường dẫn của youtube cộng thêm kết quả result có giá trị là số của video được yêu cầu mở và tìm kiếm đến key trong dictionary [‘url\_suffix’]. Phát video youtube theo đường dẫn đã có.

1. **Hàm xem thời tiết:**

def current\_weather():

speak("Bạn muốn xem thời tiết ở đâu ạ.")

# Đường dẫn trang web để lấy dữ liệu về thời tiết

ow\_url = "https://api.openweathermap.org/data/2.5/weather?"

# lưu tên thành phố vào biến city

city = get\_audio()

# nếu biến city != 0 và = False thì để đấy ko xử lí gì cả

if not city:

pass

# api\_key lấy trên open weather map

api\_key = "3eb48f07c0a964bc11da07e05083f94b"

# tìm kiếm thông tin thời thời tiết của thành phố

call\_url = ow\_url + "q=" + city + "&appid=" + api\_key +"&units=metric"

# truy cập đường dẫn trên để lấy dữ liệu thời tiết

response = requests.get(call\_url)

# lưu dữ liệu thời tiết dưới dạng json và cho vào biến data

data = response.json()

print(data)

# kiểm tra nếu ko gặp lỗi 404 thì xem xét và lấy dữ liệu

if data["cod"] != "404":

# lấy value của key main

city\_res = data["main"]

# nhiệt độ hiện tại

current\_temperature = city\_res["temp"]

# áp suất hiện tại

current\_pressure = city\_res["pressure"]

# độ ẩm hiện tại

current\_humidity = city\_res["humidity"]

# thời gian mặt trời

suntime = data["sys"]

# lúc mặt trời mọc, mặt trời mọc

sunrise = datetime.fromtimestamp(suntime["sunrise"])

# lúc mặt trời lặn

sunset = datetime.fromtimestamp(suntime["sunset"])

# thông tin thêm

wthr = data["weather"]

# mô tả thời tiết

weather\_description = wthr[0]["description"]

# Lấy thời gian hệ thống cho vào biến now

now = datetime.now()

# hiển thị thông tin với người dùng

content = f"""

Hôm nay là ngày {now.day} tháng {now.month} năm {now.year}

Mặt trời mọc vào {sunrise.hour} giờ {sunrise.minute} phút

Mặt trời lặn vào {sunset.hour} giờ {sunset.minute} phút

Nhiệt độ trung bình là {current\_temperature} độ C

Áp suất không khí là {current\_pressure} héc tơ Pascal

Độ ẩm là {current\_humidity}%

"""

speak(content)

else:

# nếu tên thành phố không đúng

speak("Không tìm thấy địa chỉ của bạn")

* Hàm giúp lấy dữ liệu thời tiết hiện tại từ trang web openweathermap
* **Ý tưởng:** Lấy dữ liệu thời tiết từ web openweathermap.org và xuất ra dữ liệu thời tiết mong muốn.
* **Thực hiện:** Đăng ký tài khoản trên web openwweathermap.org. Sau khi đăng ký, lấy api\_key được cung cấp miễn phí cho mỗi tài khoản. Thiết lập đường dẫn trang web và gán vào biến ow\_url. Lấy tên thành phố từ hàm get\_audio(). Thiết lập đường dẫn tìm kiếm dữ liệu theo cấu trúc của trang web openwweathermap được kết hợp từ biến ow\_url, api\_key và tên thành phố muốn lấy. Truy cập đường dẫn và lấy dữ liệu qua thư viện request. Lưu dữ liệu dưới dạng json vào biến data. Nếu không gặp lỗi thì lấy dữ liệu thời tiết thành phố muốn biết ở hiện tại và xuất ra nội dung tương ứng với thời gian hiện tại

1. **Hàm đổi background máy tính:**

def change\_wallpaper():

speak("Đang thay đổi!!! Ngài vui lòng đợi trong giây lát")

api\_key = "hLcHOGGlLwDNQGimKpkZMAd5HCEqGJXgKtnfnO4v3dk"

url = 'https://api.unsplash.com/photos/random?client\_id=' + \

api\_key # pic from unspalsh.com

f = urllib2.urlopen(url)

json\_string = f.read()

f.close()

parsed\_json = json.loads(json\_string)

photo = parsed\_json['urls']['full']

# Location where we download the image to.

urllib2.urlretrieve(photo, "img\\a.png")

image = os.path.join("C:\\Users\\Acer\\OneDrive\\Máy tính\\AI\\img\\a.png")

ctypes.windll.user32.SystemParametersInfoW(20, 0, image, 3)

speak("Đã đổi thành công. Bạn ra home xem có đẹp không nha ?")

* Lấy ngẫu nhiên 1 hình ảnh từ web unsplash.com và lưu về file img với tên là a.png sau đó tiến hành đổi background cho máy tính
* **Ý tưởng:** Lấy ngẫu nhiên một hình ảnh trên web unsplash (web cung cấp API ảnh miễn phí). Lưu file ảnh về hệ thống với tên a.png. Đổi hình nền máy tính bằng ảnh a.png vừa mới lưu về
* **Thực hiện:** Đăng ký tài khoản trên web unsplash.com. Lấy api\_key được cung cấp miễn phí. Thiết lập đường dẫn url chọn ngẫu nhiên ảnh trên web cùng với api\_key của tài khoản. Sử dụng thư viện urllib2 để duyệt web ngầm từ url đã thiết lập. Tạo biến json\_string lấy kết quả trả về của trang web bằng cách đọc trang web. Sau khi đã có kết quả thì đóng chạy trình duyệt ngầm. Xử lý dữ liệu của biến json\_string bằng thư viện json. Lấy ảnh thông qua dữ liệu vừa xử lý. Lưu ảnh vào đường dẫn cho trước bằng thư viện urllib2. Lấy đường dẫn tuyệt đối của ảnh muốn thay đổi ảnh nền qua thư viện os. Sử dụng thư viện ctypes để đổi ảnh nền trên máy tính.

1. **Hàm phát nhạc có sẵn:**

def play\_music():

speak("Đây là danh sách nhạc trong thư mục của ngài")

# path là tham số chứa đường dẫn thư mục chứa nhạc

path = "music"

myPATH = path

# lấy file nhạc ra

ds = os.listdir(myPATH)

print(ds)

#in danh sách bài hát

for a in ds:

print(str(ds.index(a))+ ":" + a)

robot\_brain= str(ds.index(a))+ ":" + a

speak(robot\_brain)

speak("Ngài muốn chọn bài hay phát hết??")

master = get\_audio()

if "chọn bài" in master:

# mở bài nhạc đã chọn

speak("Ngài hãy chọn bài!! Chỉ nói số thôi nha ")

while True:

number = get\_audio()

i= int(w2n(number))

speak("Mời Ngài thưởng thức")

print("\nĐang phát bài : " + ds[i])

os.system(myPATH + "\\" + ds[i])

print("\nĐã phát xong bài : \t\t" + ds[i])

speak("Đã hết")

speak("bạn có muốn nghe nữa không??")

master = get\_audio()

if "không" in master:

break

elif "phát hết" in master:

# phát hết nhạc trong mục

for i in range(len(ds)):

print("\nĐang phát bài : " + ds[i])

os.system(myPATH + "\\" + ds[i])

print("\nĐã phát xong bài : \t\t" + ds[i])

speak("Đã hết")

* Các video âm nhạc có sẵn được lưu trong thư mục music. Hàm giúp lấy và phát video trong máy theo yêu cầu
* **Ý tưởng:** Lấy danh sách các file nhạc từ đường dẫn cho trước. Tiến hành phát nhạc theo yêu cầu người dùng. Nếu người dùng yêu cầu chọn bài để phát thì phát bài theo yêu cầu của họ. Nếu người dùng yêu cầu phát hết thì phát hết nhạc trong danh sách.
* **Thực hiện:** Lấy danh sách file nhạc từ đường dẫn cho trước. Duyệt và xuất ra các file nhạc có trong danh sách. Lấy yêu cầu người dùng. Nếu người dùng yêu cầu chọn bài. Tiếp tục lấy số của bài muốn phát qua model w2n. phát bài theo kết quả lấy được. Nếu người dùng yêu cầu phát hết thì phát hết nhạc trong danh sách.

1. **Hàm tìm kiếm và đọc wiki**

def knowledge():

try:

speak("Thưa Ngài, Ngài muốn tìm gì ạ ?")

text = get\_audio()

contents = wikipedia.summary(text).split('\n')

speak(contents[0])

dem = 0

for content in contents[1 : ]:

if dem < 2:

speak("Bạn có muốn biết thêm không ???")

ans = get\_audio()

if 'có' not in ans:

break

dem += 1

speak(content)

speak("Đã hết cảm ơn Ngài đã lắng nghe")

except:

speak(f"Alice không định nghĩa được thuật ngữ của Ngài !!!")

* Tìm kiếm và mở wikipedia, đọc nội dung sau khi tìm kiếm trên wiki
* **Ý tưởng:** Lấy nội dung từ kết quả tìm kiếm theo từ khóa trong yêu cầu của người dùng. Đọc nội dung đoạn đầu tiên nếu muốn biết thêm nội dung thì đọc nội dung đoạn thứ 2 rồi kết thúc hàm.
* **Thực hiện:** Tìm kiếm trên wiki theo lệnh người dùng và lấy nội dung đoạn đầu tiên qua thư viện wikipedia. Nếu người dùng muốn biết thêm thông tin thì đọc nội dung thứ 2 rồi kết thúc hàm. Xử lý ngoại lệ, có lỗi xảy ra thì thông báo trợ lý ảo không định nghĩa được thuật ngữ muốn tìm kiếm.

1. **Hàm Liệt kê các chức năng:**

def func1():

content="""

Alice có những chức năng sau đây:

1.Chào hỏi

2.Thông báo thời gian

3.Dự báo thời tiết

4.Tìm kiếm trên google

5.Tìm kiếm và mở video trên youtube

6.Thay đổi hình nền máy tính

7.Mở nhạc trong thư mục có sẵn

8.Đọc thông tin trên wikipedia

9.Tạm biệt"""

speak(content)

* Liệt kê các chức năng hiện có trong trợ lý ảo
* **Ý tưởng:** Liệt kê các chức năng có trong chương trình trợ lý ảo
* **Thực hiện:** Gán chuỗi có các chức năng của trọ lý ảo vào biến content. Khi chạy hàm, trợ lý ảo nói nội ding trong biết content

1. **Hàm main để chạy các chức năng:**

def main():

print("AI Alice Starting......")

Hello()

while True:

master = get\_audio()

if "thời gian" in master:

time\_message()

elif master == "":

speak("Ngài hãy ra lệnh đi!!")

elif "google" in master:

findgoogle()

elif "youtube" in master:

findyoutube()

elif "wiki" in master:

knowledge()

elif "mở nhạc" in master:

play\_music()

elif "đổi hình nền" in master:

change\_wallpaper()

elif "thời tiết" in master:

current\_weather()

elif "chức năng" in master:

func1()

elif "tạm biệt" in master:

robot\_brain = "tạm biệt ngài!! Chúc ngài may mắn !!!"

speak(robot\_brain)

break

else:

speak("Tôi chưa hiểu ý của ngài, Ngài lặp lại được không?")

* Hàm khởi chạy các chức năng theo yêu cầu của người sử dụng
* **Ý tưởng:** Chạy các hàm chức năng theo yêu cầu người dùng
* **Thực hiện:** Khi chạy hàm main, trợ lý ảo đầu tiên sẽ chạy hàm chào hỏi (Hello()). Tạo vòng lặp while, bên trong chạy các chức năng theo lệnh của người dùng lấy được từ hàm get\_audio(). Vòng lặp kết thúc khi lệnh của người dùng là tạm biệt. Nếu lệnh của người dùng không có trong điều kiện để chạy hàm thì trợ lý ảo yêu cầu người dùng lặp lại

1. **Giao diện**
2. **Khai báo và khởi tạo giao diện**

window = Tk()

window.geometry("1000x600")

window.title("AI ALICE")

window.configure(bg = "#333131")

* Khai báo và tạo giao diện bằng tkinter
* Giao diện có kích thước, tiêu đề và màu nền

1. **Tạo Canvas:**

canvas = Canvas(

window,

bg = "#333131",

height = 600,

width = 1000,

bd = 0,

highlightthickness = 0,

relief = "ridge")

canvas.place(x = 0, y = 0)

canvas1 = Canvas(

window,

bg = "#F8EEEE",

bd = 0,

height = 541,

width = 473,

highlightthickness = 0,

relief = "ridge"

)

canvas1.place(x = 0, y = 0)

* Tạo 2 canvas chia giao diện thành 2 phần với chức năng riêng biệt
* Mỗi canvas đều có độ dài, rộng , màu sắc và vị trí khác nhau

1. **Nội dung trong Canvas:**

img0 = PhotoImage(file = f"img0.png")

b0 = Button(

image = img0,

borderwidth = 0,

bg = "#333131",

highlightthickness = 0,

command = main,

activebackground= "#333131",

relief = "flat")

b0.place(

x = 667, y = 278,

width = 134,

height = 130,

)

Label = Label(

window,

text= "Trợ Lý ảo Alice",

bg = "#3C3939",

fg= "white",

font ="Inter, 40",

)

Label.place(

x= 559, y=31,

width = 376,

height =113

)

* Trong canvas có button(b0) và label
* Button (b0) chứa hình ảnh img0.png và label chứa tên trợ lý ảo
* Button (b0) chứa command khi click sẽ khởi chạy hàm main thực hiện các chức năng trợ lý ảo theo yêu cầu người dùng.

1. **Nội dung trong Canvas1:**

text\_area = Text(

canvas1,

height= 541,

width= 473

)

text\_area.place(x = 0, y = 0)

entry0\_img = PhotoImage(file = f"img\_textBox0.png")

entry0\_bg = canvas.create\_image(

207.5, 570.5,

image = entry0\_img)

entry0 = Entry(

bd = 0,

bg = "#fbf8f8",

highlightthickness = 0,

)

entry0.place(

x = 36.5, y = 550,

width = 342.0,

height = 39)

img1 = PhotoImage(file = f"img1.png")

b1 = Button(

image = img1,

borderwidth = 0,

bg = "#333131",

highlightthickness = 0,

activebackground= "#333131",

relief = "flat")

b1.place(

x = 408, y = 550,

width = 51,

height = 41)

* Canvas1 chứa Text (text\_area), button(b1) và entry(entry0)
* text\_area là một textbox lớn chứa nội dung đối thoại của người sử dụng và trợ lý ảo
* entry0 và b1 thì giúp người sử dụng nhập lệnh và in ra màn hình khi click. entry chứa img\_textbox0.png và b1 chứa img1.png

1. **Lệnh chạy giao diện tkinter:**

window.resizable(False, False)

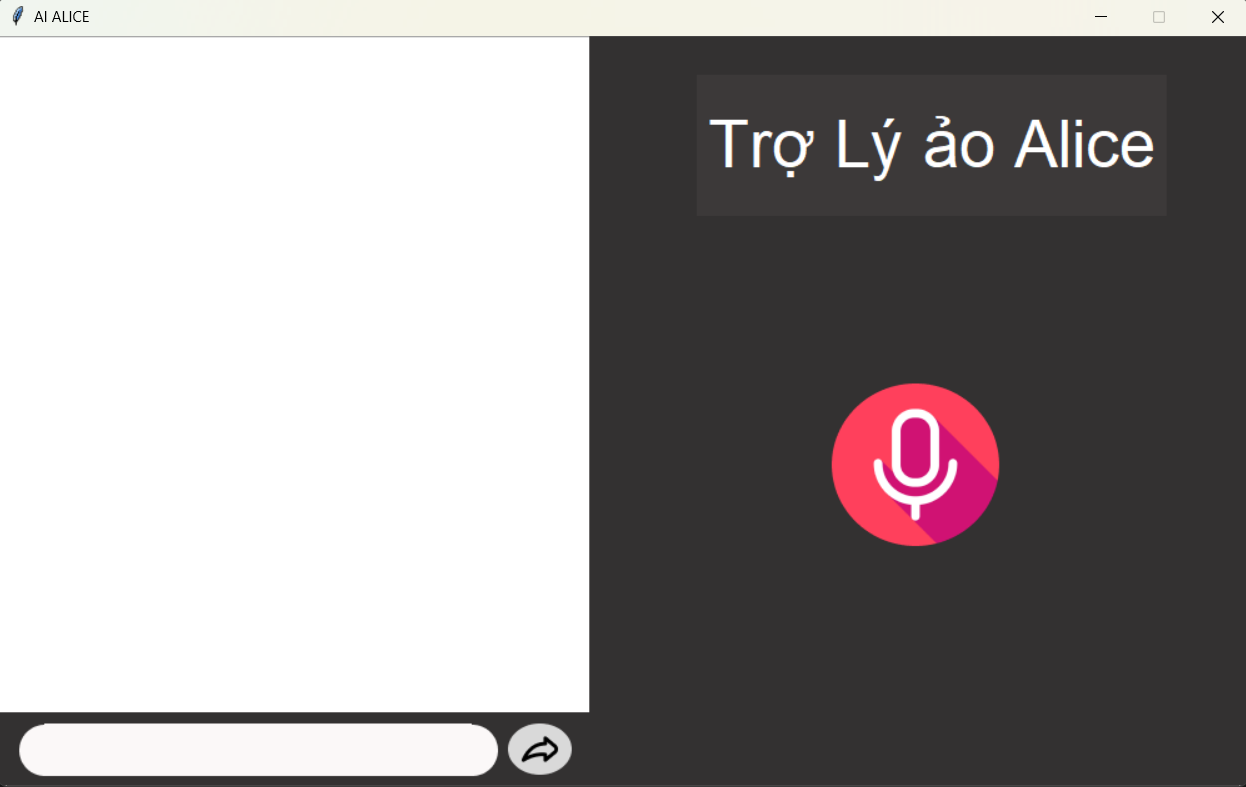
window.mainloop()

- Chạy giao diện với nội dung trên

# 

# [CHƯƠNG 4: GIAO DIỆN VÀ DEMO](https://docs.google.com/document/d/1pBMq_OnytAk68APTzbhrs0Yx5Kck2ZCH/edit" \l "heading=h.43ky6rz)

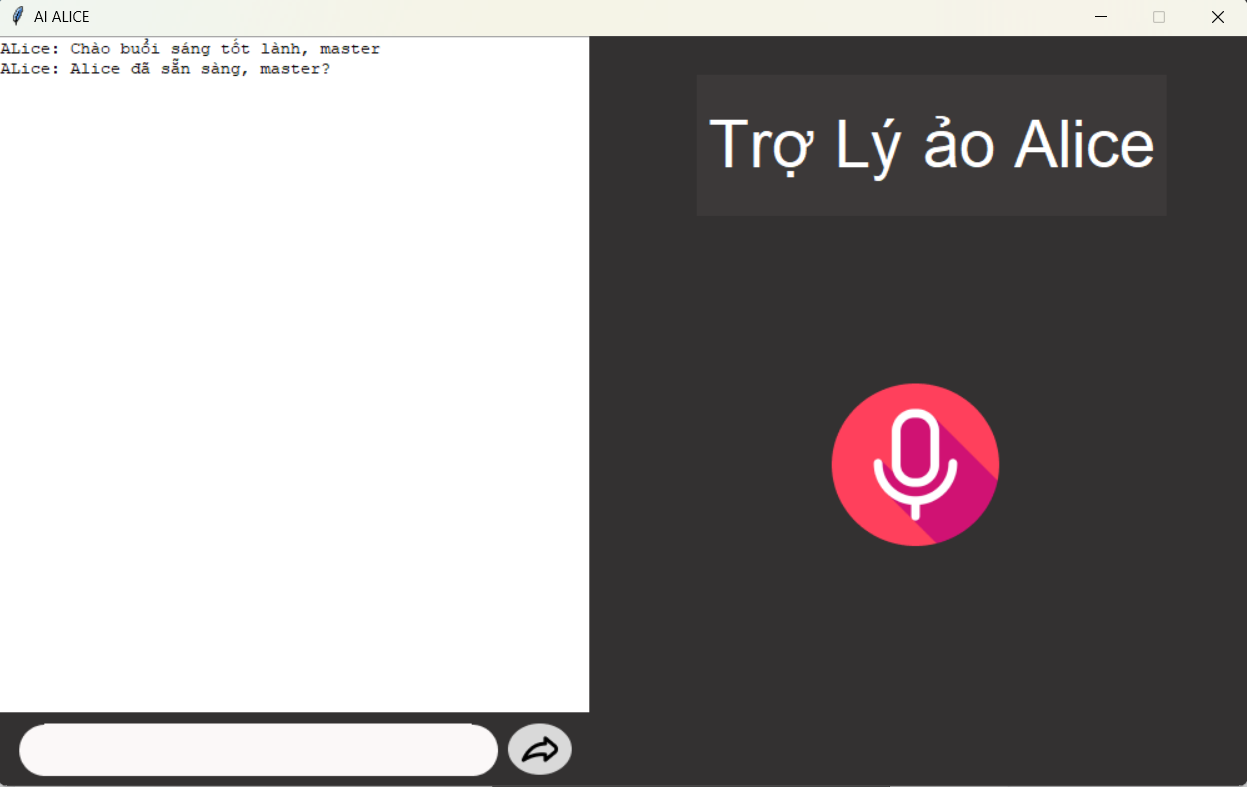
1. **Giao diện chính:**



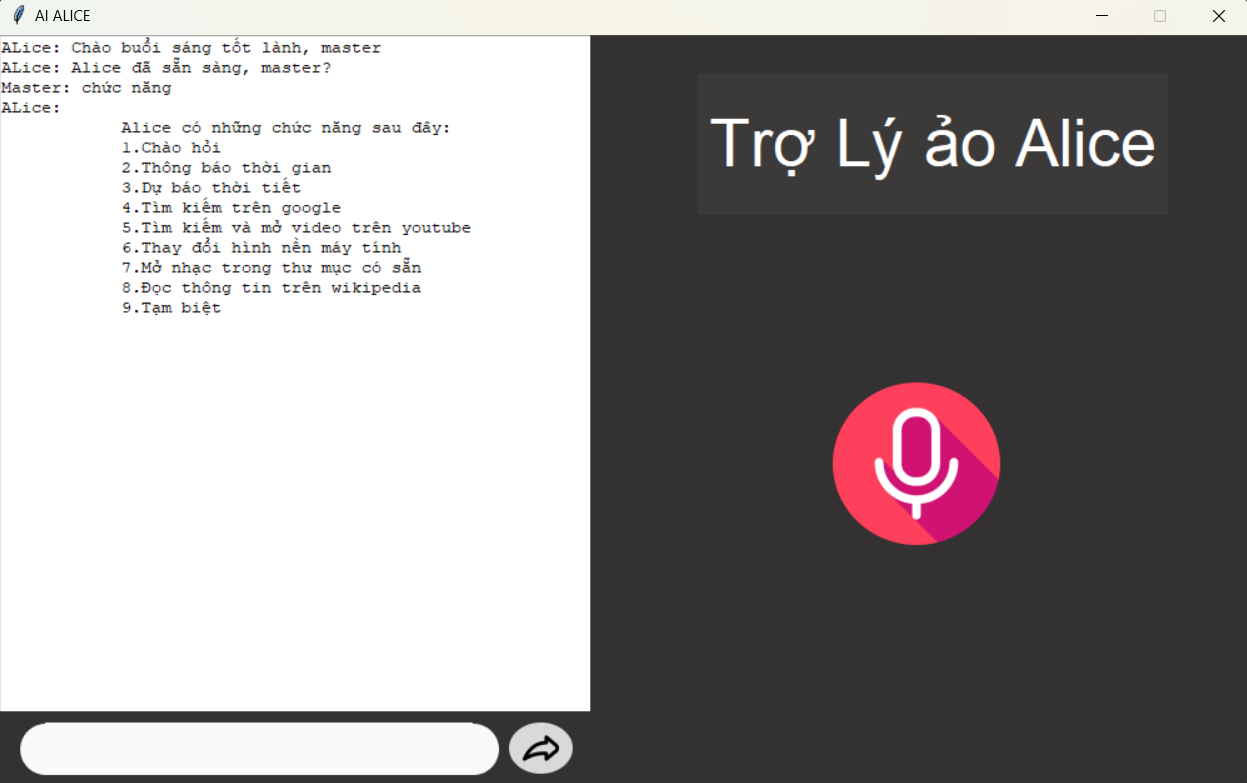
*Hình 2: Giao diện chính*

* Khi chạy source code giao diện với tkinker sẽ xuất hiện giao diện như hình 2.
* Khi nhấn nút có icon micro thì trợ lý ảo sẽ chào hỏi thực hiện các chức năng theo yêu cầu người dùng
* Khi nhập lệnh vào textbox nhỏ bên dưới cùng phía bên trái và nhấn nút gửi thì trên textbox lớn phía trên sẽ được in ra lệnh của người dùng
* Textbox lớn sẽ in ra cuộc đối thoại và chức năng thực hiện của trợ lý ảo và người dùng

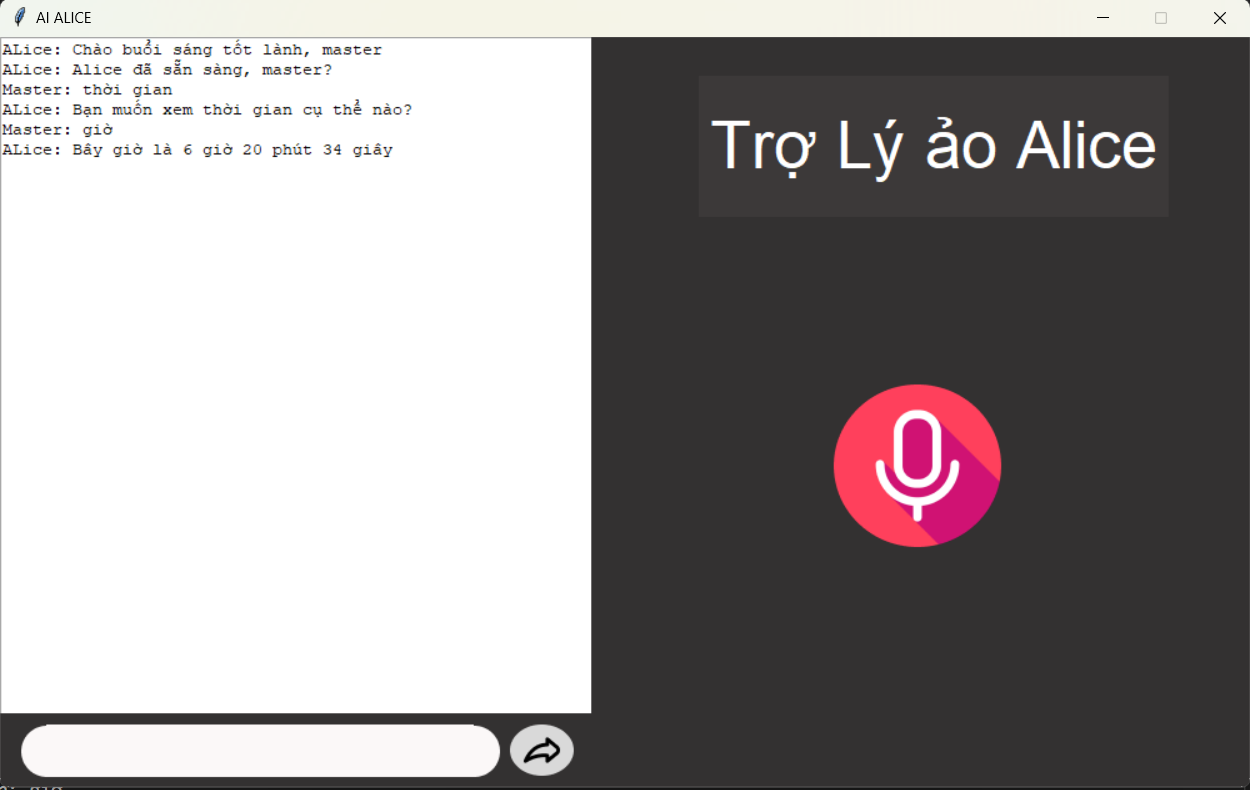
1. **Demo chương trình theo từng chức năng:**
2. **Chức năng chào hỏi:**

****

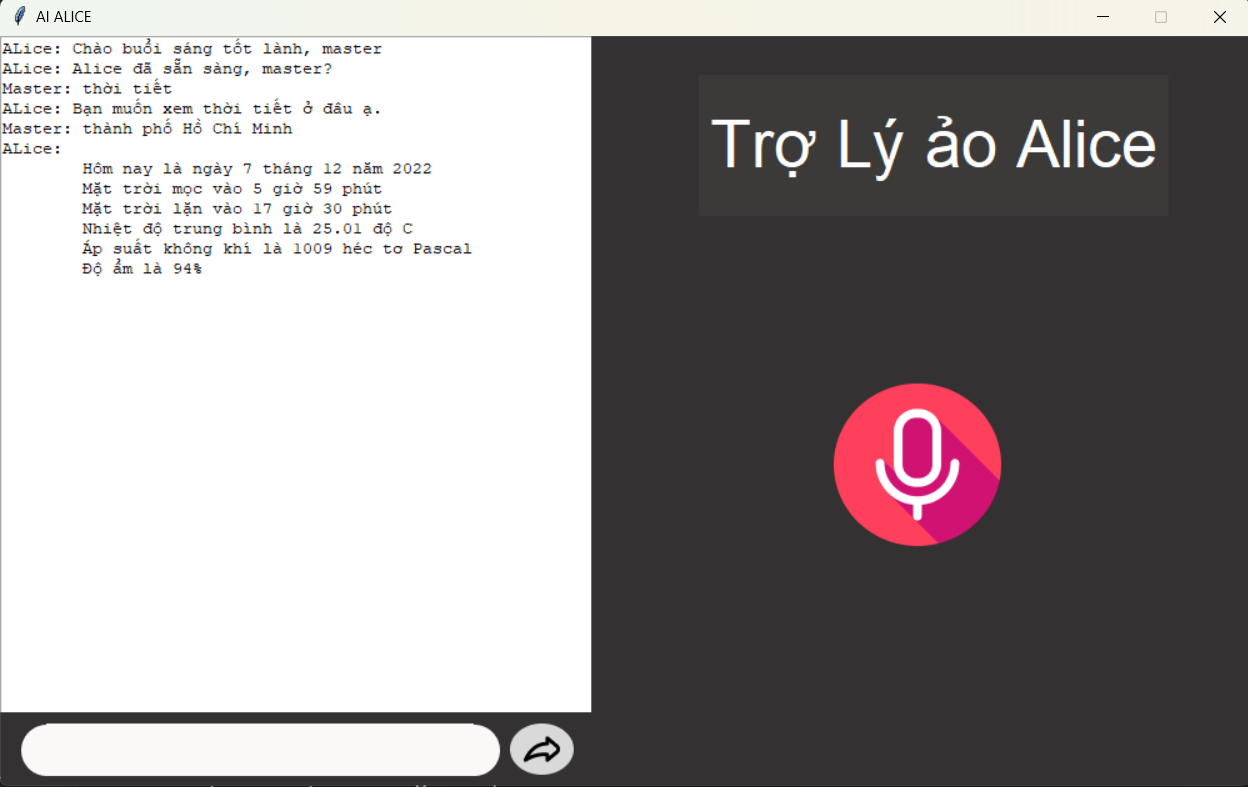
1. **Liệt kê các chức năng:**

****

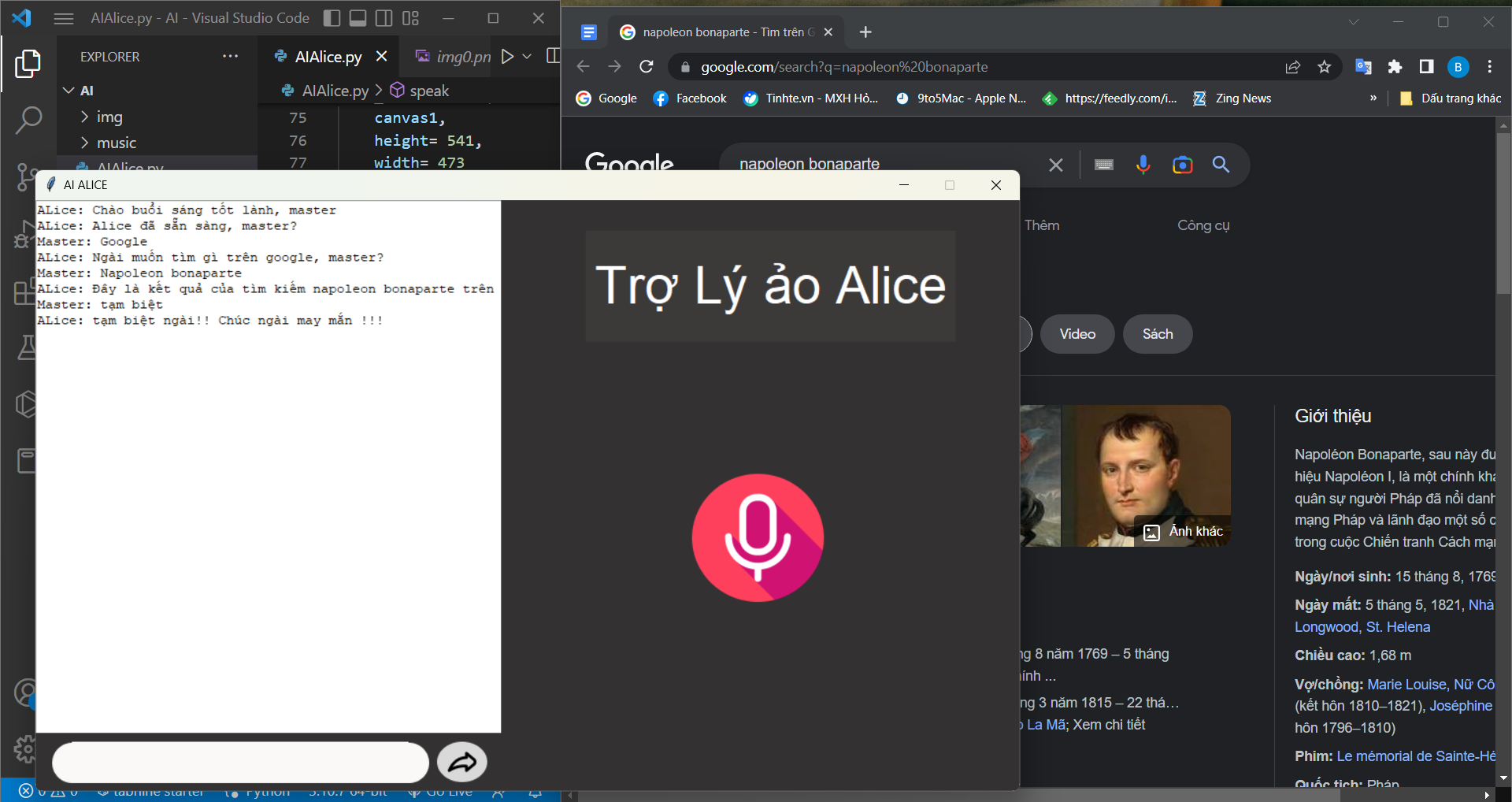
1. **Thông báo thời gian**

****

1. **Dự báo thời tiết:**

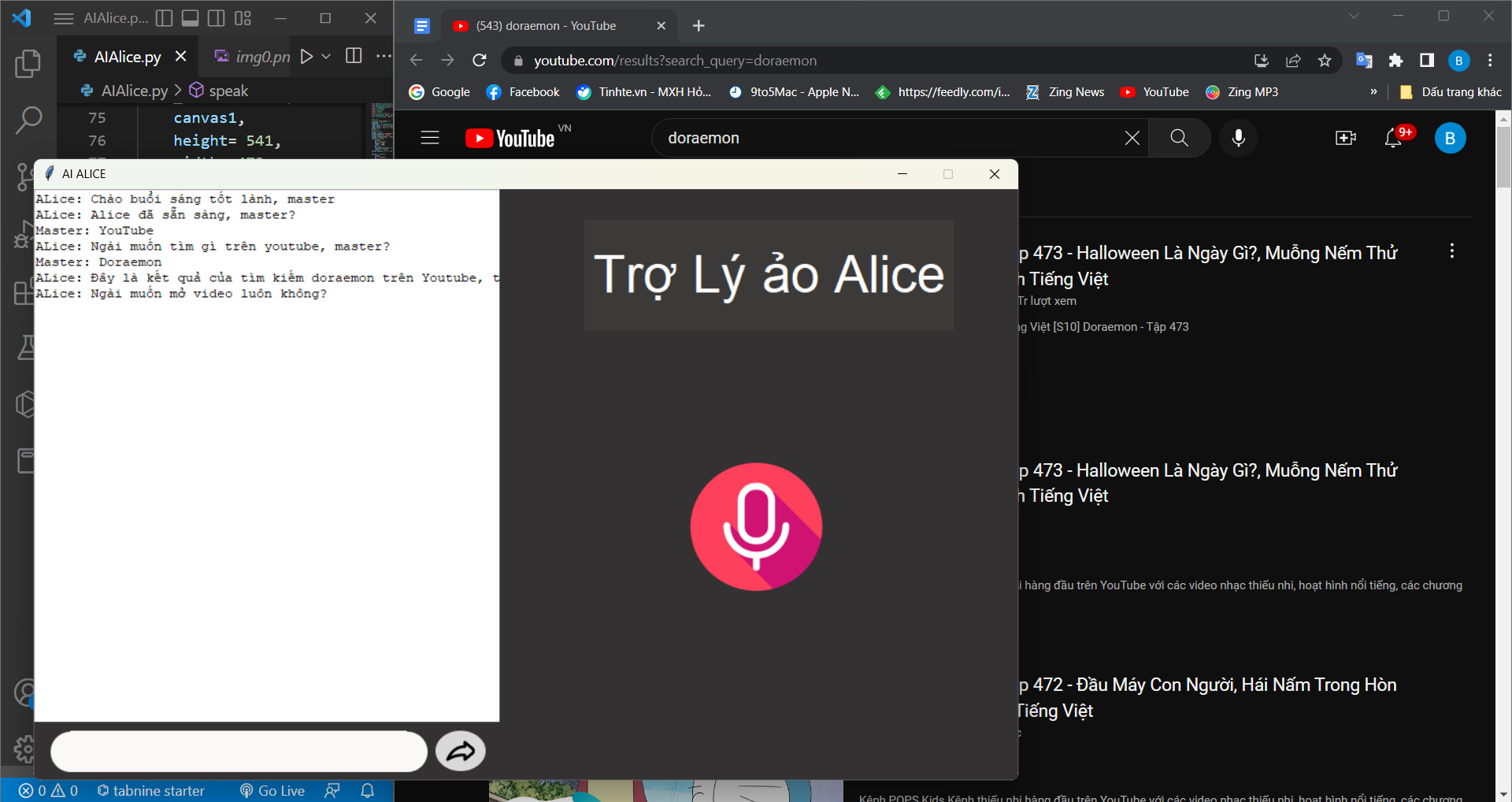
****

1. **Tìm kiếm trên Google**

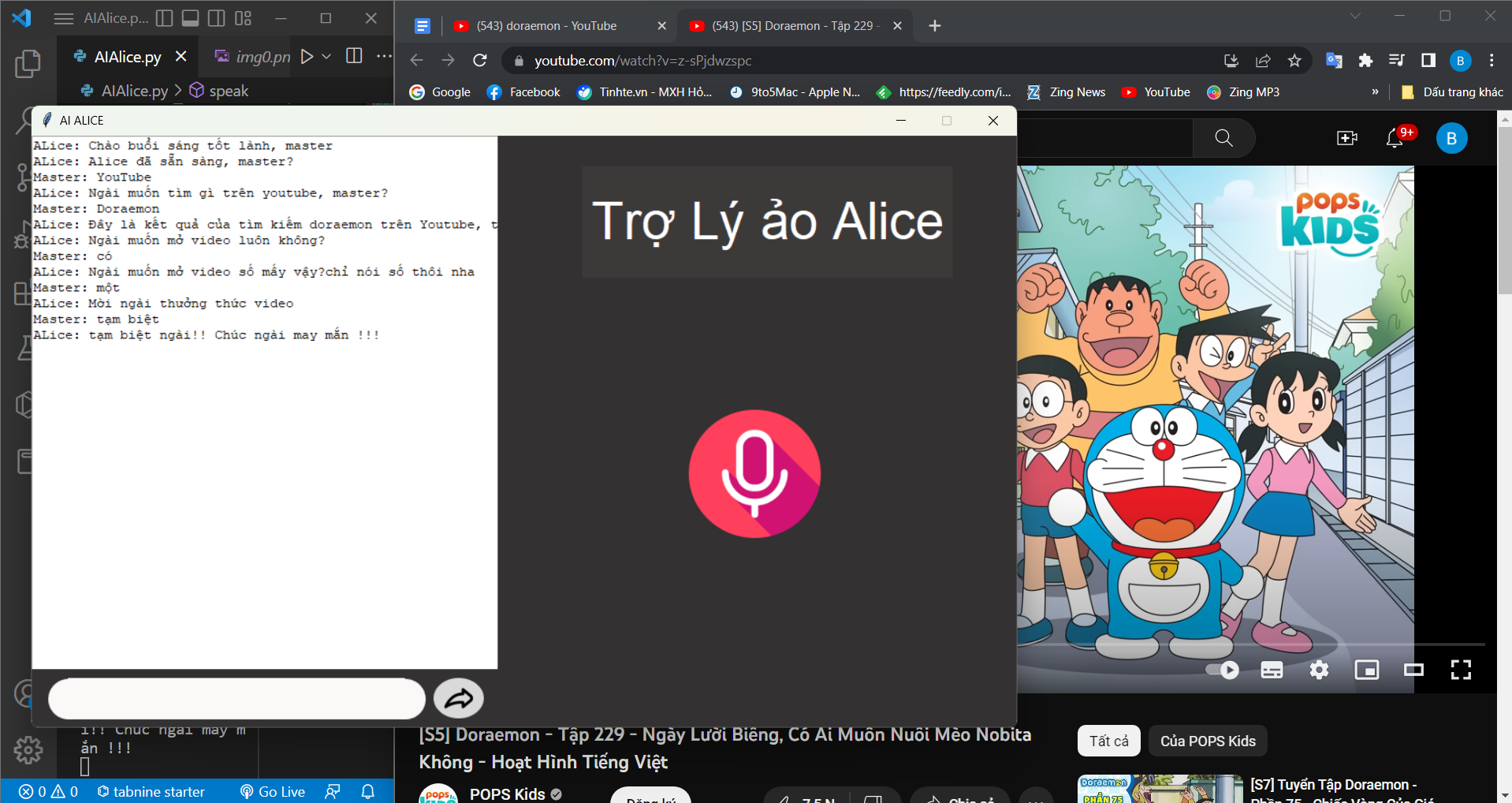
****

1. **Tìm kiếm và mở video trên youtube**

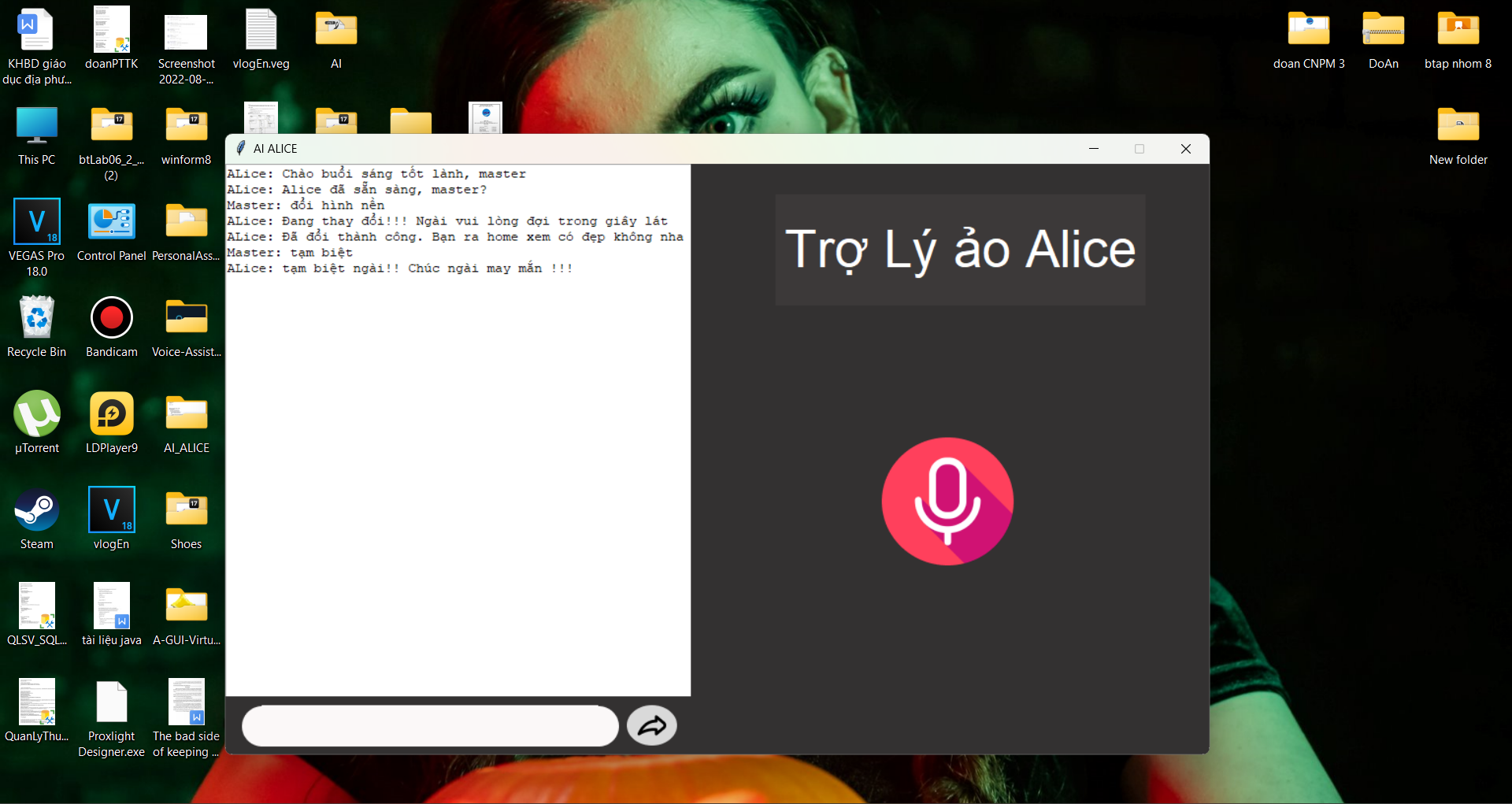
* **Tìm kiếm trên youtube**

****

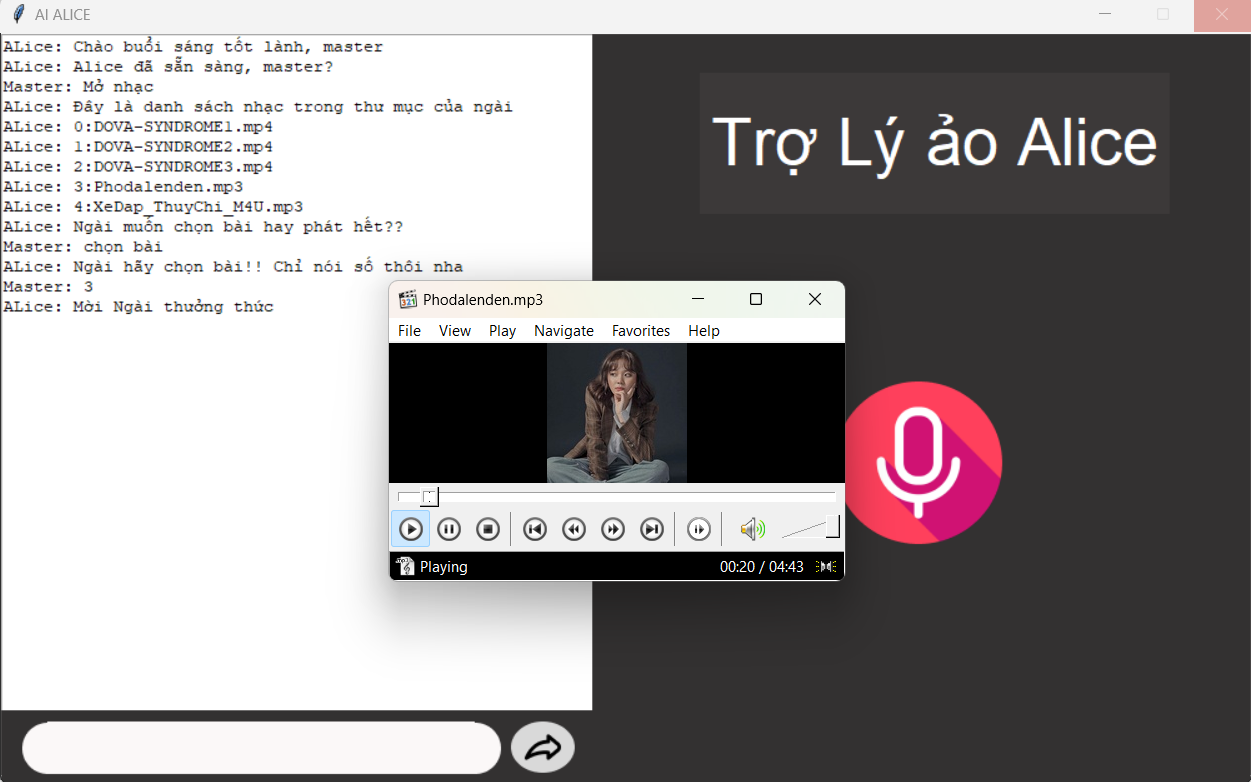
* **Mở video trên youtube**

****

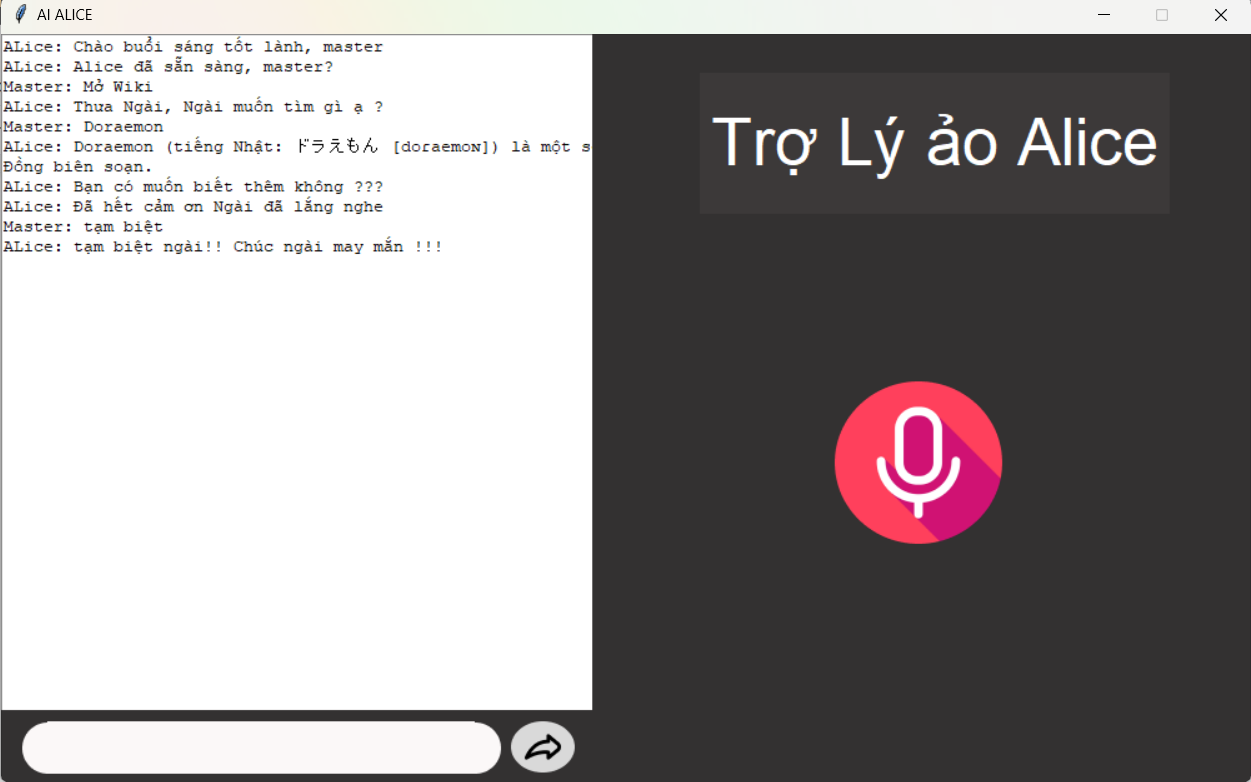
1. **Thay đổi hình nền**

****

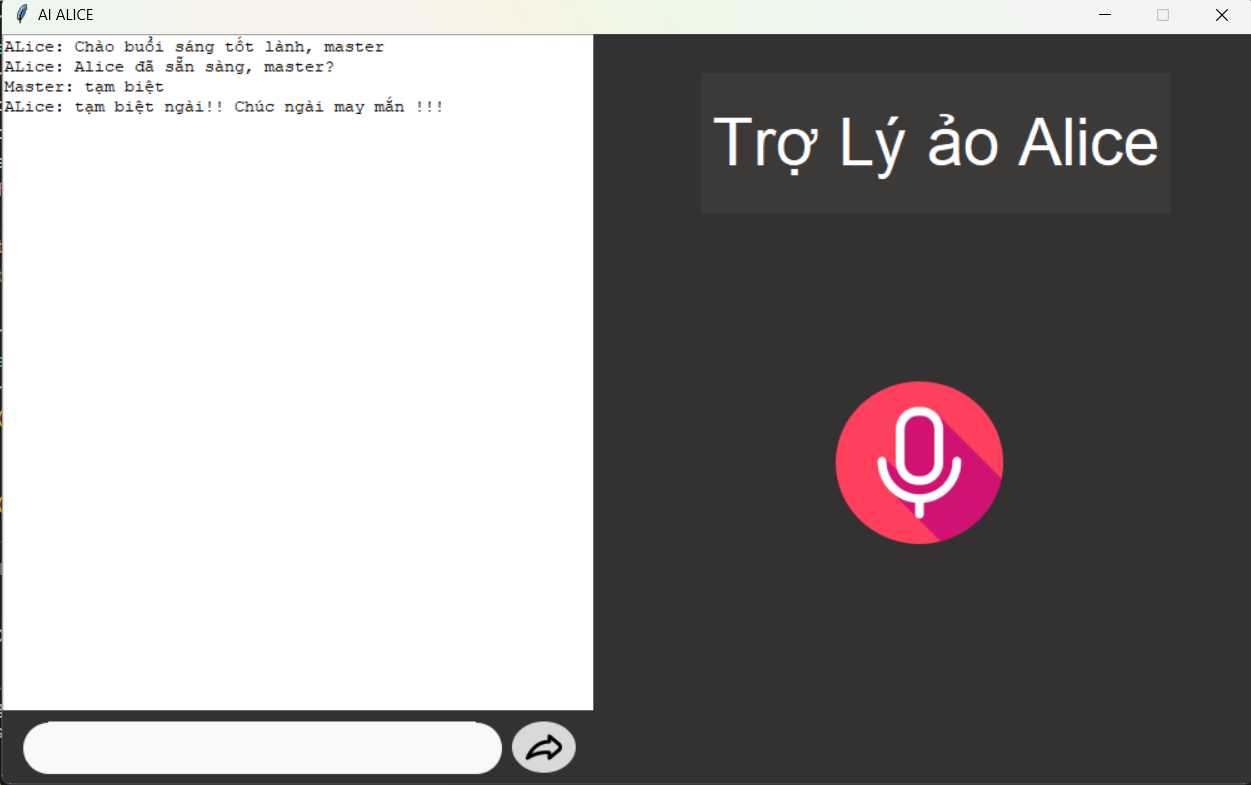
1. **Mở nhạc trong thư mục có sẵn**

****

1. **Đọc thông tin trên wikipedia**

****

1. **Tạm biệt**

****

# [CHƯƠNG 5:KẾT LUẬN](https://docs.google.com/document/d/1pBMq_OnytAk68APTzbhrs0Yx5Kck2ZCH/edit" \l "heading=h.39kk8xu)

## I. Những điều đã đạt được

Trợ lý ảo Alice đã có hầu hết các chức năng cơ bản mà 1 trợ lý ảo nên có tuy nhiên vẫn còn một số hạn chế chưa khắc phục được như chưa có nút tắt khi đang đối thoại, chưa xử lý phần nhập lệnh với textbox dưới, giao diện đơn điệu chưa thực sự đẹp mắt và cần có nhiều chức năng hơn, ……

## II. Hướng phát triển cho tương lai

Trợ lý ảo Alice vẫn có tiềm năng phát triển trong tương lai. Nếu có thêm nhiều chức năng và tối ưu được các mặt hạn chế thì trợ lý ảo Alice sẽ trở nên hoàn hảo và hoàn thiện hơn. Trong tương lai, nếu có thể trợ lý ảo có thể thực hiện công nghệ IOT, đưa công nghệ lên một thời đại mới.

# 

# [TÀI LIỆU THAM KHẢO](https://docs.google.com/document/d/1pBMq_OnytAk68APTzbhrs0Yx5Kck2ZCH/edit" \l "heading=h.2nusc19)

<https://www.youtube.com/watch?v=Nx_kwUQKqV8>

<https://www.youtube.com/watch?v=wVboOz_O8rE>

Link Source code: <https://github.com/buikien2k2/AIALICE>