## horizontal line



Alvin: The Youtube Chatbot

09.04.2022

**─**

Raghav Sharma: 200953042

Shraddha Somani: 200953049

Harshvardhan Pandey: 200953048

# 

# Table of Contents

# Abstract

# Introduction

# Background Information

# Methodology

# Implementation

# Results

# 

# Abstract

**Alvin: The youtube chatbot** is a python3 based chatbot integrated with “Graphical User Interface (GUI)” application. As the name suggests, this chatbot aims to revolutionize the youtube-surfing experience for the user.

# Introduction

**Alvin: The youtube chatbot** provides a unique user-experience for youtube surfing.

The chatbot is housed in the backend of a user-friendly “Tkinter” based Graphical User Interface.

The following is a list of functions Alvin serves:

1. Generic GUI-chatbot experience
2. Youtube streaming and surfing
3. Playlist creation and visualization
4. Custom made playlist streaming

# Background Information

**Alvin: The youtube chatbot** is developed on the following foundations:

1. The *tkinter* Python interface
2. The *urllib* package
3. The *re* module
4. The *pywhatkit* library

*Tkinter* is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit. This will serve as the GUI for **Alvin: The youtube chatbot.**

The *urllib* package is the main package in python to work with URLs and it includes several modules. This project uses the *urllib.request module* to get the HTML for the search results page on Youtube.

The *re* module or the *regular expressions* module specifies a set of strings that matches it. The functions in this module let you check if a particular string matches a given regular expression (or if a given regular expression matches a particular string, which comes down to the same thing). **Alvin: The youtube chatbot** uses the *re* module to find occurrences that include the 11-characters identifier in the HTML of the youtube search.

The *pywhatkit* library is used to perform google searches and play youtube videos on the GUI.

# Methodology

With the help of online resources like Google, Stack Overflow, and other coding platforms, we were able to find and understand code that would help us in achieving our goal for the product.

The GUI’s editing was done with the aid of YouTube tutorials as well as the manual’s provided by the university.

There were many obstacles encountered, for example, figuring out which module to use to open and save the URLs from YouTube. This was overcome by extensive research as well as hours spent understanding different modules and libraries that Python provides, in hopes of finding the one that will suit this project the most.

# Implementation

**Alvin: The youtube chatbot** presents the user with its GUI, where the user can view multiple sub-modules of the chatbot as previously mentioned.

The user is required to select the desired submodule and await further instructions as clearly depicted on the GUI.

The user is then asked to enter the required inputs and finally executes the desired task.

A user can:

Play as song after mentioning the name of the artist/the name of the song to the chatbot

Talk to the chatbot freely by pressing Talk To Me which enables a casual conversation

Add to playlist by mentioning the names of the songs/artists

Save a playlist to local drive

Play a playlist from local drive

Show a playlist that the user might have just created

# Results

Code:

from doctest import master

from tkinter import \*

import urllib.request

import re

import pywhatkit

root = Tk()

root.config(bg="red")

root.title("Alvinnnnnnn!!!")

BG\_GRAY = "#ABB2B9"

BG\_COLOR = "#17202A"

TEXT\_COLOR = "#EAECEE"

FONT = "Helvetica 14"

FONT\_BOLD = "Helvetica 13 bold"

playlist = []

namep=[]

songdetails = []

def add\_to():

play = "You -> Added " + e.get()

txt.insert(END, "\n" + play+" song")

user = e.get().lower()

namep.append(user)

e.delete(0, END)

def showplaylist():

txt.insert(END, "\n" + "Alvin -> Your playlist is: ")

for item in namep:

txt.insert(END, "\n" +item)

e.delete(0, END)

def play():

play = "You -> " + e.get()

txt.insert(END, "\n" + play)

user = e.get().lower()

txt.insert(END, "\n" + "Alvin -> Playing "+user)

artist = user+" song"

artist = artist.replace(" ", "+")

t = "https://www.youtube.com/results?search\_query=" + artist

html = urllib.request.urlopen(t)

video\_ids = re.findall(r"watch\?v=(\S{11})", html.read().decode())

z = "https://www.youtube.com/watch?v=" + video\_ids[0]

pywhatkit.playonyt(z)

e.delete(0, END)

def saveplaylist():

user = e.get().lower()

name = user + ".txt"

f = open(name, "w") #creating file

#write into file

f = open(name, "a")

for item in namep:

f.write(item + "\n")

f.close()

e.delete(0, END)

txt.insert(END, "\n" + "Alvin -> Playlist saved as "+name+" !!!!!")

def playplaylist():

user = e.get().lower()

name = user + ".txt"

f = open(name, "r") # creating file

for i in f.readlines():

i = i + "song"

artist = i.replace("\n", "+")

artist = artist.replace(" ", "+")

t = "https://www.youtube.com/results?search\_query=" + artist

html = urllib.request.urlopen(t)

video\_ids = re.findall(r"watch\?v=(\S{11})", html.read().decode())

z = "https://www.youtube.com/watch?v=" + video\_ids[0]

pywhatkit.playonyt(z)

e.delete(0, END)

txt.insert(END, "\n" + "Alvin -> Playeeeeieiiiiing playlist " +user+" !!!")

# Send function

def send():

send = "You -> " + e.get()

txt.insert(END, "\n" + send)

user = e.get().lower()

if (user == "hello"):

txt.insert(END, "\n" + "Alvin -> Hi there, how can I help?")

elif (user == "hi" or user == "hii" or user == "hiiii" or user=="hey"):

txt.insert(END, "\n" + "Alvin -> Hi there, what can I do for you?")

elif (user == "how are you"):

txt.insert(END, "\n" + "Alvin -> fine! and you")

elif (user == "fine" or user == "i am good" or user == "i am doing good"):

txt.insert(END, "\n" + "Alvin -> Great! how can I help you.")

elif (user == "thanks" or user == "thank you" or user == "now its my time"):

txt.insert(END, "\n" + "Alvin -> My pleasure !")

elif (user == "what do you sell" or user == "what kinds of items are there" or user == "have you something"):

txt.insert(END, "\n" + "Alvin -> We have coffee and tea")

elif (user == "tell me a joke" or user == "tell me something funny" or user == "crack a funny line"):

txt.insert(

END, "\n" + "Alvin -> What did the buffalo say when his son left for college? Bison.! ")

elif (user == "goodbye" or user == "see you later" or user == "see yaa"):

txt.insert(END, "\n" + "Alvin -> Have a nice day!")

elif (user == "which lab is your favourite?" or user == "which is your favourite lab?" or user == "which lab is your favourite"):

txt.insert(END, "\n" + "Alvin -> why do you even askkkk..Obviously Advanced Programming Lab")

else:

txt.insert(END, "\n" + "Alvin -> Sorry! I didn't understand that")

e.delete(0, END)

alvinicon = PhotoImage(file=r"your\_img.png")

lable1 = Label(root, image=alvinicon)

lable1.place(x=0,y=0,relwidth=1,relheight=1)

#fg=TEXT\_COLOR, text="Alvinnnnnnnnnn!!!!!", font=FONT\_BOLD, pady=10, width=20, height=1).grid(row=0)

txt = Text(root, bg="#A52A2A", fg=TEXT\_COLOR, font=FONT, width=60)

txt.grid(row=1, column=0, columnspan=2)

scrollbar = Scrollbar(txt)

scrollbar.place(relheight=1, relx=0.974)

e = Entry(root, bg="#2C3E50", fg=TEXT\_COLOR, font=FONT, width=55)

e.grid(row=2, column=0)

send = Button(root, text="Sup?", font=FONT\_BOLD, bg=BG\_GRAY,

command=send).grid(row=2, column=1)

play = Button(root, text="Play", font=FONT\_BOLD, bg="red",

command=play).grid(row=2, column=2)

add\_to = Button(root, text="Add to Playlist", font=FONT\_BOLD, bg="#EB5406",

command=add\_to).grid(row=2, column=3)

show = Button(root, text="Show Playlist", font=FONT\_BOLD, bg="#A52A2A",

command=showplaylist).grid(row=2, column=4)

save = Button(root, text="Save Playlist", font=FONT\_BOLD, bg="#800080",

command=saveplaylist).grid(row=2, column=5)

playplaylist = Button(root, text="Play Playlist", font=FONT\_BOLD, bg="#9966CB",

command=playplaylist).grid(row=2, column=6)

root.mainloop()

Graphical user interface

Description automatically generatedGraphical user interface

Description automatically generatedGraphical user interface

Description automatically generated

* Graphical user interface

  Description automatically generatedContents of Saved playlist on local drive as “demop”:

as it was

count on me