TIFFANY:
VOICE ASSISTANT

-GROUP 139

TEAM MEMBERS

KALMIT KULKARNI - 21BCE10724

KAUSTUBH TUNGAR - 21BCE10732

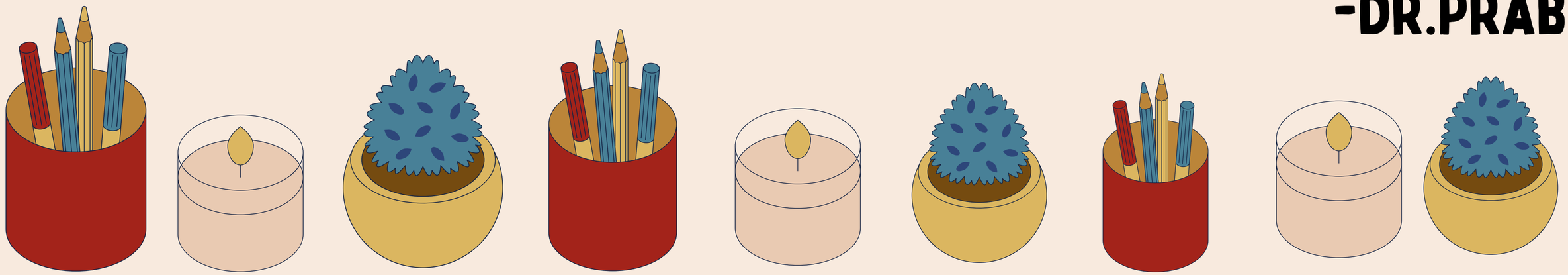
RISHI JAIN - 21BCE10745

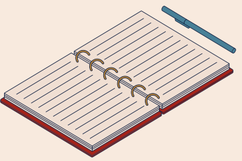
DEVANG SONI - 21BCE11522

YASH SADARANGANI - 21BCE11524

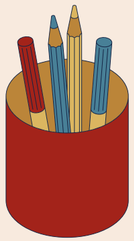
GUIDED BY

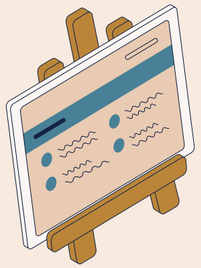
-DR.PRABU M





CONTENT





INTRODUCTION

OVER THE YEARS, HUMANS HAVE PROGRESSED IN INVENTING NEW TECHNOLOGIES FOR REDUCING HUMAN EFFORTS AND SAVING HUMAN LIFE. SINCE THE DEVELOPMENT OF IOT REDUCES HUMAN LABOUR TO NIL THE DEVELOPMENT OF IOT (INTERNET OF THINGS) HAS BEEN ADVANCED IN SEVERAL STUDY FIELDS LIKE HOME AUTOMATION, PERSONAL ASSISTANT AI, SMART CITY, SMART FARMING ETC. SO THE PERSONAL ASSISTANT HELPS REDUCE THE MANUAL EFFORTS BEING INPUT BY HUMANS IN THEIR DAY-TO-DAY TASKS. THE VOICE-CONTROLLED PERSONAL ASSISTANT RECEIVES THE VOICE COMMAND AS INPUT TO PERFORM NUMEROUS TASKS. ANY VOICE COMMAND SYSTEM NEEDS THREE ESSENTIAL COMPONENTS WHICH ARE A SPEECH TO TEXT CONVERTER, A QUERY PROCESSOR, AND A TEXT-TO-SPEECH CONVERTER. VOICE HAS BEEN A VERY INTEGRAL PART OF COMMUNICATION NOWADAYS.

THESE INNOVATIONS HAVE ATTRIBUTED TO THE TECHNOLOGY INDUSTRY USING DEEP LEARNING METHODS IN MAKING AND USING SOME OF THE SPEECH RECOGNITION SYSTEMS, GOOGLE WAS ABLE TO REDUCE WORD ERROR RATE BY 6 TO 10 RELATIVES, FOR THE SYSTEM THAT HAD THE WORD ERROR RATE OF 17 TO 52. TEXT TO SPEECH CONVERSION IS THE PROCESS OF CON- VERTING A MACHINE RECOGNIZED TEXT INTO ANY LANGUAGE WHICH COULD BE IDENTIFIED BY A SPEAKER WHEN THE TEXT IS READ OUT LOUD. IT IS A TWO-STEP PROCESS WHICH IS DIVIDED INTO FRONT END AND BACK END. THE FIRST PART IS RESPONSIBLE FOR CONVERTING NUMBERS AND ABBREVIATIONS TO A WRITTEN WORD FORMAT. THIS IS ALSO REFERRED TO AS NORMALIZATION OF TEXT.

EXISTING WORK

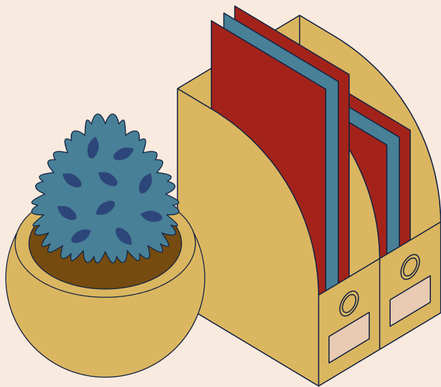
LOGIN PORTAL FOR FACE RECOGNITION - DEVANG SONI

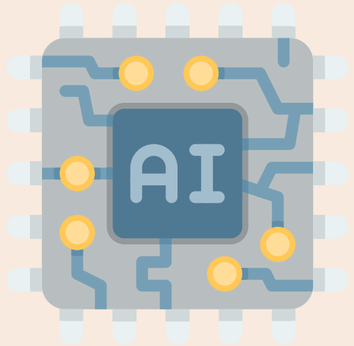
FACE RECOGNITION - KAUSTUBH TUNGAR

FACE RECOGNITION - YASH SADARANGANI

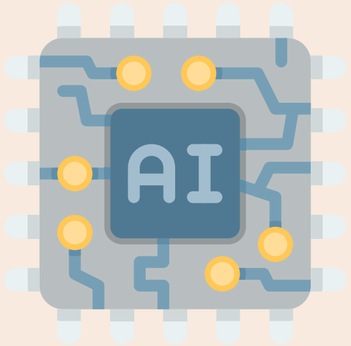
VOICE ASSISTANT - RISHI JAIN

GRAPHICAL USER INTERFACE - KALMIT KULKARNI

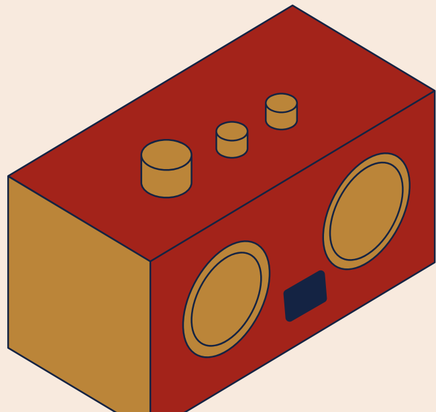




NOVELTY OF PROJECT



THE AIM OF THIS PROJECT IS TO DEMONSTRATE THE IMPLEMENTATION OF A VOICE COMMAND SYSTEM AS AN INTELLIGENT PERSONAL ASSISTANT (IPA) THAT CAN PERFORM VARIOUS TASKS OR SERVICES FOR AN INDIVIDUAL. IT IS BASED ON THE CONCEPTS OF PYTHON, SPEECH RECOGNITION, NATURAL LANGUAGE PROCESSING AND ARTIFICIAL INTELLIGENCE.



REAL TIME USAGE

- ADD EVENTS TO A CALENDER
- CONTROLS SMART HOME DEVICES
- MAKES AND RECEIVES PHONE CALL
- CREATES TEXT MESSAGES
- GIVES INFORMATION OF NEW AND WEATHER REPORT
- FIND HOTELS AND RESTAURANTS
- MUSIC PLAYBACK
- STREAMING PODCAST

HARDWARE & SOFTWARE REQUIREMENTS

HARDWARE

- INTEGRATED CAMERA

SOFTWARE

MODULES NEEDED:

- SUBPROCESS:- THIS MODULE IS USED TO GET SYSTEM SUBPROCESS DETAILS USED IN VARIOUS COMMANDS I.E SHUTDOWN, SLEEP, ETC. THIS MODULE COMES BUILT-IN WITH PYTHON.
- PYTTSX3:- THIS MODULE IS USED FOR CONVERTING TEXT TO SPEECH IN A PROGRAM THAT WORKS OFFLINE. TO INSTALL THIS MODULE TYPE THE BELOW COMMAND IN THE TERMINAL. PIP INSTALL PYTTSX3.
- TKINTER:- THIS MODULE IS USED FOR BUILDING GUI AND COMES INBUILT WITH PYTHON. THIS MODULE COMES BUILT-IN WITH PYTHON.

- WIKIPEDIA : AS WE ALL KNOW WIKIPEDIA IS A GREAT SOURCE OF KNOWLEDGE, WE HAVE USED THE WIKIPEDIA MODULE TO GET INFORMATION FROM WIKIPEDIA OR TO PERFORM A WIKIPEDIA SEARCH.

- SPEECH RECOGNITION : SINCE WE'RE BUILDING AN APPLICATION OF VOICE ASSISTANT, ONE OF THE MOST IMPORTANT THINGS IN THIS IS THAT YOUR ASSISTANT RECOGNIZES YOUR VOICE (MEANS WHAT YOU WANT TO SAY/ ASK).

- WEB BROWSER : TO PERFORM WEB SEARCH. THIS MODULE COMES BUILT-IN WITH PYTHON.

- FACE RECOGNITION : FACE RECOGNITION IS A TECHNOLOGY IN COMPUTER VISION. IN FACE RECOGNITION/DETECTION WE LOCATE AND VISUALIZE THE HUMAN FACES IN ANY DIGITAL IMAGE. IT IS A SUBDOMAIN OF OBJECT DETECTION, WHERE WE TRY TO OBSERVE THE INSTANCE OF SEMANTIC OBJECTS. THESE OBJECTS ARE OF PARTICULAR CLASS SUCH AS ANIMALS, CARS, HUMANS, ETC. FACE DETECTION TECHNOLOGY HAS IMPORTANCE IN MANY FIELDS LIKE MARKETING AND SECURITY.

- OPEN CV - PYTHON IS A LIBRARY OF PYTHON BINDINGS DESIGNED TO SOLVE COMPUTER VISION PROBLEMS. PYTHON IS A GENERAL PURPOSE PROGRAMMING LANGUAGE THAT BECAME VERY POPULAR VERY QUICKLY, MAINLY BECAUSE OF ITS SIMPLICITY AND CODE READABILITY

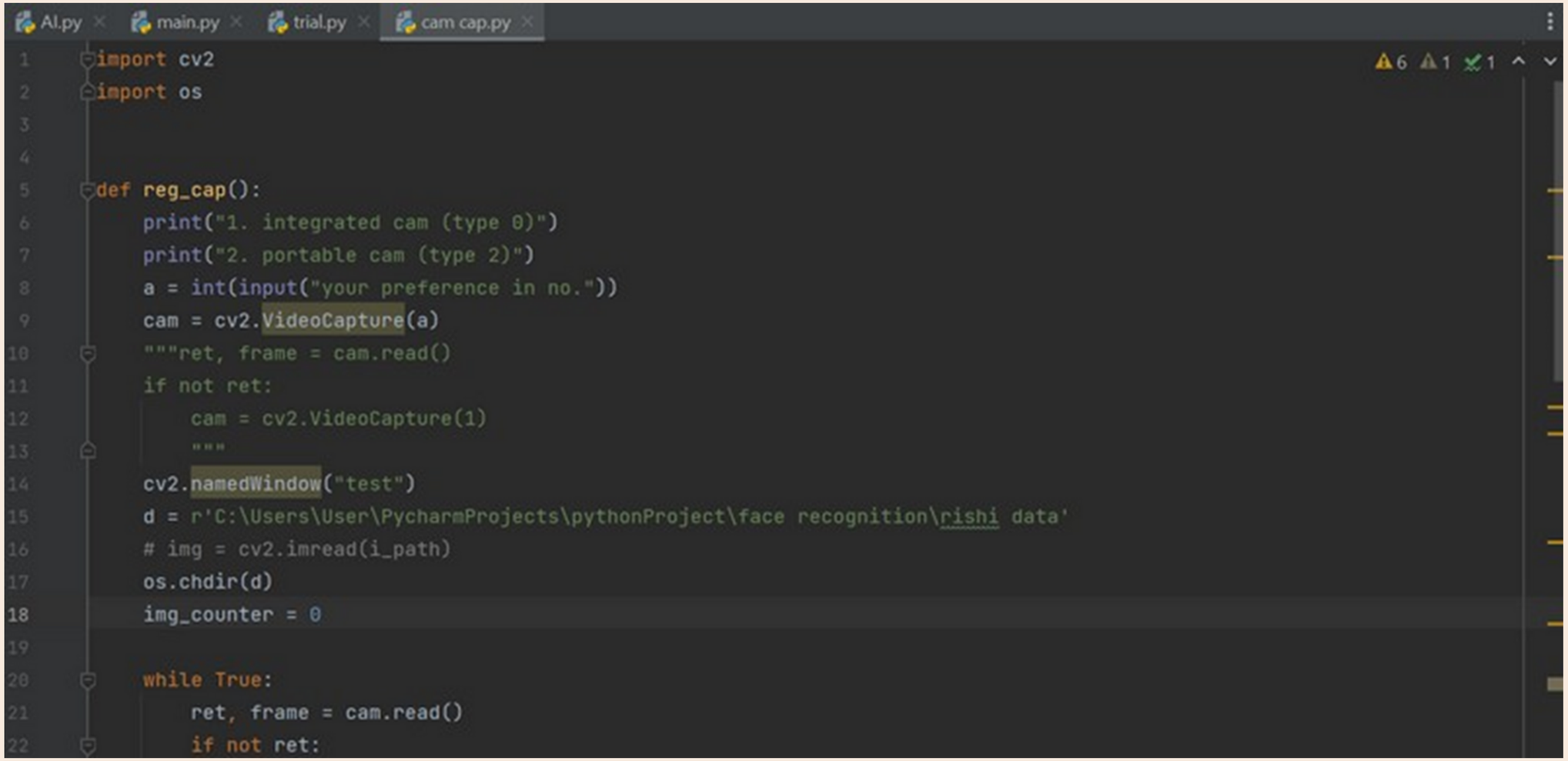
LITERATURE REVIEW

ALEXA, ECHO`S ENGINE, CAN WORK ON VARIOUS TABLETS AND DEVICES, ENCOURAGING NEW DEVELOPERS TO DESIGN APPS THAT WORK TOGETHER WITH IT. ON THE OTHER HAND, SIRI WORKS ONLY WITH IOS DEVICES. WHEN MICROSOFT LAUNCHED CORTANA, IT WAS PROMOTED AS TAKING THE TOP PROS FROM NOW AND SIRI, AND THAT IS PARTIALLY TRUE. CORTANA FALLS BEHIND IN ITS INTEGRATION WITH THE MOST IMPORTANT THIRD-PARTY APPS AVAILABLE ON THE MARKET. UNTIL MICROSOFT CORTANA IS AVAILABLE ON ANDROID AND IOS DEVICES, IT REMAINS THE ONLY GOOD OPTION FOR SYSTEMS WITH MICROSOFT WINDOWS 10 INSTALLED.

NOWADAYS SPEECH RECOGNITION CAN BE USED IN DIFFERENT AREAS. ITS MOST IMPORTANT APPLICATIONS ARE AIRCRAFT, SPEECH-TO-TEXT PROCESSING, SIMPLE DATA ENTRY, CALL ROUTING, SMART SEARCH, ETC. SPEECH RECOGNITION MOSTLY DEPENDS ON THE STATISTICAL MODELS. THESE MODELS TRANSFER SPEECH INTO TEXT FORM AND VICE VERSA. DIFFERENT TYPES OF STATIS- TICAL MODEL AVAILABLE AND USED IN THIS TYPE OF SYSTEM ARE ACOUSTIC MODEL, LANGUAGE MODEL, LEXICON MODEL, HIDDEN MARKOV MODEL. A VOICE CONTROLLED PERSONAL ASSISTANT IS CAPTURING PHOTOS AND RECOGNIZING FACES IN THE CAPTURED PHOTO, CHECKING SIMILARITY BETWEEN TWO FACES. IT ALSO PERFORMS ARITHMETIC CALCULATIONS BASED ON VOICE COMMANDS AND GIVES BACK THE COMPUTED SOLUTION THROUGH A ROBOTIC VOICE

MODULE DESCRIPTION

1). Capturing Image



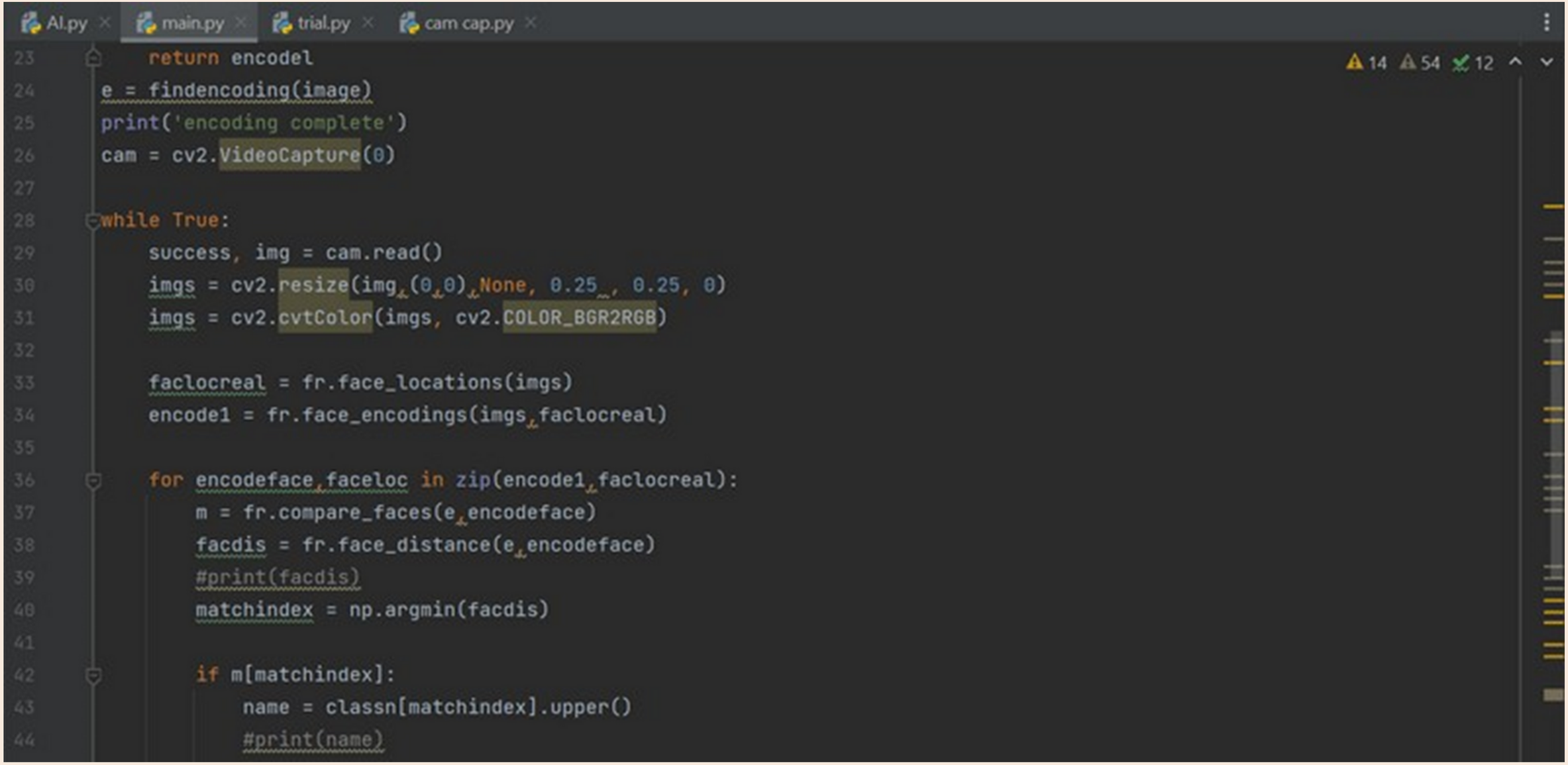
AI.py | main.py | trial.py | cam cap.py

```
        ret, frame = cam.read()
        if not ret:
            print("failed to grab frame")
            break
        cv2.imshow("test", frame)

        k = cv2.waitKey(1)
        if k % 256 == 27:
            # ESC pressed
            print("Escape hit, closing...")
            break
        elif k % 256 == 32:
            # SPACE pressed
            img_name = "pic_frame_{}.png".format(img_counter)
            cv2.imwrite(img_name, frame)
            print("{} written!".format(img_name))
            img_counter += 1

    cam.release()

    cv2.destroyAllWindows()
reg_cap()
```

Face Recognition

AI.py | main.py | trial.py | cam cap.py

```python
import cv2
import numpy as np
import face_recognition as fr
import os

path = 'rishi data'
image = []
classn = []
mlist = os.listdir(path)
print(mlist)
for cl in mlist:
    l = cv2.imread(f'{path}/{cl}')
    image.append(l)
    classn.append(os.path.splitext(cl)[0])
print(classn)

def findencoding(image):
    encodel=[]
    for img in image:
        img=cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
        encode = fr.face_encodings(img)[0]
        encodel.append(encode)
```


AI.py
main.py
trial.py
cam cap.py
14
54
12
return encodel
e = findencoding(image)
print('encoding complete')
cam = cv2.VideoCapture(0)
while True:
success, img = cam.read()
imgs = cv2.resize(img,(0,0),None, 0.25, 0.25, 0)
imgs = cv2.cvtColor(imgs, cv2.COLOR_BGR2RGB)
faclocreal = fr.face_locations(imgs)
encode1 = fr.face_encodings(imgs,faclocreal)
for encodeface,faceloc in zip(encode1,faclocreal):
m = fr.compare_faces(e,encodeface)
facdis = fr.face_distance(e,encodeface)
#print(facdis)
matchindex = np.argmin(facdis)
if m[matchindex]:
name = classn[matchindex].upper()
#print(name)


main.py
trial.py
cam cap.py
14
54
matchindex = np.argmin(facdis)
if m[matchindex]:
name = classn[matchindex].upper()
#print(name)
y1,x2,y2,x1 = faceloc
y1, x2, y2, x1 = y1*4,x2*4,y2*4,x1*4
cv2.rectangle(img,(x1,y1),(x2,y2),(0,0,255),2)
cv2.rectangle(img,(x1,y2-35),(x2,y2),(0,0,255),cv2.FILLED)
cv2.putText(img,name,(x1+6,y2-6),cv2.FONT_HERSHEY_COMPLEX,1,(255,255,255),2)
cv2.imshow('webcam',img)
cv2.waitKey(1)

Voice Assistant

AI.py | main.py | trial.py | cam cap.py

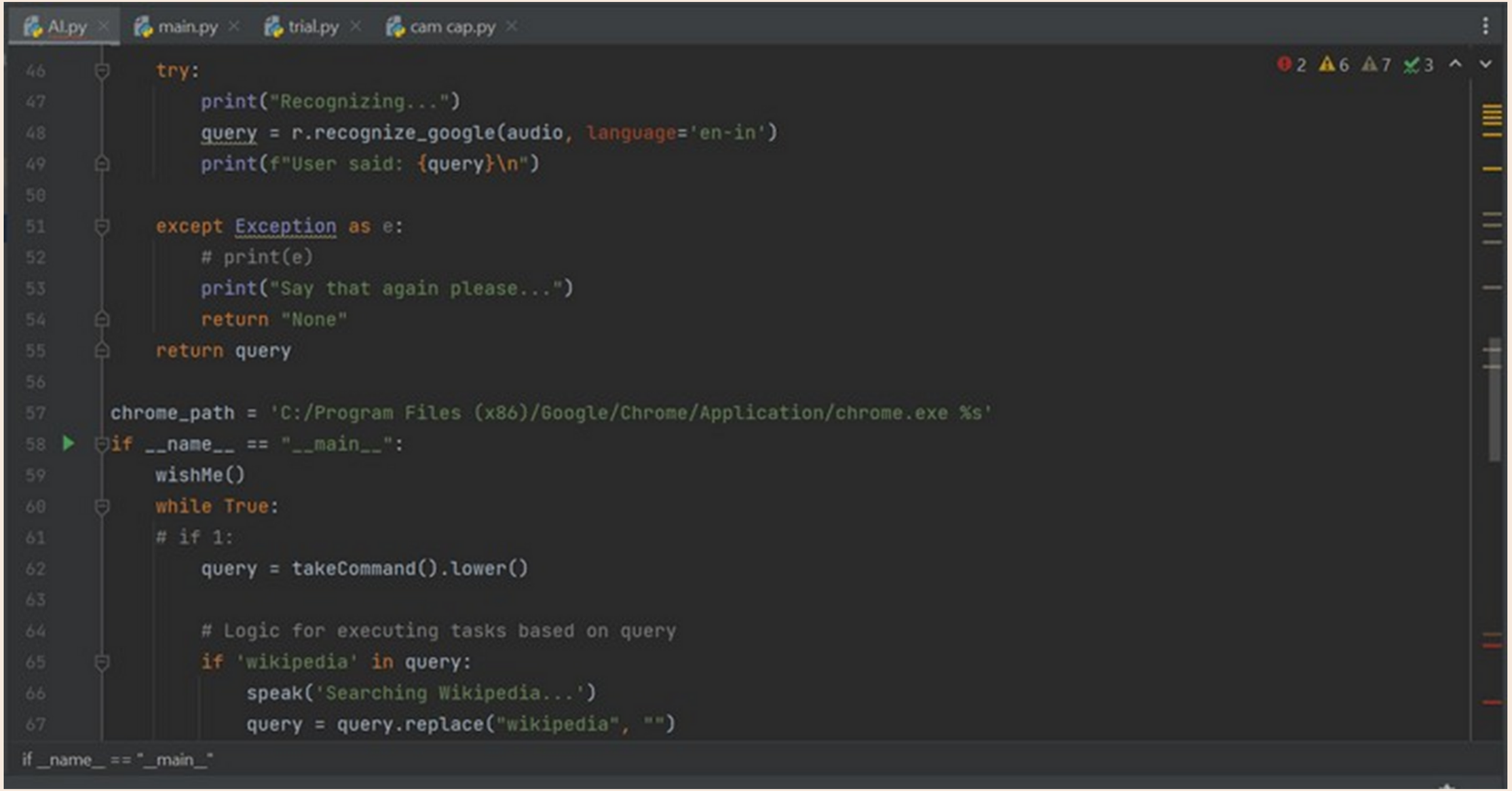
```python
import pyttsx3
import speech_recognition as sr
import datetime
import wikipedia
import os
import smtplib
import pyaudio
import google
import webbrowser
import subprocess
import pyjokes

engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
# print(voices[1].id)
engine.setProperty('voice', voices[1].id)


def speak(audio):
    engine.say(audio)
    engine.runAndWait()

```

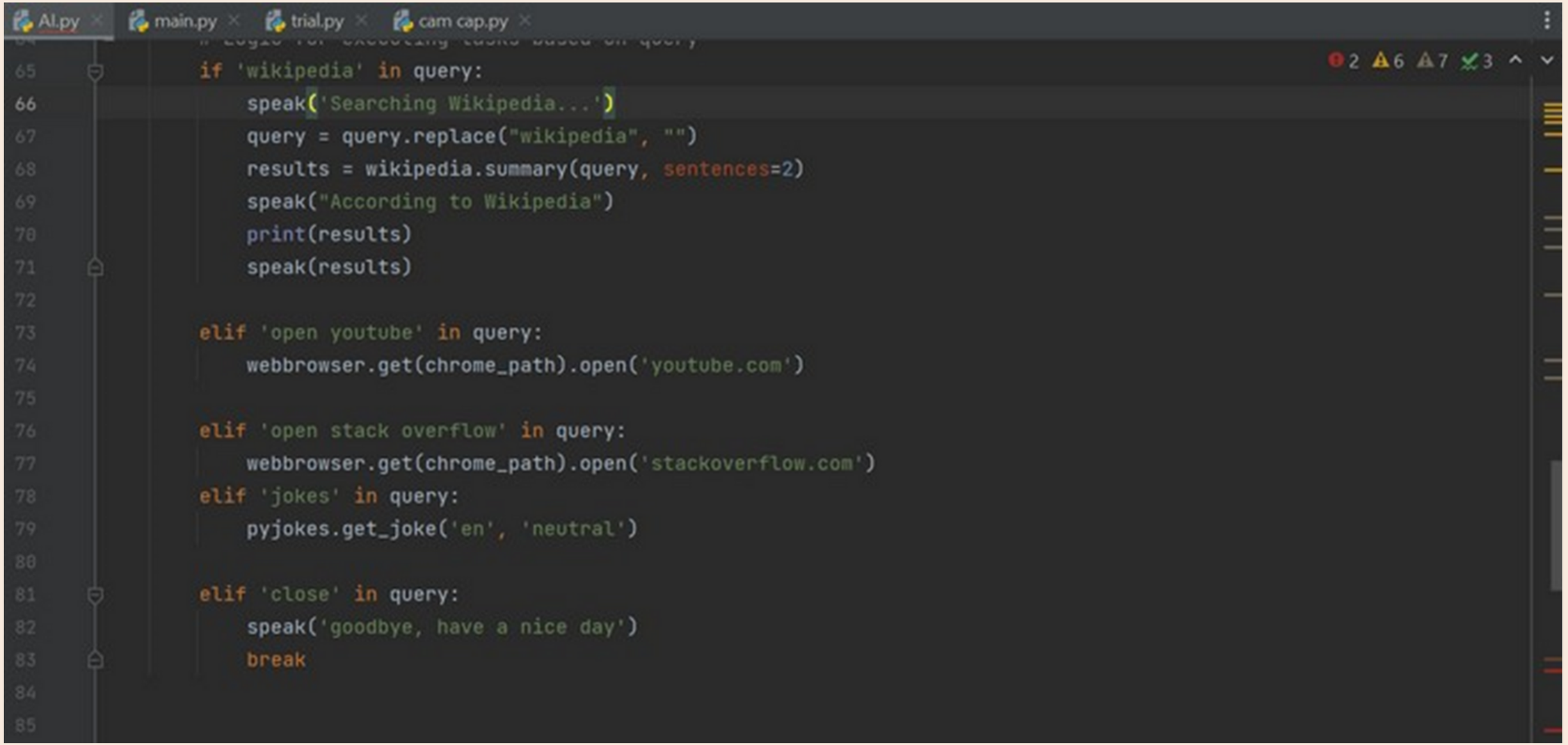
AI.py | main.py | trial.py | cam cap.py

```
def wishMe():
    hour = int(datetime.datetime.now().hour)
    if hour>=0 and hour<12:
        speak("Good Morning!")

    elif hour>=12 and hour<18:
        speak("Good Afternoon!")

    else:
        speak("Good Evening!")

    speak("I am tiffany. Please tell me how may I help you")

def takeCommand():
    #It takes microphone input from the user and returns string output

    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("Listening...")
        r.pause_threshold = 1
        audio = r.listen(source)

```

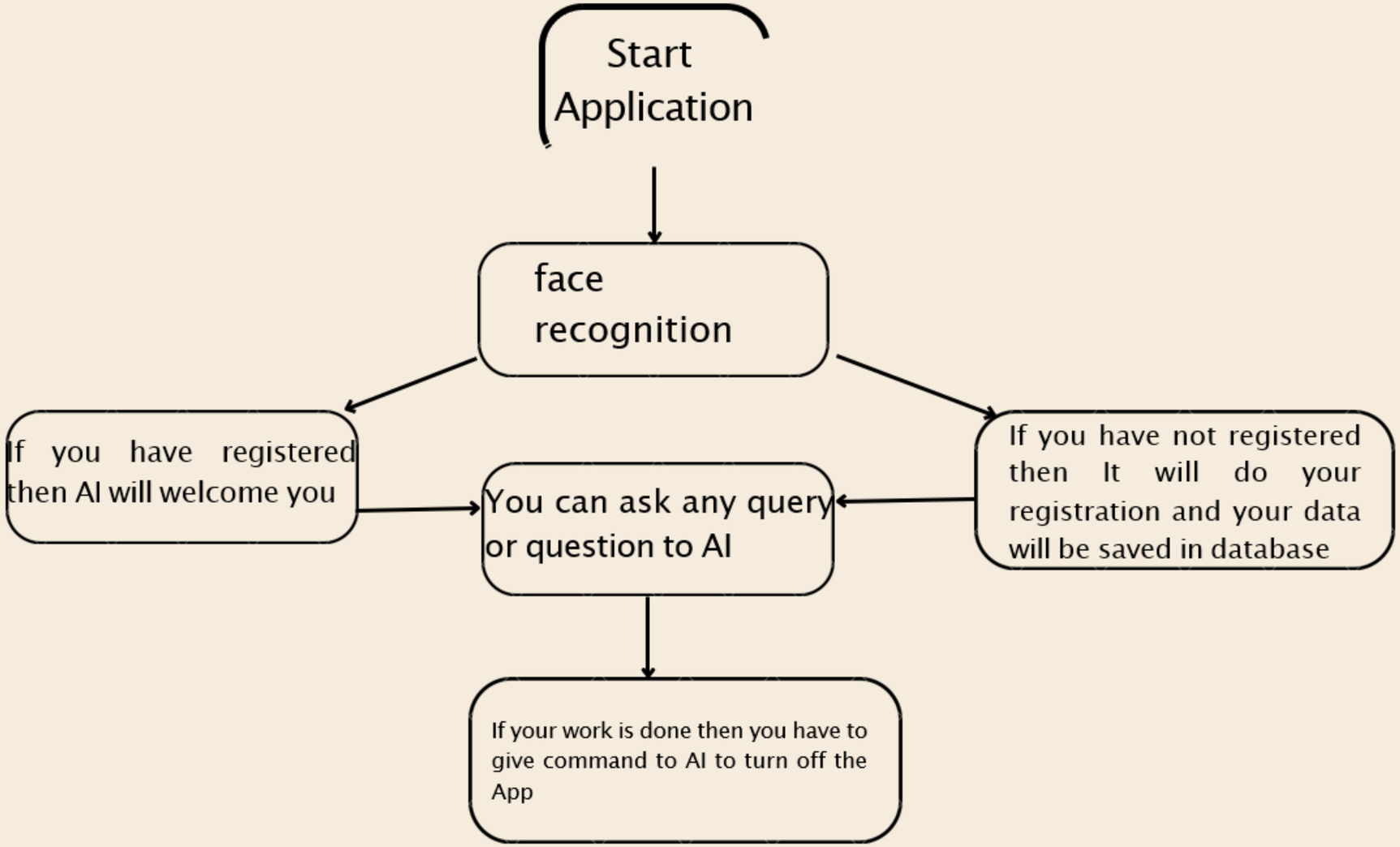
6 19 3

if __name__ == "__main__" › while True › elif 'close' in query


AI.py
main.py
trial.py
cam cap.py
2 6 7 3
try:
print("Recognizing...")
query = r.recognize_google(audio, language='en-in')
print(f"User said: {query}\n")
except Exception as e:
# print(e)
print("Say that again please...")
return "None"
return query
chrome_path = 'C:/Program Files (x86)/Google/Chrome/Application/chrome.exe %s'
if __name__ == "__main__":
wishMe()
while True:
# if 1:
query = takeCommand().lower()
# Logic for executing tasks based on query
if 'wikipedia' in query:
speak('Searching Wikipedia...')
query = query.replace("wikipedia", "")
if __name__ == "__main__"


AI.py
main.py
trial.py
cam cap.py
2 6 7 3
if 'wikipedia' in query:
speak('Searching Wikipedia...')
query = query.replace("wikipedia", "")
results = wikipedia.summary(query, sentences=2)
speak("According to Wikipedia")
print(results)
speak(results)
elif 'open youtube' in query:
webbrowser.get(chrome_path).open('youtube.com')
elif 'open stack overflow' in query:
webbrowser.get(chrome_path).open('stackoverflow.com')
elif 'jokes' in query:
pyjokes.get_joke('en', 'neutral')
elif 'close' in query:
speak('goodbye, have a nice day')
break

MODULE WORK FLOW EXPLANATION.



IMPLEMENTATION AND CODING.

1). Libraries and Pre-Define Inputs

```python
# libraries and modules used
import cv2
import numpy as np
import face_recognition as fr
import os
import pyttsx3
import datetime
import webbrowser
import speech_recognition as sr
from random import choice
import subprocess
import wikipedia
import pywhatkit as kit
import traceback
from tkinter import *
from PIL import Image, ImageTk
import time
# main code
query=''
root = Tk()
root.wm_geometry("1920x1080")
root.wm_minsize(550, 950)
root.wm_maxsize(1920, 1080)
```

```
root.wm_iconwindow()
root.wm_title("Tiffany")
img = Image.open("Hello I'M Tiffany.png")
width, height = img.size

photo = ImageTk.PhotoImage(img)
pic = Label(root, image=photo)
pic.config(highlightbackground="red")
pic.pack()
root.after(3000,lambda:root.destroy())
root.mainloop()
path = r'C:\Users\User\PycharmProjects\pythonProject\main\tiffany\rishi data'
image = []
classn = []
mlist = os.listdir(path)
print(mlist)
for cl in mlist:
    l = cv2.imread(f'{path}/{cl}')
    image.append(l)
    classn.append(os.path.splitext(cl)[0])
print(classn)

```

```
engine = pyttsx3.init()
voices = engine.getProperty("voices")
engine.setProperty("voice", voices[1].id)
engine.setProperty("rate", 150)
bot_name = "Tiffany"

paths = {
    'notepad': "C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Accessories\\Notepad",
    'google chrome': 'C:/Program Files (x86)/Google/Chrome/Application/chrome.exe %s',
    'calculator': "C:\\Windows\\System32\\calc.exe",

}
positive_response = ["Cool, I am on it sir!", " Okay sir, I'm working on it!", "Just a second sir!"]
negative_response = ["I think its invalid Command ", "My inventor didn't taught me this!",
                     "Sorry!, i dont know how to do this"]
gratitude = ["I am happy to help!", "My pleasure sir!", "No problem!"]

```

2). Pre Define Function

```
64  def speak(text):
65      engine.say(text)
66      engine.runAndWait()
67
68  def unauthorized():
69      speak("You are unauthorized")
70
71  def take_command():
72      r = sr.Recognizer()
73      with sr.Microphone() as source:
74          speak("listening")
75          print('Listening....')
76          r.pause_threshold = 2
77          audio = r.listen(source, timeout=2,phrase_time_limit=7 )
78
79      try:
80          print('Recognizing...')
81          query = r.recognize_google(audio, language='en-in')
82          if 'exit' in query or 'stop' in query:
83              hour = datetime.datetime.now().hour
84              if 21 <= hour < 6:
85                  speak("Good night sir, take care!")
86              else:
```

```
87                  speak('Have a good day sir!')
88              exit()
89          return query
90
91
92
93      except Exception as e:
94          traceback.print_exc()
95          speak('Sorry, I could not understand. Could you please say that again?')
96          query = take_command()
97      return query
98
99  def validate_command(query):
100     if "text file" in query:
101         speak(choice(positive_response))
102         new_text_file()
103     elif "command prompt" in query:
104         speak(choice(positive_response))
105         open_cmd()
106     elif "calculator" in str.lower(query):
107         speak(choice(positive_response))
108         open_calculator()
109     elif "notepad" in str.lower(query):
```

```
109     elif "notepad" in str.lower(query):
110         speak(choice(positive_response))
111         open_notepad()
112     elif "google" in str.lower(query):
113         speak(choice(positive_response))
114         open_google()
115     elif "send a WhatsApp message" in query:
116         speak(choice(positive_response))
117         send_whatsapp_message()
118     elif "according to wikipedia" in query:
119         speak(choice(positive_response))
120         search_on_wikipedia()
121     elif "thank you" in query:
122         speak(choice(gratitude))
123
124     elif "close" in query:
125         speak('goodbye, have a nice day')
126     else:
127         speak(choice(negative_response))
```

```python
def take_user_input():
    r = sr.Recognizer()
    with sr.Microphone() as source:
        print('Listening....')
        r.pause_threshold = 2
        audio = r.listen(source, phrase_time_limit=7)
        try:
            print('Recognizing...')
            query = r.recognize_google(audio, language='en-in')
        except Exception:
            speak('Sorry, I could not understand. Could you please say that again?')
            query = 'None'
        return query

def search_on_wikipedia():

    query = take_command()
    results = wikipedia.summary(query, sentences=2)
    print(results)
    speak(results)
```

```
150     def search_on_google():
151         query = take_command()
152         webbrowser.get(paths['google chrome']).open(query)
153
154     def open_cmd():
155         os.system('start cmd')
156
157     def open_calculator():
158         subprocess.Popen(paths['calculator'])
159
160     def open_google():
161         webbrowser.get(paths['google chrome']).open('www.google.com ')
162
163     def open_notepad():
164         os.startfile(paths['notepad'])
165
166     def open_opera():
167         os.startfile(paths['opera'])
```

```
169  def new_text_file():
170      # make new notepad file
171      speak("What will be the name of the file? ")
172      file_name = take_user_input()
173      file = "C:\\Users\\rishi Mukesh Jain\\desktop\\"+file_name + ".txt"
174      f = open(file, "w+")
175      speak("What do you want to write into the file")
176      content = take_user_input()
177      f.write(content)
178      speak("New text file made with name " + file_name)
179      f.close()
180
181  def send_whatsapp_message():
182      speak("whom do you want to send? please enter on console")
183      number = input("enter the number: ")
184      speak("what do you want to send")
185      message = take_user_input()
186      kit.sendwhatmsg_instantly(f"+91{number}", message)
187
188      file_name = take_user_input()
189
```

```
189
190 def greet():
191     current_time = datetime.datetime.now().hour
192
193     if 4 < current_time <= 12:
194         speak("Good morning " + name + "!")
195     elif 12 < current_time <= 16:
196         speak("Good afternoon " + name + "!")
197     elif 16 < current_time <= 21:
198         speak("Good evening " + name + "!")
199     elif 21 < current_time < 23 or 0 <= current_time <= 4:
200         speak("Its a late night " + name + "!")
201
202     speak("What would you like to do ?")
203
204 def findencoding(image):
205     encodel=[]
206     for img in image:
207         img=cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
208         encode = fr.face_encodings(img)[0]
209         encodel.append(encode)
210     return encodel
211
```

3).Program Structure

```
214     e = findencoding(image)
215     print('encoding complete')
216     cam = cv2.VideoCapture(0)
217     name = ''
218     timeout = time.time() + 5
219     while True:
220         test = 0
221         success, img = cam.read()
222         imgs = cv2.resize(img, (644,555), None, 0.25, 0.25, 0)
223         imgs = cv2.cvtColor(imgs, cv2.COLOR_BGR2RGB)
224
225         faclocreal = fr.face_locations(imgs)
226         encode1 = fr.face_encodings(imgs, faclocreal)
227
228         for encodeface, faceloc in zip(encode1, faclocreal):
229             m = fr.compare_faces(e, encodeface)
230             facdis = fr.face_distance(e, encodeface)
231             print(facdis)
232             matchindex = np.argmin(facdis)
233             print(matchindex)
234             if m[matchindex]:
235                 name += classn[matchindex]
236
```

```
236
237         cv2.waitKey(1)
238
239         print(name)
240         if name in classn:
241             break
242         else:
243             print('not reg')
244
245         if test == 5 or time.time() > timeout :
246             break
247         test = test - 1
248
249     print(name)
250     print(classn)
251
252     if name in classn:
253         r=Tk()
254         greet()
255         def tiffany(event):
256             query = take_command()
257             print("You said \" " + query + " \"")
258             validate_command(query)
259             return query
```

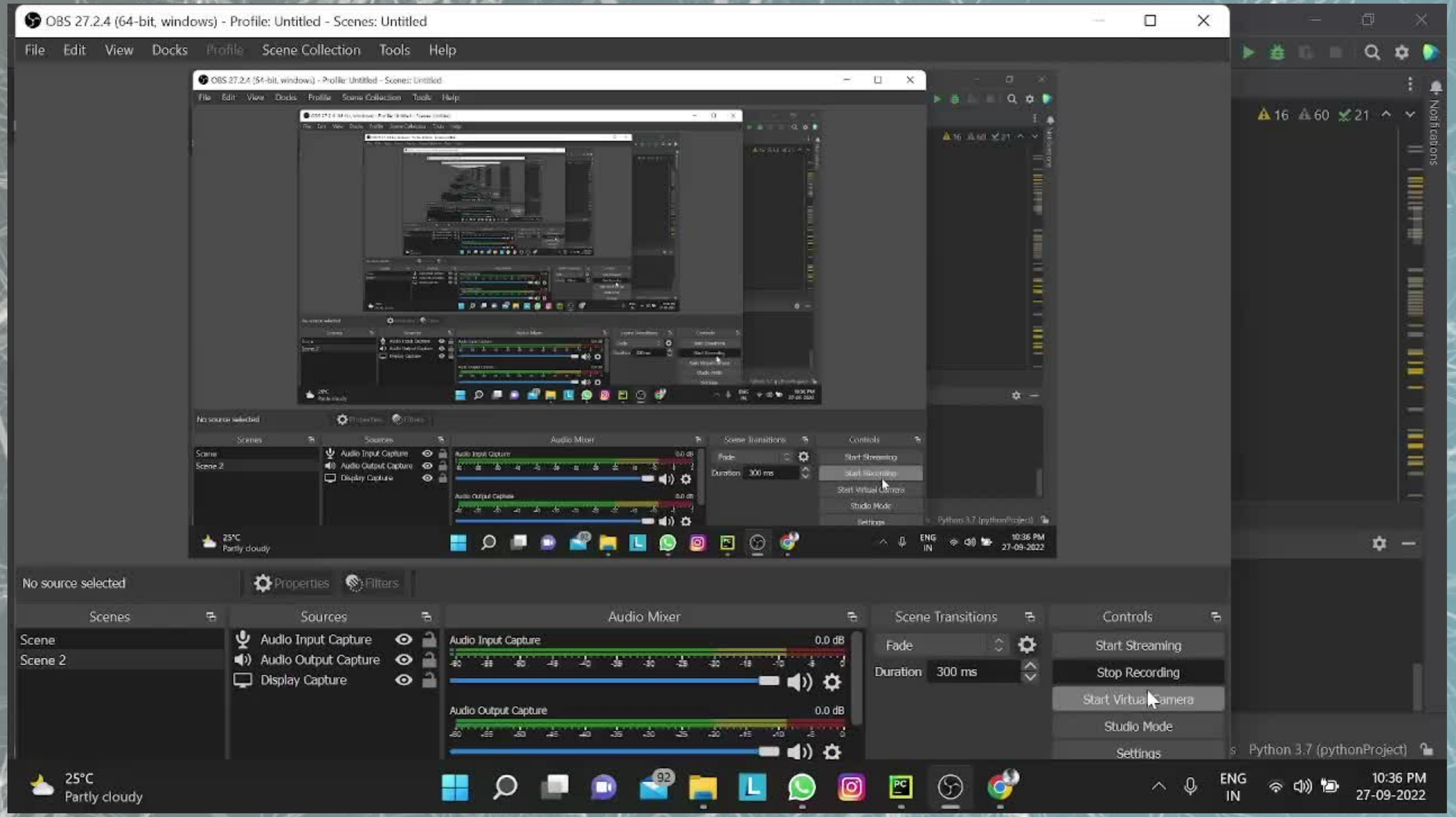
```
    r.wm_geometry('450x850')
    mic = PhotoImage(file="002-microphone.png")
    widget = Button(r, image=mic, bg="#151B54", borderwidth='0')
    widget.pack(side=BOTTOM, anchor=N, pady=300)
    widget.bind('<Button-1>', tiffany)
    r.configure(bg="#151B54")
    if 'close' in query:
        r.destroy()

    r.mainloop()
else:
    win = Tk()
    win.wm_geometry("890x600")
    Label(win, text="New User Portal", bg="#4863A0", fg="#50C878", font="comicsans 50 bold", borderwidth=3,
          relief='sunken').grid(row=0, column=3)
```

```
277    def reg_cap():
278        n=namevalue.get()
279        cam = cv2.VideoCapture(0)
280        cv2.namedWindow("test")
281        d = r'C:\Users\User\PycharmProjects\pythonProject\main\tiffany\rishi data'
282        os.chdir(d)
283        while True:
284            ret, frame = cam.read()
285            if not ret:
286                print("failed to grab frame")
287                break
288            cv2.imshow("test", frame)
289
290            k = cv2.waitKey(1)
291            if k % 256 == 27:
292                # ESC pressed
293                print("Escape hit, closing...")
294                break
295            elif k % 256 == 32:
296                # SPACE pressed
297                img_name = "{}.png".format(n)
298                cv2.imwrite(img_name, frame)
299                print("{} written!".format(img_name))
```

16 75 21

```
302         cam.release()
303
304         cv2.destroyAllWindows()
305
306     name = Label(win, text="Name", bg="#342D7E", fg="#50C878")
307     n = Label(win, text="* hit spacebar to click", bg="#342D7E", fg="#50C878", font="comicsans 10 bold")
308     n1 = Label(win, text="* hit esc to exit camera portal", bg="#342D7E", fg="#50C878", font="comicsans 10 bold")
309     n2 = Label(win, text="* once registration is done you can exit and try again", bg="#342D7E", fg="#50C878",
310                font="comicsans 10 bold")
311
312     name.grid(row=3, column=2)
313     n.grid(row=11, column=3)
314     n1.grid(row=12, column=3)
315     n2.grid(row=14, column=3)
316     namevalue = StringVar()
317     nameentry = Entry(win, textvariable=namevalue)
318     nameentry.grid(row=3, column=3)
319     Button(win, fg="red", text="click your photo", command=reg_cap, anchor=S).grid(row=9, column=3)
320     win.configure(bg="#342D7E")
321     if namevalue.get() in classn:
322         n2 = Label(win, text="* registered you can exit and try again", bg="#342D7E", fg="#50C878",
323                    font="comicsans 10 bold")
324         n2.grid(row=14, column=3)
325     win.mainloop()
```

DEMO VIDEO

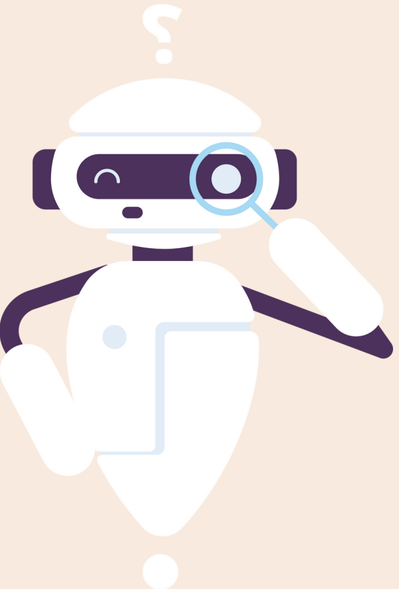

OBS 27.2.4 (64-bit, windows) - Profile: Untitled - Scenes: Untitled
File Edit View Docks Profile Scene Collection Tools Help
No source selected
Properties
Filters
Scenes
Scene
Scene 2
Sources
Audio Input Capture
Audio Output Capture
Display Capture
Audio Mixer
Audio Input Capture 0.0 dB
Audio Output Capture 0.0 dB
Scene Transitions
Fade
Duration 300 ms
Controls
Start Streaming
Stop Recording
Start Virtual Camera
Studio Mode
Settings
25°C
Partly cloudy
ENG
IN
10:36 PM
27-09-2022
Python 3.7 (pythonProject)

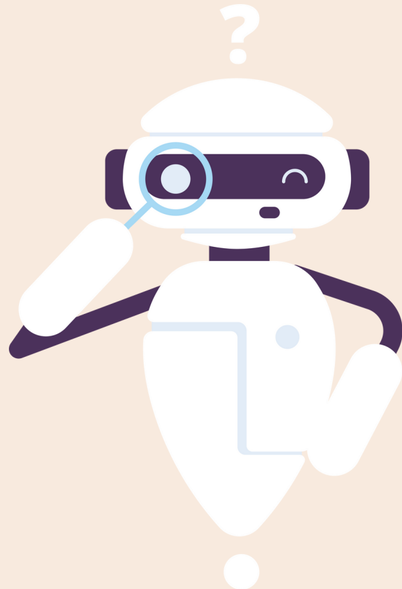
RESULT AND DISCUSSION (INPUT AND OUTPUT)

WHEN THE USER STARTS THE APPLICATION, IT WILL START WITH RECOGNIZING THE USER`S FACE AND WILL CHECK IF YOU HAVE REGISTERED YOUR FACE AND NAME OR NOT. IF YOU HAVE REGISTERED YOUR FACE AND NAME THEN IT WILL DIRECTLY GIVE YOU ACCESS TO AI/VOICE ASSISTANT BUT IF YOU HAVE NOT REGISTERED THEN IT WILL START WITH REGISTERING YOUR FACE AND NAME FOR FACE RECOGNITION AND WILL SAVE IT IN ITS DATABASE AND THEN IT WILL WELCOME YOU AND THEN YOU CAN ASK THEM ANY QUERY YOU HAVE. WHEN YOUR WORK IS DONE WITH THE VOICE ASSISTANT THEN YOU CAN SIMPLY COMMAND THE AI TO TURN OFF THE APP.

CONCLUSION

FACE RECOGNITION IS AN UPCOMING TECHNOLOGY THAT CAN PROVIDE MANY BENEFITS. IT CAN SAVE RESOURCES AND TIME, AND EVEN GENERATE NEW INCOME STREAM, FOR COMPANIES.VOICE ASSISTANCE MAY BE A TOOL THAT WE CANNOT LIVE WITHOUT IN FUTURE TIMES. VOICE ASSISTANCE WILL GET ACCESS WHEN THE FACE IS RECOGNIZED. THIS WILL BE THE FUTURE OF OUR TECHNOLOGY.





THANK YOU FOR LISTENING!