

Karmayogi Polytechnic College, Shelve

Department of Computer Technology

Micro-Project Report

Programming with Python (22616)



“Voice assistant for covid19 symptoms identification.”

Guided by:

Prof. M. N. Shaikh

Presented By:

- 1. Mst. Bagal Rohit Ravindrakumar (CM 312)**
- 2. Mst. Rokade Rohan Shivaji (CM 328)**
- 3. Mst. Lokare Prajval Dattatray (CM 337)**
- 4. Mst. Pawar Sachin Namdev (CM 352)**

During Academic Year 2020-21

Title of Micro-Project:

Voice assistant for covid19 symptoms identification.

1.0 Brief Description:

The novel coronavirus (COVID-19) was first observed in late 2019 in Wuhan, China, and the patients suffered from a form of pneumonia. The virus was identified as genus beta coronavirus, placing it in the same category as the previously discovered deadly viruses such as Severe Acute Respiratory Syndrome (SARS) and Middle East Respiratory Syndrome (MERS). The virus has now spread across more than 200 countries. The WHO declared this virus a global health emergency in early 2020. National emergency was declared in the US in March 2020. More than 540,000 people have died from this virus across the globe, with more than 130,000 deaths in the US alone as of July 5, 2020.

Hospitals and clinics around the world have been overwhelmed with the cases of COVID-19. With a lot of panic and rumors, people are visiting clinics and hospitals for other non-related symptoms. These visits are causing increased healthcare costs and spread of infection, while overloading the healthcare system. Self-assessment is therefore being studied as one of the solutions to this problem. This technique, i.e. self-assessment, has been used in the field of healthcare for a long time as it helps in learning, functioning more effectively, and fostering self-agency and authority.

Some recent examples of applications that have been built to self- assess the COVID-19 are. Though these applications are beneficial, they might not be accessible to all. Moreover, such apps are not useful for someone who does not know how to read, use a computer, or is visually impaired. Therefore, in this paper, we propose a novel idea to use the IVACS for self-assessment of the COVID-19. This interactive application, based on medical condition, helps in more precise clinical decision making about seeking medical care or taking rest at home; without burdening the hospitals at these challenging times. It also educates people with critical information. Primary contributions of this proposed work are below:

- A new IVACS architecture for the self-assessment of COVID-19,
- A detailed study on the performance of the proposed IVACS,
- A study on the performance of the user in cohesion with IVACS,
- A measurement of the perceived mental overload in user due to IVACS.

2.0 Aim of Micro Project:

At the time of writing this paper, the world has around eleven million cases of COVID-19, scientifically known as severe acute respiratory syndrome corona-virus 2 (SARS-COV2). One of the popular critical steps various health organizations are advocating to prevent the spread of this contagious disease is self-assessment of symptoms. Multiple organizations have already pioneered mobile and web-based applications for self-assessment of COVID-19 to reduce the spread of this global pandemic. We propose an intelligent voice-based assistant for COVID19 self-assessment (IVACS). This interactive assistant has been built to diagnose the symptoms related to COVID-19 using the guidelines provided by the Centers for Disease Control and Prevention (CDC) and the World Health Organization (WHO).

3.0 Actual Procedure Followed:

Sr.No.	Details of Activity	Name of responsible Team members
1	Identify the problem	Mst. Bagal Rohit Ravindrakumar
2	Preparation of project proposal	Mst. Rokade Rohan Shivaji
3	Collect relevant data /literature review	Mst. Lokare Prajval Dattatray
4	Analysis of data	Mst. Pawar Sachin Namdev
5	Prototype /Model	Mst. Pawar Sachin Namdev
6	Report preparation	Mst. Rokade Rohan Shivaji
7	Presentation of micro project	Mst. Bagal Rohit Ravindrakumar

4.0 Actual Resources Used:

Sr. No	Name of Resource/material	Specifications	Quantity	Remarks
1	Computer System	Computer (i3-i5 preferable RAM>2GB)	1	-
2	Operating System	Windows/Linux	1	-
3	Development Software	Python IDE	1	-

5.0 Source code:

```
from tkinter import *
import pyttsx3 #text to speech package
import speech_recognition as sr #speech speech recognition package
import datetime #date and time package (built in)
```

```
#Complete UI design...
```

```
#Actual code for assistant starts here...
```

```
#declaring Global Variables
fever=None
contact=None
cough=None
tiredness=None
pains=None
sourThroat=None
rash= None
difficultiesInBreathing=None
chestPain=None
lossSpeech=None
```

```
lossTeast=None
```

```
score=0
```

```
engine=pyttsx3.init('sapi5') #initilizing the microsoft speech API
voices=engine.getProperty('voices') # get built in voice from microsoft API and set in voices variable
# print(voices[1].id)
engine.setProperty('voice',voices[1].id) # set the engine voice
```

```
def speak(audio): #speech methtod from text to speech
    engine.say(audio)
    engine.runAndWait()
```

```
def wish(): #this method will wish you after exicuting the project
    hour = int(datetime.datetime.now().hour) #we need ol=nly hour for wishing
    if hour>=0 and hour<12:
        speak('Good Morning')
    elif hour>=12 and hour<18:
        speak("Good Afternoon")
    elif hour>=18 and hour<21:
        speak("Good Evening")
    check()
```

```
def takevoice(): #this method will listen user voice
    r=sr.Recognizer() #this is for recognizing users voice
    with sr.Microphone() as source: #exception handaling
        print("Listening...")
        r.pause_threshold=1
        audio= r.listen(source)
    try:
        print("Recognizing Your Ansewer...")
        query=r.recognize_google(audio,language="english") #trying to recognize vice by using google API
    except Exception as e:
        print("Cant recognize your answer please say is again...")
        takevoice()
    return "None"
return query
```

```
def check():
    global score #accessing Global Variable

    speak("What is your name")
    user=takevoice().lower()

    speak(f"{user} do you have fever")
    global fever
    fever=takevoice().lower()
    if fever=="yes":
```

```
score +=10
# fever=True
```

```
speak(f"{user} do you have contact with covid patient in last 14 days")
global contact
contact=takevoice().lower()
if contact == "yes":
    score +=10
```

```
speak(f"{user} do you have dry cough")
global cough
cough=takevoice().lower()
if cough == "yes":
    score +=10
```

```
speak(f"{user} do you have tiredness")
global tiredness
tiredness=takevoice().lower()
if tiredness == "yes":
    score +=10
```

```
speak(f"{user} do you have Headach and body pains")
global pains
pains=takevoice().lower()
if pains == "yes":
    score +=10
```

```
speak(f"{user} do you have rash on skin")
global rash
rash=takevoice().lower()
if rash == "yes":
    score +=5
```

```
speak(f"{user} do you have sour Throat")
global sourThroat
sourThroat=takevoice().lower()
if sourThroat == "yes":
    score +=10
```

```
speak(f"{user} do you have difficulties in breathing or shorten of breath")
global difficultiesInBreathing
difficultiesInBreathing=takevoice().lower()
if difficultiesInBreathing == "yes":
```

```
score +=30
```

```
speak(f"{user} do you have cheast pain or cheast pressure")
global cheastPain
cheastPain=takevoice().lower()
if cheastPain == "yes":
    score +=20
```

```
speak(f"{user} do you loss speach or body movement")
global lossSpeech
lossSpeech=takevoice().lower()
if lossSpeech == "yes":
    score +=30
```

```
speak(f"{user} do you loss of test or smell")
global lossTeast
lossTeast=takevoice().lower()
if lossTeast == "yes":
    score +=30
```

```
testCases()
```

```
def testCases():
    # speak(f"you have {fever}")
    # speak(f"you have {contact}")

    #print(score)
    #main logic of the project
    if score>90:
        speak("you have covid ")
    elif score>=75 and score<100:
        speak("You Have highest chansen of having Covid Go to test covie center for testing")
    elif score>=70 and (difficultiesInBreathing or cheastPain or lossSpeech or lossTeast):
        speak("you have chances to have covid consalt a doctor")
    elif score>=30 and score>50 and (difficultiesInBreathing or cheastPain or lossSpeech or lossTeast):
        speak("you must become home qurant in for 14 days")
    elif score>35 and score<70:
        speak("you must become home qurantine for 7 days")
    elif score<=30 :
        speak("you have no covid, Have a great day...")

    speak("Thanks for using our software...")
```

```

# elif score>70
# if __name__=="__main__":
#     if btn:
#         wish()
#         check()

# window.mainloop()

window=Tk()
# add widgets here

window.title('Covid19 Voice Assistance For symptoms Identification')
lbl=Label(window, text="Voice Assistant", fg='red', font=("Algerian", 16)) #label...
lbl.place(x=60, y=30) #placing...

btn=Button(window, text="  Start  ", fg='blue',command=wish)#button for starting assistant...
btn.place(x=120, y=80)

# Creating a photoimage object to use image
photo = PhotoImage(file = r"C:\Users\USER\OneDrive\Pictures\Camera Roll\listening.png")#icons for listening
view
photo1= PhotoImage(file = r"C:\Users\USER\OneDrive\Pictures\Camera Roll\speaking1.png")#icons for
speaking view

# here, image option is used to
# set image on button
btn1 = Button(window, text = 'Click Me !', image = photo).pack(side = LEFT, padx=20,pady=50)#button for
listening view
btn2 = Button(window, text = 'Click Me !', image = photo1).pack(side = RIGHT, padx=20,pady=50)#button for
speaking view

window.geometry("300x200+10+20")

window.mainloop()

```

6.0 Outputs of the Micro-projects:



7.0 Skill Developed/ learning out of this Micro-Project:

Through this project we've learnt how to develop GUI in Python in Tkinter and came to know some new packages while developing this project, how to install those packages, import it in a file, use those package, what is purpose of those packages, how it works.