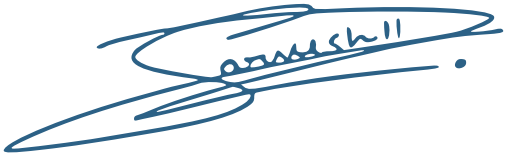
CS PROJECT

Virtual Assistant

➔ **Sarvesh Dakhore**

➔ **Devank Yadav**



Signature (Devank Yadav)

Signature (Sarvesh Dakhore)

MAY 30, 2021

Acknowledgement

It gives me immense pleasure to express my gratitude towards my computer science teacher Mr Kailash Mehta whose valuable guidance, constant support, constructive comments and sympathetic attitude has been the light that helped me patch this project and make it a full-proof success. His suggestions and instructions have served as the major contributor towards the completion of this project. I would like to thank him for giving me this opportunity to hone my skills.

Secondly, I would thank my parents and friends for their heart pledged to support that helped me complete the project within a limited time.

Certificate

This is to certify that Devank Yadav and Sarvesh Dakhore, students of Class 12 S3, have successfully completed the project titled Virtual Assistant. The project has been satisfactorily performed during the academic year 2021-22 towards partial fulfilment of COMPUTER SCIENCE PRACTICAL EXAMINATION conducted by CBSE.

INTERNAL EXAMINER EXTERNAL EXAMINER

INDEX

PROJECT (Virtual Assistance)

```
enter password of mysql localhost server:
```

```
14
15
16  import mysql.connector
17  from mysql.connector import Error
18
19  apps_list_win=[[""],["MsEdge", "Microsoft Edge","edge","edge browser","micrisoft browser"],["chro
20
21  while True:
22      pas=str(input("enter password of mysql localhost server: "))
23      try:
24          mydb = mysql.connector.connect(host="localhost",user="root",passwd=pas,charset = 'utf8')
25          pop=0
26      except Error:
27          print("wrong password... Try again \n ------------------------------------")
28          pop=1
29      if pop==0:
30          break
```

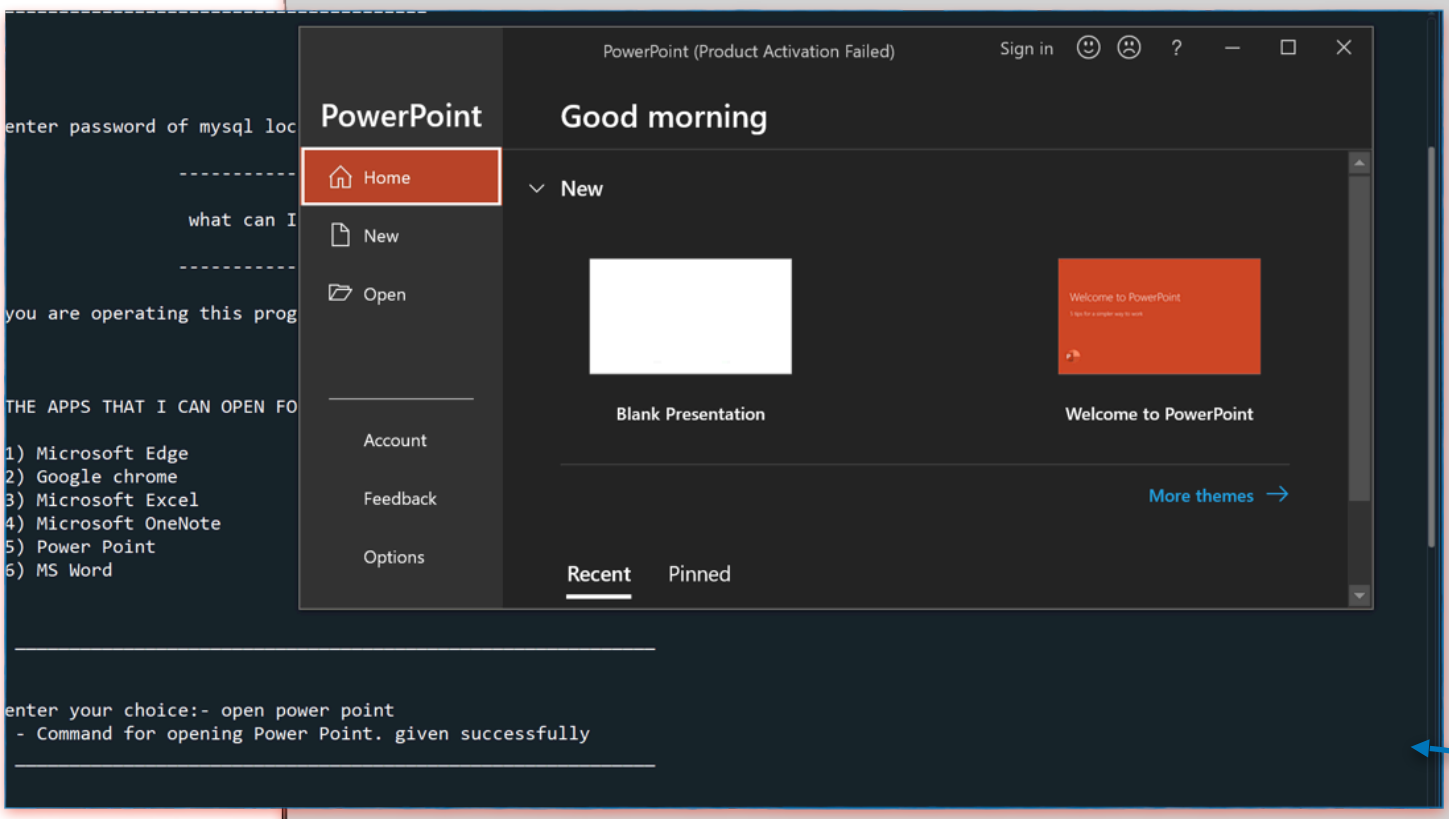
- ➔ After execution of code, It will ask for the pre-set password of the localhost server and user name "**root**" of the default server
- ➔ input() function used in line "**22**", and saved in variable "**pas**".

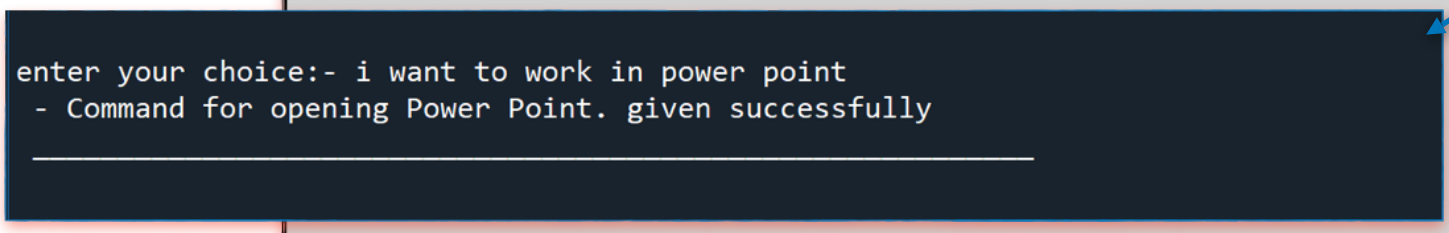
```
enter password of mysql localhost server: [redacted]

                ------------------------------

                 what can I help you with ?

                ------------------------------

you are operating this program on windows.....


THE APPS THAT I CAN OPEN FOR YOU :

1) Microsoft Edge
2) Google chrome
3) Microsoft Excel
4) Microsoft OneNote
5) Power Point
6) MS Word


 ____________________________________________________


enter your choice:-
```

- Now our program will look for a database, if available it will check for a table of name and mac address of that machine but if anything is missing in the table it will create it, we have added several applications by default which you can open
- In line "**29**" show_apps() func. Is defined which retrieve data from database

```
43
44    to_clear_screen=[]
45    app_name=""
46    print("\n\t\t\t\t---------------------------------- \n")
47    print("\t\t\t\t what can I help you with ?")
48    print("\n\t\t\t\t---------------------------------- \n")
49
50    voice_control=0
51    pyttsx3.speak("How Can I help You with .")
52    show_apps()
53
```

```
29    def show_apps():
30        print("\n \n")
31        print("THE APPS THAT I CAN OPEN FOR YOU : \n")
32        x=database.retrive_app_data()
33        for i in range(1,len(x)):
34            q=str(i)
35            print(q+") "+x[i][1])
36        print("\n")
37
```





➔ Our program understands the English language

➔ So, Now Let's try to communicate with it and give it tasks

➔ We gave the command for opening the MS PowerPoint and it returns after completion of the task

```
enter your choice:- 6
 - Command for opening MS Word. given successfully
 ____________________________________________________


enter your choice:- 8
Unable to understand
 ____________________________________________________
```

```
322         if to_control==1 and to_control_understanding==1:
323                 if req_2.isnumeric():
324                     req_2=int(req_2)
325                     if req_2 in range(0,len(x)):
326                         row=req_2
```

➔ Our Program is a Dynamic software which updates itself over-time

➔ we can also open apps and software by the auto-assigned number given by the program.

➔ From line "**322-326**", its checking whether number is in range or not, if we go out of range, it will go for NLP, if still it docent found anything it will not process anything and prompt a message unable to understand

```
enter your choice:- do not open power point
It's a negative comment so i can't proceed.
 ____________________________________________________
```

```
311     elif to_control==1 and to_control_app_opening==1:
312         x=database.retrive_app_data()
313         for h in sentece_postm:
314             if (h in negative_comment) and h not in speak_turn_off :
315                 print(negative_comment_speach)
316                 if speak_mode==1:
317                     pyttsx3.speak(negative_comment_speach)
318                 break
319             if (h in negative_comment) and (h in speak_turn_off):
320                 break
```

➔ From line "**322-326**", its checking whether the input lies in negative comment parameters or not, if not then it gonna process ahead, else it will break the operation.

```
321
322         if to_control==1 and to_control_understanding==1:
323                 if req_2.isnumeric():
324                     req_2=int(req_2)
325
326                 if req_2 in range(0,len(x)):
327                     row=req_2
328
329                 else:
330                     for r in range(0,len(x)):
331                         for d in x[r]:
332                             if d in req_1:
333                                 row=r
334                                 break
```

```
337
338 if row ==0 and to_control_understanding==1:
339     print("Unable to understand")
340     if speak_mode==1 and voice_control==0:
341         pyttsx3.speak("Unable to understand")
342 else:
343     app_name=x[row][0]
344     if platform.system()=="Windows":
345         if os.system('start ' +app_name) == 1:
346             opo=0
347             print("App does not exist. \n It has been removed as its not existed \n")
348             print(line+"\n")
349             database.remove_app_sql(row)
350
351         if "quit" in sentece_postm:
352             to_opo=0
353             wer=app_name
354             os.system("taskkill /im "+wer+".exe /f")
355
356         else:
357             app_name1=x[row][1]
358             print(" - Command for opening "+app_name1+". given successfully")
359
360             if speak_mode==1:
361                 pyttsx3.speak("Command for opening "+app_name1+". given successfully")
362
```

- ➔ From line ”**321 - 334**” the calculation of row takes place
- ➔ The var. “row” represents the app serial number which given by code.
- ➔ In line “**351**” its opening app and checking the exit code with an “**1**”. “**1**” is the exit code which is given when the app does not exist.
- ➔ If error code does not match then it checks whether the user wants to quit the app or not if yes it will kill the app if not that it will execute else.

```
enter your choice:- do not speak

 - speaking mode turned off successfully
 ____________________________________________________


enter your choice:- do not speak
It's a negative comment so i can't proceed.
 ____________________________________________________


enter your choice:-
```

➔ Our program is also inclined towards the needs of physically challenged people and allows the text to speak

➔ As our program is NLP enabled, it processes and if the input is negative and there is no command then it will say “negative comment”

```
161
162     if speak_mode==1 and to_control_switch==0:
163         for h in speak_turn_off:
164             if h in requirements:
165                 if speak_mode==0:
166                     break
167                 else:
168                     for k in negative_comment:
169                         if k in requirements:
170                             speak_mode=0
171                             to_control=0
172                             print("\n - speaking mode turned off successfully")
173                             break
174             elif speak_mode==0:
175                 break
176
177     elif speak_mode==0 and to_control_switch==1:
178         for h in speak_turn_off:
179             if h in requirements:
180                 if speak_mode==1:
181                     break
182                 else:
183                     for k in positive_comment:
184                         if k in requirements:
185                             speak_mode=1
186                             to_control=0
187                             print("\n - speaking mode turned on successfully")
188                             pyttsx3.speak("speaking mode turned on successfully")
189                             break
190             elif speak_mode==1:
191                 break
```

```
enter your choice:- fghmdjtyrhnhmfsjt
Unable to understand
```

➔ If the user enters something that is misspelt or wrong then our program will respond by “Unable to Understand”

```
enter your choice:- turn on voice command

 - Voice control turned on successfully
 ____________________________________________________________

Say what you want?

hello

 Speech done.....
 ____________________________________________________________


Unable to understand
 ____________________________________________________________

Say what you want?

it's a testing

 Speech done.....
 ____________________________________________________________


Unable to understand
 ____________________________________________________________
```

➔ Whatever you speak it recognises and converts it into text

➔ Our program is also inclined towards the needs of physically challenged people and allows the speech to text.

```
127
128         if voice_control==1 and to_control_switch==0:
129             for h in voice_off:
130                 if h in requirements:
131                     if voice_control==0:
132                         break
133                     else:
134                         for k in negative_comment:
135                             if k in requirements:
136                                 voice_control=0
137                                 to_control=0
138                                 print("\n - Voice control turned off successfully")
139                                 pyttsx3.speak("Voice control turned off successfully")
140                                 break
141                 elif voice_control==0:
142                     break
143
144         elif voice_control==0 and to_control_switch==1:
145             for h in voice_off:
146                 if h in requirements:
147                     if voice_control==1:
148                         break
149                     else:
150                         for k in positive_comment:
151                             if k in requirements:
152                                 voice_control=1
153                                 to_control=0
154                                 print("\n - Voice control turned on successfully")
155                                 pyttsx3.speak("Voice control turned on successfully")
156                                 break
157                 elif voice_control==1:
158                     break
159
```

➔ The code above is to turn off and on voice control, by default its turned off,

➔ From line "**128 - 158**" works to determine whether the switch should be turned on or off.

```
Say what you want?

Chrome

 Speech done.....
 ____________________________________________________


 - Command for opening Google chrome. given successfully
 ____________________________________________________

Say what you want?

quit Chrome

 Speech done.....
 ____________________________________________________


 ____________________________________________________

Say what you want?

turn off voice control

 Speech done.....
 ____________________________________________________


 - Voice control turned off successfully
 ____________________________________________________

enter your choice:-
```

```
82      if voice_control==1:
83          r = sr.Recognizer()
84          with sr.Microphone() as source:
85              print("Say what you want? \n")
86              r.adjust_for_ambient_noise(source)
87              audio = r.listen(source,phrase_time_limit=3)
88              try:
89                  requirements = r.recognize_google(audio)
90              except sr.UnknownValueError or LookupError:
91                      print("No Input")
92                      requirements=""
93              print(requirements)
94              print("\n Speech done.....")
95              print(line+"\n \n")
96
```

➜ We can open or quit an app by using voice

➜ We imported "**speech_recognition**" module as "**sr**" from "**speechrecognition**" library.

```
enter your choice:- show apps



THE APPS THAT I CAN OPEN FOR YOU :

1) Microsoft Edge
2) Google chrome
3) Microsoft Excel
4) Microsoft OneNote
5) Power Point
6) MS Word


 ____________________________________________________________


enter your choice:-
```

➔If the user inputs “Show apps”, The program will show names of all assigned apps or the app data which is saved in our database

```
321
322         if to_control==1 and to_control_understanding==1:
323                 if req_2.isnumeric():
324                     req_2=int(req_2)
325
326                 if req_2 in range(0,len(x)):
327                     row=req_2
328
329                 else:
330                     for r in range(0,len(x)):
331                         for d in x[r]:
332                             if d in req_1:
333                                 row=r
334                                 break
```

```
enter your choice:- i want to remove app



THE APPS THAT I CAN OPEN FOR YOU :

1) Microsoft Edge
2) Google chrome
3) Microsoft Excel
4) Microsoft OneNote
5) Power Point
6) MS Word


 ____________________________________________________________


Enter number/to exit '0' :- 2
Google chrome Removed successfully !!



THE APPS THAT I CAN OPEN FOR YOU :

1) Microsoft Edge
2) Microsoft Excel
3) Microsoft OneNote
4) Power Point
5) MS Word



 ____________________________________________________________
 ____________________________________________________________
```

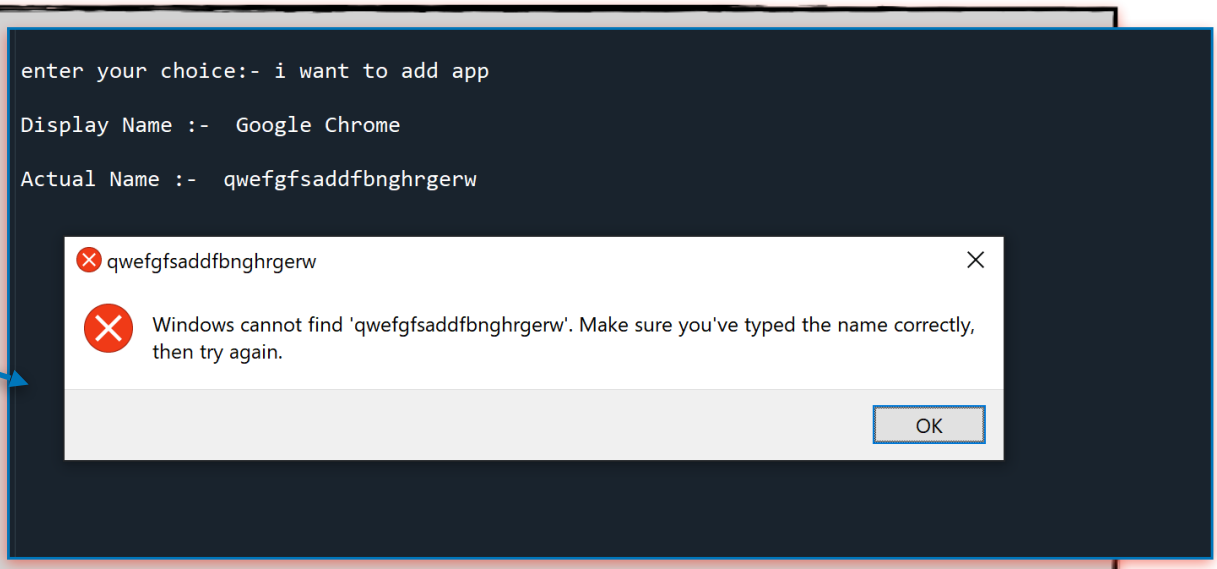
➔ The user can remove any app from the database, by giving input to remove an app and then by using the assigned no. of any app

```
205
206     if ("remove" in requirements or "delete" in requirements) and ("app" in re
207         to_control_app_opening=0
208         show_apps()
209         t=len(x)+1
210         print(line)
211         while True:
212             rm_app=""
213             print()
214             try:
215                 rm_app=int(input("Enter number/to exit '0' :- "))
216             except ValueError:
217                 pass
218             if rm_app==0:
219                 print("Exited!!")
220                 if speak_mode==1:
221                     pyttsx3.speak("Exited")
222
223             if rm_app in range(1,t):
224                 database.remove_app_sql(rm_app)
225                 print(x[rm_app][1]+" Removed successfully !!")
226                 show_apps()
227                 to_control_app_opening=0
228                 print("\n"+line)
229                 if speak_mode==1:
230                     print("\n")
231                     pyttsx3.speak(x[rm_app][1]+" Removed successfully")
232                 break
233             else:
234                 print("\n Wrong Input!!")
235                 print(line)
236                 if speak_mode==1:
237                     pyttsx3.speak("Wrong Input")
238
```

➔This piece of code checks whether the input is integer and in the range

➔If it found out of range then it prompts a message wrong input and ask for input again.

➔To exit remove app feature you just need to give “**0**” as input, hurray!! You exited, after this it will prompt message “Exited”



➔ The user can add any app in the database, by giving input to add app and then by giving the actual name of the app after that our program will check if there is an app by that name, if there is then it will add it and if there is not then it prompt error.

```
239
240         if ("add" in requirements) and ("app" in requirements or "apps" in require
241             to_control_app_opening=0
242             while True:
243                 pop_1=1
244                 n_app=input("Display Name :-  ")
245                 if len(n_app.split()) == 0:
246                     print("\n No Input")
247                     print(line)
248                     if speak_mode==1:
249                         pyttsx3.speak("No Input")
250 
251                 elif n_app=="exit":
252                     break
253                 else:#bnvsd
254                     n_app_act=input("Actual Name :-  ")
255                     if len(n_app_act.split()) == 0:
256                         print("\n No Input")
257                         print(line)
258                         if speak_mode==1:
259                             pyttsx3.speak("No Input")
260 
261                     elif n_app_act=="exit":
262                         break
263 
264                     if os.system('start ' +n_app_act) == 1:
265                         print("App does not exist.")
266                         print(line+"\n")
267                         pop_1=0
268 
```

```
239
240     if ("add" in requirements) and ("app" in requirements or "apps" in require
241         to_control_app_opening=0
242         while True:
243             pop_1=1
244             n_app=input("Display Name :-  ")
245             if len(n_app.split()) == 0:
246                 print("\n No Input")
247                 print(line)
248                 if speak_mode==1:
249                     pyttsx3.speak("No Input")
```

➔ If input meets add app requirements then it will ask for "**Display Name**", you can see it assigns to var. "**n_app**" in line "**244**", the input in that var will be shown or displayed on app list with "**Display Name**" you given it can be anything exclusive "**exit**".

```
251             elif n_app=="exit":
252                 break
253             else:
254                 n_app_act=input("Actual Name :-  ")
255                 if len(n_app_act.split()) == 0:
256                     print("\n No Input")
257                     print(line)
258                     if speak_mode==1:
259                         pyttsx3.speak("No Input")
260
261                 elif n_app_act=="exit":
262                     break
```

➔ If input is "**exit**" for "**Display Name**" then you will be out of the loop means you exited.

➔ If input is not "**exit**" then it will register input as display name and ask for "**Actual Name**" which is the original app name which is registered in your OS.

➔ If want to exit give input "**exit**" or "**0**" you will be exited

```
263
264     if os.system('start ' +n_app_act) == 1:
265         print("App does not exist.")
266         print(line+"\n")
267         pop_1=0
268
269
270     if pop_1!=0:
271         print("\nTrying to opening app .....")
272         wero()
273         print("-------- app was opened for confirmation..... \n"
274
```

➔ It gonna try to open app with that "**Actual Name**" if it fail to open a prompt wi occurs as we seen at starting and if gets success it gonna close that app.

➔ The function "**wero()**" used in line "**272**" is used to quit application and afterwards it will prompt confirmation message.

```
25
26      def wero():
27          os.system("taskkill /im "+n_app_act+".exe /t /f")
28
```

```
Display Name :-  Google Chrome

Actual Name :-  chrome

Trying to opening app .....
-------- app was opened for confirmation......
Confirmation done...
 ____________________________________________________________


Nick Name/exit->0 :-
```

- ➔ Well let's try again we have a “Display Name” is "Google Chrome" and the “Actual Name” which is registered in OS is "chrome"
- ➔ The program was able to check and now determined that the app exist.
- ➔ Now it's asking for “Nick Name”, it can’t contain “0” because its restricted if you included “0” in “Nick Name” then you will be exited.
- ➔ Nickname is also referred as an alternate of the actual name you can put your comment or any key word when that key word is in your sentence the program will figure out what you want

```
276     nic_name_app=[]
277     n_app_nic=""
278     while True:
279         n_app_nic=input("Nick Name/exit->0 :-  ")
280         n_app_nic=n_app_nic.lower()
281         if len(n_app_nic.split()) == 0:
282             print("\n No Input")
283             print(line)
284             if speak_mode==1:
285                 pyttsx3.speak("No Input")
286         elif n_app_nic=="exit" or "0" in n_app_nic:
287             break
```

```
292
293         app_list_n.append(n_app_act)
294         app_list_n.append(n_app)
295
296         for i in range(len(nic_name_app)):
297             app_list_n.append(nic_name_app[i])
298
299         database.add_app_sql(app_list_n)
300         to_control_understanding=0
301         print("\n"+line+"\n")
302         print(n_app+" Added successfully !!")
303         show_apps()
304         if speak_mode==1:
305             pyttsx3.speak(n_app+" Added successfully")
306         break
```

➔ After we exited from "**Nick Name**", the code from line "**293 - 297**" is responsible to create a final list of name "**app_list_n**" containing all information regarding app.

➔ After it in line "**299**" the data is updated to database.

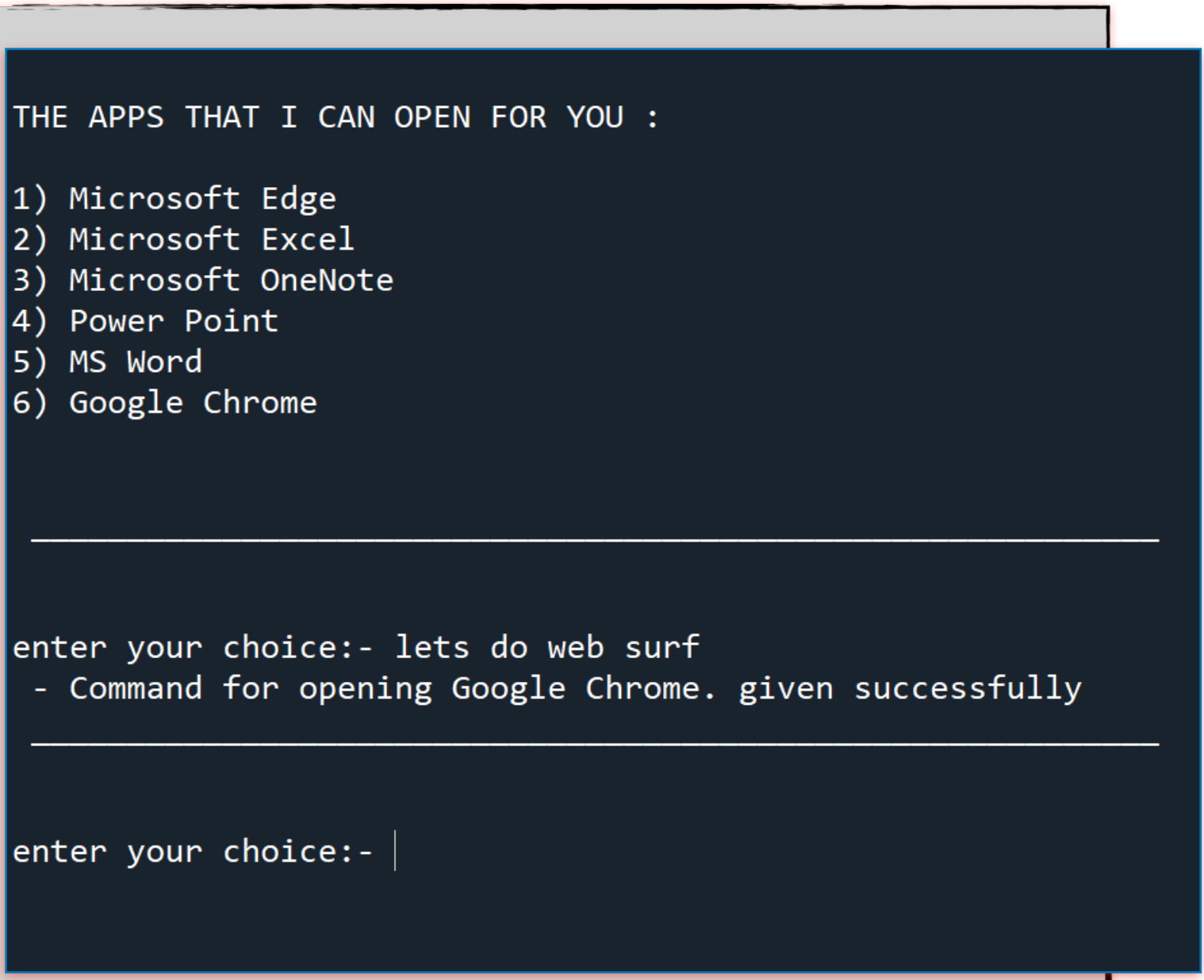
➔ Once it is updated it will prompt message of successful addition of that app.

```
73
74  def add_app_sql(tmp_sql_list):
75      cursor.execute("insert into "+mac_ad+" values('"+lts(tmp_sql_list)+"')")
76      mydb.commit()
77
```

```
11  import uuid
12  mac_ad="a"+str((hex(uuid.getnode())))
13
```

➔ Google Chrome is added successfully and now it's showing Google Chrome which is a display name in the app list.

```
Google Chrome Added successfully !!



THE APPS THAT I CAN OPEN FOR YOU :

1) Microsoft Edge
2) Microsoft Excel
3) Microsoft OneNote
4) Power Point
5) MS Word
6) Google Chrome


 ____________________________________________________________


enter your choice:-
```



➔ We also added one feature to quickly open a web browser, if you input “Web Surf” it will open it

➔ Let's add one more app, MS OneNote, here we completed all required credentials and OneNote is successfully added.

```
enter your choice:- lets add an app

Display Name :-  OneNote

Actual Name :-  onenote

Trying to opening app .....
-------- app was opened for confirmation......
Confirmation done...
 ____________________________________________________________


Nick Name/exit->0 :-  microsoft one note

Nick Name/exit->0 :-  one note

Nick Name/exit->0 :-  onenote

Nick Name/exit->0 :-  raju

Nick Name/exit->0 :-  ramu

Nick Name/exit->0 :-  0

 ____________________________________________________________

OneNote Added successfully !!
```

```
THE APPS THAT I CAN OPEN FOR YOU :

1) Microsoft Edge
2) Microsoft Excel
3) Microsoft OneNote
4) Power Point
5) MS Word
6) Google Chrome
7) OneNote


 ____________________________________________________________


enter your choice:- hey raju
 - Command for opening OneNote. given successfully
 ____________________________________________________________


enter your choice:- one note
 - Command for opening OneNote. given successfully
 ____________________________________________________________


enter your choice:-
```

➔ We are now able to open it via its Nickname "Raju" and as well as its actual name.

```
THE APPS THAT I CAN OPEN FOR YOU :

1) Microsoft Edge
2) Microsoft Excel
3) Microsoft OneNote
4) Power Point
5) MS Word
6) Google Chrome
7) OneNote
```

➔Let's see the updated database of the program

```
enter your choice:- goodbye bro
MySQL connection is closed

  - Good bye!!  ,Hope we will meet soon

_____________________________________________________


In [2]:
```

➔To exit the program Say "Goodbye"

```
                  --------------------------------

                   what can I help you with ?

                  --------------------------------

you are operating this program on windows.....



THE APPS THAT I CAN OPEN FOR YOU :

1) Microsoft Edge
2) Microsoft Excel
3) Microsoft OneNote
4) Power Point
5) MS Word
6) Google Chrome
7) OneNote


 ____________________________________________________________


enter your choice:-
```

➔We again executed the program and still, we can see the updated list.

```
enter your choice:- bye
MySQL connection is closed

  - Good bye!!  ,Hope we will meet soon

 ____________________________________________________________


In [3]:
```

➔Once again let's say bye to exit our program

```
108     if "exit" in requirements or "exit()" in requirements
109         database.close_s()
110         print("\n "+exit_speach+" \n \n"+line+"\n \n")
111         if speak_mode==1:
112             pyttsx3.speak(exit_speach)
113         break
```

```
63  def close_s():
64      if mydb.is_connected():
65          cursor.close()
66          mydb.close()
67          print("MySQL connection is closed")
68
```

```
11  import uuid
12  mac_ad="a"+str((hex(uuid.getnode())))
13
```

➔We used “**uuid**” library to extract Mac Address of the machine and added an “**a**” at starting to make a unique table for each different user.

```
43  cursor.execute("show tables")
44  x=cursor.fetchall()
45  tb_l=[]
46  for i in x:
47      tb_l.append(i[0])
48  if mac_ad in tb_l:
49      pass
50  else:
51      t_s="create table if not exists "+ mac_ad +" (app_data text)"
52      cursor.execute(t_s)
53      for i in range(1,len(apps_list_win)):
54          cursor.execute("insert into "+mac_ad+" values('"+lts(apps_list_win[i])+"')")
55          mydb.commit()
```

CODE

- database.py
- py-project-cs.py

database.py

```
print('''
==========================================

Creater:
        1) Sarvesh Dakhore
        2) Devank Yadav

==========================================

 ''')

import uuid
mac_ad="a"+str((hex(uuid.getnode())))

import mysql.connector
from mysql.connector import Error

apps_list_win=[[""],["MsEdge", "Microsoft Edge","edge","edge
browser","micrisoft browser"],["chrome","Google
chrome","browser","net surfing in google ","net surfing"],
["Excel","Microsoft Excel","excel","microsoft excel","data entry"],
["onenote","Microsoft OneNote","one note","office","ms office"],
```

```
["powerpnt","Power Point","powerpoint","power
point","Presentation","ppt presentation"],["winword","MS
Word","word","writing pad"]]


while True:
    pas=str(input("enter password of mysql localhost server: "))
    try:
        mydb =
mysql.connector.connect(host="localhost",user="root",passwd=pas,ch
arset = 'utf8')
        pop=0
    except Error:
        print("wrong password... Try again \n
-----------------------------------")
        pop=1
    if pop==0:
        break


#-----------------------------------------------------------------
def lts(lst):
    return (str('0'.join(lst)))


def stl(string):
    return (list(string.split('0')))
#-----------------------------------------------------------------


cursor=mydb.cursor()
```

```
cursor.execute("CREATE DATABASE if not exists
usr_data_1464657451")
cursor.execute("USE usr_data_1464657451")
cursor.execute("show tables")
x=cursor.fetchall()
tb_l=[]
for i in x:
    tb_l.append(i[0])
if mac_ad in tb_l:
    pass
else:
    t_s="create table if not exists "+ mac_ad +" (app_data text)"
    cursor.execute(t_s)
    for i in range(1,len(apps_list_win)):
        cursor.execute("insert into " + mac_ad + " values('"+
lts(apps_list_win[i])  +"')")
        mydb.commit()


def retrive_app_data():
    cursor.execute("select*from "+mac_ad)
    app_d=list(cursor.fetchall())
    app_list_win=[[""]]
    for i in range(0,len(app_d)):
        app_list_win.append(stl(app_d[i][0]))
    return (app_list_win)
```

```
def close_s():
    if mydb.is_connected():
        cursor.close()
        mydb.close()
        print("MySQL connection is closed")


def remove_app_sql(s_no):
    apps_list_win=retrive_app_data()
    cursor.execute("delete from "+ mac_ad +" where app_data='"+
lts(apps_list_win[s_no]) +"'")
    mydb.commit()


def add_app_sql(tmp_sql_list):
    cursor.execute("insert into " + mac_ad + " values('"+
lts(tmp_sql_list) +"')")
    mydb.commit()
```

py-project-cs.py

```
import os
import platform
import pyttsx3
import speech_recognition as sr
import database

line = "
_____
___ "
speak_mode = 0
negative_comment_speach = "It's a negative comment so i can't
proceed."
exit_speach = " - Good bye!!  ,Hope we will meet soon"
negative_comment = ["dont",'not to','not',"don't",'never do this', 'never
do' , "never" , "turn off","stop"]
positive_comment= ["do","now","start","turn on","turnon"]
voice_off=["voice","voice control"]
speak_turn_off= "speak speaking"
speak_turn_off= speak_turn_off.split()

def nif():
```

```
    print("\n - No input found")
    if speak_mode==1:
            pyttsx3.speak("No input found.")

if platform.system() == "Windows":
    print("you are operating this program on windows.....")
    x=database.retrive_app_data()

    def speech_or_text(text_to_process):
        print(text_to_process)
        if speak_mode==1:
            pyttsx3.speak(str(text_to_process))

    def wero():
        os.system("taskkill /im "+n_app_act+".exe /t /f")

    def show_apps():
        print("\n \n")
        print("THE APPS THAT I CAN OPEN FOR YOU : \n")
        x=database.retrive_app_data()
        for i in range(1,len(x)):
            q=str(i)
            print(q+") "+x[i][1])
        print("\n")
```

```
    def open_app():
        os.system('start ' +app_name)
else:
    exit



to_clear_screen=[]
app_name=""
print("\n\t\t\t\t------------------------------ \n")
print("\t\t\t\t what can I help you with ?")
print("\n\t\t\t\t------------------------------ \n")

voice_control=0
pyttsx3.speak("How Can I help You with .")
show_apps()




while True:                                                    # Start
from here
    opo=1
    to_clear_screen.append("1")
    requirements=""
```

```
to_control_switch=1
to_control_app_opening=1
to_control_understanding=1


if len(to_clear_screen)>20:
    show_apps()
    to_clear_screen.clear()

    if platform.system()=="Windows":
        os.system("cls")

app_list_n=[]

to_control=1
to_control_b=1


print(line +" \n ")

if voice_control==1:
    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("Say what you want? \n")
        r.adjust_for_ambient_noise(source)
```

```
        audio = r.listen(source,phrase_time_limit=3)
        try:
            requirements = r.recognize_google(audio)
        except sr.UnknownValueError or LookupError:
                print("No Input")
                requirements=""
        print(requirements)
        print("\n Speech done.....")
        print(line+"\n \n")




    else:
        requirements = input("enter your choice:- ")
    req_2=requirements
    requirements=requirements.lower()
    req_1 = requirements



    if "exit" in requirements or "exit()" in requirements or "good bye"
in requirements or "bye" in requirements:
        database.close_s()
        print("\n "+exit_speach+" \n \n"+line+"\n \n")
```

```
    if speak_mode==1:
        pyttsx3.speak(exit_speach)
    break


row = 0        #row


sentece_postm = requirements.split()


      # IN SEARCH OF NEGATIVE COMMENT
for h in negative_comment:
    if h in requirements:
        to_control_switch=0


if voice_control==1 and to_control_switch==0:
    for h in voice_off:
        if h in requirements:
            if voice_control==0:
                break
            else:
                for k in negative_comment:
                    if k in requirements:
```

```
                    voice_control=0
                    to_control=0
                    print("\n - Voice control turned off successfully")
                    pyttsx3.speak("Voice control turned off
successfully")
                    break
            elif voice_control==0:
                break

    elif voice_control==0 and to_control_switch==1:
        for h in voice_off:
            if h in requirements:
                if voice_control==1:
                    break
                else:
                    for k in positive_comment:
                        if k in requirements:
                            voice_control=1
                            to_control=0
                            print("\n - Voice control turned on successfully")
                            pyttsx3.speak("Voice control turned on
successfully")
                            break
            elif voice_control==1:
                break
```

```
if speak_mode==1 and to_control_switch==0:
    for h in speak_turn_off:
        if h in requirements:
            if speak_mode==0:
                break
            else:
                for k in negative_comment:
                    if k in requirements:
                        speak_mode=0
                        to_control=0
                        print("\n - speaking mode turned off successfully")
                        break
        elif speak_mode==0:
            break

elif speak_mode==0 and to_control_switch==1:
    for h in speak_turn_off:
        if h in requirements:
            if speak_mode==1:
                break
            else:
                for k in positive_comment:
```

```
                if k in requirements:
                    speak_mode=1
                    to_control=0
                    print("\n - speaking mode turned on successfully")
                    pyttsx3.speak("speaking mode turned on
successfully")
                    break
        elif speak_mode==1:
            break

    if to_control==1:
            # IN SEARCH OF NEGATIVE COMMENT
        for h in sentece_postm:
            if (h in negative_comment) and h not in speak_turn_off :
                speech_or_text(negative_comment_speach)
                to_control=0
                break
            if (h in negative_comment) and (h in speak_turn_off):
                break


    if ("remove" in requirements or "delete" in requirements) and
("app" in requirements or "apps" in requirements) and to_control==1:
        to_control_app_opening=0
        show_apps()
```

```
t=len(x)+1
print(line)
while True:
    rm_app=""
    print()
    try:
        rm_app=int(input("Enter number/to exit '0' :- "))
    except ValueError:
        pass
    if rm_app==0:
        speech_or_text("Exited!!")


    if rm_app in range(1,t):
        database.remove_app_sql(rm_app)
        show_apps()
        to_control_app_opening=0
        print("\n"+line)
        speech_or_text(x[rm_app][1]+" Removed successfully !!")
        break
    else:
        speech_or_text("\n Wrong Input!!")
        print(line)
```

```
    if ("add" in requirements) and ("app" in requirements or "apps" in
requirements) and (to_control==1):
        to_control_app_opening=0
        while True:
            pop_1=1
            n_app=input("Display Name :-  ")
            if len(n_app.split()) == 0:
                speech_or_text("\n No Input")
                print(line)

            elif n_app=="exit":
                break
            else:
                n_app_act=input("Actual Name :-  ")
                if len(n_app_act.split()) == 0:
                    speech_or_text("\n No Input")
                    print(line)

                elif n_app_act=="exit":
                    break

                if os.system('start ' +n_app_act) == 1:
                    print("App does not exist.")
                    print(line+"\n")
                    pop_1=0
```

```
            if pop_1!=0:
                print("\nTrying to opening app .....")
                wero()
                print("-------- app was opened for confirmation.....
\n"+"Confirmation done... \n"+line+"\n")



                nic_name_app=[]
                n_app_nic=""
                while True:
                    n_app_nic=input("Nick Name/exit->0 :-  ")
                    n_app_nic=n_app_nic.lower()
                    if len(n_app_nic.split()) == 0:
                        speech_or_text("\n No Input")
                        print(line)

                    elif n_app_nic=="exit" or "0" in n_app_nic:
                        break
                    else:
                        nic_name_app.append(n_app_nic)

                app_list_n.append(n_app_act)
                app_list_n.append(n_app)
```

```
            for i in range(len(nic_name_app)):
                app_list_n.append(nic_name_app[i])

            database.add_app_sql(app_list_n)
            to_control_understanding=0
            print("\n"+line+"\n")
            speech_or_text(n_app+" Added successfully !!")
            show_apps()
            break


    if "show" in requirements and "app" in requirements and
to_control==1:
        show_apps()

    elif to_control==1 and to_control_app_opening==1:
        x=database.retrive_app_data()
        for h in sentece_postm:
            if (h in negative_comment) and h not in speak_turn_off :
                speech_or_text(negative_comment_speach)
                break
            if (h in negative_comment) and (h in speak_turn_off):
                break
```

```
    if to_control==1 and to_control_understanding==1:
        if req_2.isnumeric():
            req_2=int(req_2)
            if req_2 in range(0,len(x)):
                row=req_2

        else:
            for r in range(0,len(x)):
                for d in x[r]:
                    if d in req_1:
                        row=r
                        break

        if row ==0 and to_control_understanding==1:
            speech_or_text("Unable to understand")

        else:
            app_name=x[row][0]
            if platform.system()=="Windows":
                if os.system('start ' +app_name) == 1:
                    opo=0
                    print("App does not exist. \n It has been removed as
its not existed \n")
                    print(line+"\n")
                    database.remove_app_sql(row)
```

```
                if "quit" in sentece_postm:
                    to_opo=0
                    wer=app_name
                    os.system("taskkill /im "+wer+".exe /f")


                else:
                    app_name1=x[row][1]
                    speech_or_text(" - Command for opening
"+app_name1+". given successfully")
```

THE END

DEVELOPERS

- **Sarvesh Dakhore**
- **Devank Yadav**