



NATIONAL PUBLIC SCHOOL **BANASHANKARI**

Computer Science Project
Covid Quarantine Assistance Program



Shaunak Saxena

Class: 12- B

Roll No: 19

Acknowledgement

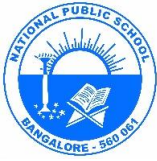
“Computer Science is a science of abstraction -creating the right model for a problem and devising the appropriate mechanizable techniques to solve it.” — Alfred Aho.

Firstly, I would like to express my sincere gratitude to our respected principal ma'am, **Mrs. Jayanti Nair** and our Academic director **Ms Grace C.D.** for giving us the opportunity work on this project.

I would also like to extend my gratitude to our computer science teacher, **Ms. Shivshakti ma'am** for her constant guidance and support. Her enthusiastic interest and encouragement has helped me complete this project successfully.

I would like to thank my parents for their continual support throughout this journey.

Finally, I would also like to express my sincere gratitude to my co-creators, **Sai Eeshwar.D** and **Siddhartha Reddy Kayitha** for their constant support and dedication towards our mutual goal.



NATIONAL PUBLIC SCHOOL BANASHANKARI

CERTIFICATE

This is to certify that **Shaunak Saxena** of Grade XII has successfully completed the Computer Science project titled “COVID Quarantine Assistance Program ” as prescribed by CBSE for the year 2020-21.

Date: _____

Signature of Teacher: _____

Signature of Principal: _____

AISSCE Code No: _____

Signature of Examiner: _____

Contents

Introduction	5
System Requirements	7
Flow Chart	8
Functions List	9
Source Code	11
Sample Outputs	35
Constraints and Scope	39
Bibliography	40

Introduction

The arrival of the corona virus pandemic was unprecedented, unparalleled and a rather complicated scenario. It affected the lives of millions in India and around the world. There was great skepticism about the situation when people first learned about the virus. With countries enforcing lockdowns and the number of cases increasing rapidly, the entire world seemed to be in turmoil. The group of people affected most is the elderly. Due to weaker immune systems, senior citizens are more likely to be infected with the virus. They account for around 80% of all COVID-19 deaths in the US.

Moreover, many older people who live alone face a greater level of difficulty as it is a challenge for them to take precautions, identify the symptoms and act accordingly. Those with physical disabilities may not be able to reach the closest hospital for a check-up. Even if they manage to reach one, the havoc of most Indian hospitals will not be the most comfortable setting for them.

Covid Quarantine Assistance program has been created to provide aid to senior citizens and other individuals amidst the pandemic. The program will help those who have minor symptoms and wish to determine if they're infected but can't approach hospitals to get professionally checked, or perhaps aren't knowledgeable about the symptoms. Although the program is not capable of providing a definitive result of presence of the virus as a professional COVID test does, it will help users to identify if they're likely infected with COVID-19.

The prompt runs on the basis of simple health related statistics, which it uses to provide a health score and a report suggesting the likeliness of being infected. The program will also contact the respective COVID-19 authorities to provide further assistance if the

user is highly likely to be infected, in addition to sending emergency messages to the user's primary contacts.

This automation incorporated by the program ensures that any individual likely to be affected is provided with professional help as soon as possible. The health report provided to authorities ensures that they are aware of the seriousness of the patient.

The government has created a robust health response and is working hard to ramp up its testing and flatten the curve. We play a huge role in tackling this pandemic by being well informed about the COVID-19 virus, its symptoms and how it spreads. Taking proper precautions such as wearing masks, applying sanitizers frequently and practicing social distancing makes a huge difference and the Covid Quarantine Assistance program was created to play its role in this change.

System and Python Requirements

The following is a comprehensive list of the minimum system requirements to be able to execute this project successfully--

- Windows or MacOS operating system
- Python version<3.9
- Mysql
- Source Code
- Internet connection
- Hard disc space - at least 2MB
- Computer peripherals such as keyboard and mouse

The following libraries must be installed to python (if not pre installed) through pip install command--

- tkinter
- mysql python connector
- smtplib
- csv
- pillow

Flow Chart



Functions List

Function Name	Parent Function	Description
select0()		Takes user login page
select1()		Takes user to register page
select2()		Takes user to health stats page and prints day number
select3()		Takes user to day 3 negative report page
select4()		Takes user to day 5 negative report page
select5()		Takes user to day 5 positive report page
day_no_check()	select2()	Checks number of days of program already completed
login()		Checks if login info is correct
username_check()	login()	Checks if username entered exists in database
variables()	details()	Accepts values on registration page
validate()	details()	Validates the info on registration page
exists_id()	validate()	Checks if user id is already existing in the database
exists_email()	validate()	Checks if the entered registration email already exists in database
exists_phone()	validate()	Checks if the user phone number is already existing in the database

check(email)	validate()	Checks if the email entered is valid
details()		Accepts registration data and validates it
save()		Validates health stats values and saves them into a csv file. Calculates health score and sends email accordingly.
healthvalid()	Save()	Validates the health stats entered
clicked()	Save()	Returns message box after saving health stats

******A function with a parent function here is not necessarily a nested function but is called only through the parent function

Source Code

```
1. import tkinter as tk
2. from tkinter import *
3. from tkinter import ttk
4. from PIL import Image,ImageTk #PIL -> Pillow
5. from tkinter import messagebox
6. import random
7. import mysql.connector
8. import csv
9. import smtplib
10. from email.mime.multipart import MIMEMultipart
11. from email.mime.text import MIMEText
12. from email.mime.base import MIMEBase
13. from email import encoders
14. import os.path
15. from os import path
16.
17. #####
18.
19. mydb = mysql.connector.connect(
20.     host="localhost",
21.     user="root",
22.     passwd="user",
23.     database="cqap"
24. )
25. mycursor = mydb.cursor()
26. #####
27.
28. ##tab1
29. root = Tk()
30. root.title("CQA-1")
31. root.configure(bg="white")
32. root.geometry("2000x1000")
33. tc=ttk.Notebook(root)
34. tc.pack(pady=0,padx=0)
35. page0 = Frame(tc,width=1280,height=720)
36. page1 = Frame(tc,width=1280,height=720 )
37. page2 = Frame(tc,width=1280,height=720 )
38. page3 = Frame(tc,width=1280,height=720 )
39. page4 = Frame(tc,width=1280,height=720 )
40. page5 = Frame(tc,width=1280,height=720 )
41. page0.pack(expand = 1,fill="both")
42. page1.pack(expand = 1,fill="both")
43. page2.pack(expand = 1,fill="both")
44. page3.pack(expand = 1,fill="both")
45. page5.pack(expand = 1,fill="both")
46. tc.add(page0 , text='WELCOME')
47. tc.add(page1 , text='REGISTER')
48. tc.add(page2 , text='GENERAL')
49. tc.add(page3 , text='RESULT-DAY3')
```

```

50. tc.add(page4 , text='RESULT-DAYS')
51. tc.add(page5 , text='RESULT-DAYS')
52. tc.hide(1)
53. tc.hide(2)
54. tc.hide(3)
55. tc.hide(4)
56. tc.hide(5)
57. canvas = tk.Canvas(page0, width=1280, height=720)
58. canvas.grid()
59. back=Image.open("welcome.jpeg")
60. resize=back.resize((800,690),Image.ANTIALIAS)
61. new=ImageTk.PhotoImage(resize)
62. canvas.create_image(400,360,image=new)
63. back1=Image.open("logo.png")
64. resize=back1.resize((400,450),Image.ANTIALIAS)
65. new1=ImageTk.PhotoImage(resize)
66. canvas.create_image(1100,160,image=new1)
67. noaccount=tk.Label(page0,fg="black",text="OR",bg="white",font="CaviarDreams 20
    bold ").place(x=1080,y=550)
68. user=tk.Label(page0,fg="black",text=" User ID :",font="CaviarDreams 10 bold ").
    place(x=860,y=400)
69. user1_e=tk.Entry(page0,width=45,font="CaviarDreams 10 bold ",bg="white",borderw
    idth=3)
70. user1_e.place(x=950,y=400)
71. pwd=tk.Label(page0,fg="black",text=" Enter Password: ",font="CaviarDreams 10
    bold ").place(x=810,y=440)
72. pwd1_e=tk.Entry(page0,width=45,font="CaviarDreams 10 bold ",bg="white",borderwi
    dth=3,show="*")
73. pwd1_e.place(x=950,y=440)
74.
75.
76. #####
    #####
77.
78. global t1
79. t1=user1_e.get()
80. def select1():
81.     tc.select(1)
82.     tc.hide(0)
83. def select2():
84.     tc.select(2)
85.     tc.hide(1)
86.     tc.hide(0)
87.     def day_no_check():
88.         global day_count
89.         if path.exists(filename)==False:
90.             day_count='Welcome to day 1'
91.         else:
92.             global pre_stats_rows
93.             pre_stats_rows=[]
94.             with open(filename, 'r') as file:
95.                 reader = csv.reader(file)
96.                 for row in reader:
97.                     pre_stats_rows.append(row)
98.             if len(pre_stats_rows)==4:
99.                 day_count='Welcome to day 2'
100.             elif len(pre_stats_rows)==8:

```

```

101.         day_count='Welcome to day 3'
102.     elif len(pre_stats_rows)==12:
103.         day_count='Welcome to day 4'
104.     elif len(pre_stats_rows)==16:
105.         day_count='Welcome to day 5'
106.     else:
107.         day_count= 'Day Count Limit Reached'
108.
109.         gnqu=tk.Label(page2,fg="red",text=day_count,bg="white",font=
"Georgia 15 bold italic").place(x=560,y=15)
110.         day_no_check()
111.
112.
113.     ##login
114.     def login():
115.         global t1
116.         t1=user1_e.get()
117.         global t2
118.         t2=pwd1_e.get()
119.
120.         global filename
121.         filename = t1+"_"+"daily_stats.csv"
122.
123.         if t1==" " or t2==" ":
124.             messagebox.showinfo('Error','Please fill all fields')
125.             return
126.
127.         def username_check():
128.             mycursor.execute("select username from users")
129.             tab=mycursor.fetchall()
130.             u=[]
131.             for i in tab:
132.                 u.append(i[0])
133.             for z in u:
134.                 if z==t1:
135.                     return True
136.             else:
137.                 messagebox.showinfo('Error',"Incorrect UserID")
138.
139.         pwd_check=True
140.         if username_check()==True:
141.             mycursor.execute("select pwd from users where username = ' " + t
1 + " ' ") ##gives form 't1'
142.             tab1=mycursor.fetchall()
143.             for j in tab1:
144.                 if j[0]==t2:
145.                     select2()
146.                 else:
147.                     pwd_check=False
148.
149.             if pwd_check==False:
150.                 messagebox.showinfo('Error',"Incorrect password")
151.
152.
153.     ### button from login to question
154.     ##need to add if condition to check if user id matches database
155.     weltoque_but1=Image.open("login.png")

```

```

156.         resize=weltoque_but1.resize((120,50),Image.ANTIALIAS)
157.         ne47=ImageTk.PhotoImage(resize)
158.
159.         welqnext_but1=tk.Button(page0,bg="white",image=ne47,command=login,border
width=0)
160.         welqnext_but1.place(x=1050,y=474)
161.
162.
163.         #own create button
164.         myimgnext_but1=Image.open("register.png")
165.         resize=myimgnext_but1.resize((120,50),Image.ANTIALIAS)
166.         ne44=ImageTk.PhotoImage(resize)
167.
168.         next_but1=tk.Button(page0,bg="white",image=ne44,command=select1,borderwi
dth=0)
169.         next_but1.place(x=1050,y=600)
170.
171.         #####
#####
172.
173.         def variables():
174.             global n1,n2,n3,n4,n5,n6,n7,n8,n9,n10,n11,n12,n13,n14,n15,n16,n17,n1
8,n19,n20,lpg1,lpg2
175.             n1=name_e.get()
176.             n2=username_e.get()
177.             n3=pwd_e.get()
178.             n4=cpwd_e.get()
179.             n5=varg.get()
180.             n6=ph_e.get()
181.             n7=email_e.get()
182.             n8=age_e.get()
183.             n9=address_e.get('1.0',END)
184.             n10=pr1_e.get()
185.             n11=pr1e_e.get()
186.             n12=pr2_e.get()
187.             n13=pr2e_e.get()
188.             n14=pr3_e.get()
189.             n15=pr3e_e.get()
190.             n16=md_e.get()
191.             n17=click.get()
192.             n18=allr_e.get()
193.             n19=var_t.get()
194.             n20=sptr_e.get()
195.             lpg1=[n1,n2,n3,n4,n5,n6,n7,n8,n9,n10,n11,n12,n13,n14,n15,n16,n17,n18
,n19,n20]
196.             lpg2=[n1,n2,n3,n4,n5,n6,n7,n8,n9,n10,n11,n12,n13,n14,n15,n17,n19]
197.
198.
199.
200.         def exists_id():
201.             mycursor.execute("select username from users")
202.             tab=mycursor.fetchall()
203.             u=[]
204.             for i in tab:
205.                 u.append(i[0])
206.             if n2 in u:
207.                 return True

```

```

208.         else:
209.             return False
210.
211.
212.     def exists_email():
213.         global n7
214.         mycursor.execute("select email from users")
215.         tab=mycursor.fetchall()
216.         u=[]
217.         for i in tab:
218.             u.append(i[0])
219.         if n7 in u:
220.             return True
221.         else:
222.             return False
223.
224.
225.
226.     def exists_phone():
227.         mycursor.execute("select contact_no from users")
228.         tab=mycursor.fetchall()
229.         u=[]
230.         for i in tab:
231.             u.append(i[0])
232.         for z in u:
233.             if z==n6:
234.                 return True
235.             else:
236.                 return False
237.
238.     def check(email):
239.         import re
240.         regex = '^([a-z0-9]+[.]?)[a-z0-9]+[@]\w+[.]\w{2,3}$'
241.         if(re.search(regex,email)):
242.             return True
243.         else:
244.             return False
245.
246.
247.     def validate():
248.         duplicatemails=False
249.         e_mails=[n7,n11,n13,n15]
250.         e_mails_set=set(e_mails)
251.
252.         if n1!="" and n2!="" and n3!="" and n4!="" and n5!="" and n6!="" and n7!="" and n8!="" and n9!="" and n10!="" and n11!="" and n12!="" and n13!="" and n14!="" and n15!="" and n17!="SELECT" and n19!="":
253.             if exists_id()==True:
254.                 messagebox.showinfo('Error-Username', "This username is already taken.")
255.             if exists_email()==True:
256.                 print(n7)
257.                 messagebox.showinfo('Error-Email', "The entered email-id already exists. Please login.")
258.             if exists_phone()==True:
259.                 messagebox.showinfo('Error-Contact Number', "This contact number is already taken.")

```

```

260.         a=n1.split()
261.         b=n10.split()
262.         c=n12.split()
263.         d=n12.split()
264.         for i in a:
265.             if i!=" " and i.isalpha()==False :
266.                 messagebox.showinfo('Error-
Name',"Enter alphabets only")
267.
268.         for i in b:
269.             if i!=" " and i.isalpha()==False :
270.                 messagebox.showinfo('Error-
Name',"Enter alphabets only")
271.
272.         for i in c:
273.             if i!=" " and i.isalpha()==False :
274.                 messagebox.showinfo('Error-
Name',"Enter alphabets only")
275.
276.         for i in d:
277.             if i!=" " and i.isalpha()==False :
278.                 messagebox.showinfo('Error-
Name',"Enter alphabets only")
279.         #if n1.isalpha()==False and n1.split()!=" " or n10.isalpha()==F
else and n10.split()!=" " or n12.isalpha()==False and n12.split()!=" " or n14.i
salpha()==False or n14.split()!=" " :
280.             #   messagebox.showinfo('Error-Name',"Enter alphabets only")
281.             if len(n3)< 8:
282.                 messagebox.showinfo('Error-
login',"Password should be greater than 8 characters")
283.             if n3!=n4:
284.                 messagebox.showinfo('Error-
password',"Passwords do not match")
285.             if len(n6)!=10 :
286.                 messagebox.showinfo('Error-
contact',"contact number should be 10 digits only")
287.             if len(n9)< 10 :
288.                 messagebox.showinfo('Error-
address',"Please enter valid address")
289.             if n6.isdigit()==False:
290.                 messagebox.showinfo('Error-
contact',"Enter numeric characters only")
291.             if check(n7)==False or check(n11)==False or check(n13)==False or
check(n15)==False:
292.                 messagebox.showinfo('Error-
Email',"Invalid email, Please enter a valid email")
293.             if n8.isdigit()==False:
294.                 messagebox.showinfo('Error-age',"Enter a valid age")
295.
296.             if len(e_mails)!=len(e_mails_set):
297.                 duplicatemails=True
298.                 messagebox.showinfo('Error-
Email',"Emails can not be repeated")
299.
300.             if n1.isalpha()==True and n10.isalpha()==True and n12.isalpha()=
=True and n14.isalpha()==True and len(n3)>=8 and n3==n4 and len(n6)==10 and n6.
isdigit()==True and check(n7)==True and check(n11)==True and check(n13)==True a

```



```

nd check(n15)==True and exists_id()==False and exists_email()==False and exists
_phone()==False and duplicatemails==False:
301.         sql = "INSERT INTO users (name,username,pwd,cpwd,gender,cont
act_no,email,age,address,pricon1_name,pricon1_email,pricon2_name,pricon2_email,
pricon3_name,pricon3_email,med_disorder,bld_grp,algy,treat_speci) VALUES (%s, %
s,%s, %s,%s, %s,%s, %s,%s, %s,%s, %s,%s, %s,%s, %s,%s,%s,%s)"
302.         val = (n1,n2,n3,n4,n5,n6,n7,n8,n9,n10,n11,n12,n13,n14,n15,n1
6,n17,n18,n20)
303.         mycursor.execute(sql, val)
304.         mydb.commit()
305.         messagebox.showinfo('Login',"account created")
306.         '''with open(filename,'a') as csvfile:
307.             csvwriter = csv.writer(csvfile) # creating a csv write
r object
308.             csvwriter.writerow([n2]) # writing the fields
309.             # csvwriter.writerow(rows) # writing the data rows
...
310.         select0()
311.     else:
312.         messagebox.showinfo('Error-
fields',"Fill all required fields (all except medical disorder, allergy, specia
l treatment)")
313.
314.     def details():
315.         variables()
316.         validate()
317.
318.
319.     def select0():
320.         tc.select(0)
321.
322.
323.         canvas = tk.Canvas(page1, width=1280, height=720)
324.         canvas.grid()
325.         f_back=Image.open("bgwhite2.jpg")
326.         resize=f_back.resize((1280,800),Image.ANTIALIAS)
327.         n=ImageTk.PhotoImage(resize)
328.         canvas.create_image(633,324, image=n)
329.
330.         register=tk.Label(page1,fg="black",text="CREATE NEW ACCOUNT",bg="white",
font="Georgia 20 bold").place(x=60,y=50)
331.
332.
333.         #name
334.         global name_e
335.         name=tk.Label(page1,fg="black",text="Name :",bg="White",font="Georgia 11
").place(x=60,y=110)
336.         name_e=tk.Entry(page1,width=25,font="Georgia 11",bg="white",borderwidth=
3)
337.         name_e.place(x=350,y=100)
338.
339.
340.         #username
341.         username=tk.Label(page1,fg="black",text="Username :",bg="White",font="Ge
orgia 11").place(x=60,y=140)
342.         username_e=tk.Entry(page1,width=25,font="Georgia 11",bg="white",borderwi
dth=3)

```

```

343.         username_e.place(x=350,y=135)
344.
345.
346.         #phone
347.         ph=tk.Label(page1,fg="black",text=" Contact number :",bg="white",font="Georgia 11").place(x=60,y=170)
348.         ph_e=tk.Entry(page1,width=25,font="Georgia 11",bg="white",borderwidth=3)
349.
350.         ph_e.place(x=350,y=165)
351.
352.         #email
353.         email=tk.Label(page1,fg="black",text=" Email ID :",bg="white",font="Georgia 11").place(x=60,y=205)
354.         email_e=tk.Entry(page1,width=25,font="Georgia 11",bg="white",borderwidth=3)
355.         email_e.place(x=350,y=205)
356.
357.         #age
358.         age=tk.Label(page1,fg="black",text=" Age:",bg="white",font="Georgia 11",).place(x=60,y=240)
359.         age_e=tk.Entry(page1,width=25,font="Georgia 11",relief="sunken",borderwidth=3)
360.         age_e.place(x=350,y=240)
361.
362.
363.         #address
364.         address=tk.Label(page1,fg="black",text=" Address:",bg="white",font="Georgia 11",).place(x=60,y=270)
365.         address_e=Text(page1,width=25,height=4,font="Georgia 11",relief="sunken",borderwidth=3)
366.         address_e.place(x=350,y=270)
367.
368.         #primary1name
369.         pr1=tk.Label(page1,fg="black",text=" Primary contact 1 name:",bg="white",font="Georgia 11",).place(x=60,y=360)
370.         pr1_e=tk.Entry(page1,width=25,font="Georgia 11",relief="sunken",borderwidth=3)
371.         pr1_e.place(x=350,y=360)
372.
373.         #primary1email
374.         pr1e=tk.Label(page1,fg="black",text=" Primary contact 1 email id:",bg="white",font="Georgia 11",).place(x=60,y=390)
375.         pr1e_e=tk.Entry(page1,width=25,font="Georgia 11",relief="sunken",borderwidth=3)
376.         pr1e_e.place(x=350,y=390)
377.
378.         #primary2
379.         pr2=tk.Label(page1,fg="black",text=" Primary contact 2 name:",bg="white",font="Georgia 11",).place(x=60,y=420)
380.         pr2_e=tk.Entry(page1,width=25,font="Georgia 11",relief="sunken",borderwidth=3)
381.         pr2_e.place(x=350,y=420)
382.
383.
384.         #primary2email

```

```

385.     pr2e=tk.Label(page1,fg="black",text=" Primary contact 2 email id:",bg="w
hite",font="Georgia 11",).place(x=60,y=450)
386.     pr2e_e=tk.Entry(page1,width=25,font="Georgia 11",relief="sunken",borderw
idth=3)
387.     pr2e_e.place(x=350,y=450)
388.
389.     #primary3
390.     pr3=tk.Label(page1,fg="black",text=" Primary contact 3 name:",bg="white"
,font="Georgia 11",).place(x=60,y=480)
391.     pr3_e=tk.Entry(page1,width=25,font="Georgia 11",relief="sunken",borderwi
dth=3)
392.     pr3_e.place(x=350,y=480)
393.     #primary3email
394.     pr3e=tk.Label(page1,fg="black",text=" Primary contact 3 email id:",bg="w
hite",font="Georgia 11",).place(x=60,y=510)
395.     pr3e_e=tk.Entry(page1,width=25,font="Georgia 11",relief="sunken",borderw
idth=3)
396.     pr3e_e.place(x=350,y=510)
397.
398.     #password
399.     pwd=tk.Label(page1,fg="black",text=" ENTER PASSWORD:",bg="white",font="G
eorgia 11").place(x=861,y=290)
400.     pwd_e=tk.Entry(page1,width=22,font="Georgia 10",bg="white",borderwidth=3
,show="*")
401.     pwd_e.place(x=1050,y=290)
402.
403.     #confirmpassword
404.     cpwd=tk.Label(page1,fg="black",text=" CONFIRM PASSWORD:",bg="white",font
="Georgia 11").place(x=840,y=330)
405.     cpwd_e=tk.Entry(page1,width=22,font="Georgia 10",bg="white",borderwidth=
3,show="*")
406.     cpwd_e.place(x=1050,y=330)
407.
408.
409.     #medical disorder
410.     md=tk.Label(page1,fg="black",text=" Medical Disorder (if any):",bg="whit
e",font="Georgia 11").place(x=60,y=615)
411.     md_e=tk.Entry(page1,width=25,font="Georgia 11",bg="white",borderwidth=3)
412.     md_e.place(x=350,y=615)
413.
414.     #bloofgroup
415.     bldgrp=tk.Label(page1,fg="black",text=" Blood Group :",bg="white",font="
Georgia 11").place(x=60,y=540)
416.     click=StringVar()
417.     names=['SELECT',' A+', ' B+', ' O+', ' AB+', ' A-', ' B-', ' O-', ' AB-']
418.     click.set(names[0])
419.     drop=OptionMenu(page1,click,*names)
420.     drop.config(width=20,bg="white",font="Georgia 11",borderwidth=3)
421.     drop.place(x=350,y=540)
422.
423.
424.     #gender-radio button
425.     gen=tk.Label(page1,fg="black",text="Gender :",bg="white",font="Georgia 1
1").place(x=952,y=100)
426.     varg=IntVar()

```

```

427.     R1 = Radiobutton(page1, text="M", variable=varg, value="1",command=varia
bles)
428.     R1.place(x=1050,y=100)
429.     R2 = Radiobutton(page1, text="F", variable=varg, value="2",command=varia
bles)
430.     R2.place(x=1120,y=100)
431.     R3 = Radiobutton(page1, text="Other", variable=varg, value="3",command=v
ariables)
432.     R3.place(x=1190,y=100)
433.
434.
435.     #allergies
436.     allr=tk.Label(page1,fg="black",text="Allergies to any specific medicatio
n? :",bg="white",font="Georgia 11").place(x=761,y=190)
437.     allr_e=tk.Entry(page1,width=22,font="Georgia 11",bg="white",borderwidth=
3)
438.     allr_e.place(x=1050,y=190)
439.
440.
441.     #travelhistory- raDIO BUTTON
442.     trvhis=tk.Label(page1,fg="black",text="Have you travelled internationall
y in last the 30 days :",bg="white",font="Georgia 11").place(x=647,y=140)
443.     vart=IntVar()
444.     R1 = Radiobutton(page1, text="Yes", variable=vart, value=1,command=varia
bles)
445.     R1.place(x=1050,y=140)
446.     R2 = Radiobutton(page1, text="No", variable=vart, value=2,command=variab
les)
447.     R2.place(x=1120,y=140)
448.
449.
450.     #specifictreatment
451.     sptr=tk.Label(page1,fg="black",text="Under any specfic treatment, if yes
specify :",bg="white",font="Georgia 11").place(x=720,y=235)
452.     sptr_e=tk.Entry(page1,width=22,font="Georgia 11",bg="white",borderwidth=
3)
453.     sptr_e.place(x=1050,y=235)
454.
455.
456.     #created account successfully thing button
457.     myimgreg=Image.open("create_acc.png")
458.     resize=myimgreg.resize((180,100),Image.ANTIALIAS)
459.     ne41=ImageTk.PhotoImage(resize)
460.
461.     reg_button=tk.Button(page1,bg="white",image=ne41,command=details,borderw
idth=0)
462.     reg_button.place(x=1050,y=620)
463.
464.
465.     acc_created=tk.Label(page1,fg="black",bg="white",font="Georgia 11")
466.     acc_created.place(x=960,y=550)
467.
468.     def select3():
469.         tc.select(3)
470.         tc.hide(2)
471.

```

```

472. #####
#####
#####
473. #tab3 general questions
474. Canvas = tk.Canvas(page2, width=1280, height=720)
475. Canvas.grid()
476. f_back=Image.open("bgwhite2.jpg")
477. resize=f_back.resize((1280,850),Image.ANTIALIAS)
478. no=ImageTk.PhotoImage(resize)
479. Canvas.create_image(633,324, image=no)
480.
481. gnqu=tk.Label(page2,fg="black",text="HEALTH QUESTIONS",bg="white",font="
Georgia 20 bold").place(x=60,y=60)
482.
483.
484. ##input questions
485. symp=tk.Label(page2,fg="black",text="Are you facing any of the following
symptoms?",bg="white",font="Georgia 20").place(x=60,y=100)
486. ####save function
487. def save():
488.     global filename
489.     filename = t1+"_"+"daily_stats.csv"
490.     avg_d3=0
491.     avg_d5=0
492.     h1=var1.get()
493.     h2=var2.get()
494.     h3=var3.get()
495.     h4=var4.get()
496.     h5=var5.get()
497.     h6=var6.get()
498.     h7=var7.get()
499.     h8=var8.get()
500.     h9=var9.get()
501.     h10=var10.get()
502.     h11=var11.get()
503.     h12=var12.get()
504.     h13=var13.get()
505.     h14=temp_e.get()
506.     h15=wt_e.get()
507.     h16=bp_e.get()
508.     h17=hr_e.get()
509. #####
#####
#####
510.
511. ##health validation
512. def healthvalid():
513.     if h14.isdigit()==False:
514.         messagebox.showinfo('Error-
Temperature', "Enter valid Temperature.")
515.     if h15.isdigit()==False:
516.         messagebox.showinfo('Error-Weight', "Enter valid Weight")
517.     if re.match('^[0-9]{2,3}/[0-9]{2,3}$', h16):
518.         h16_check=True
519.     else:
520.         h16_check=False

```

```

521.         messagebox.showinfo('Error-
Blood Pressure', "Enter valid Blood Pressure")
522.         if h17.isdigit()==False:
523.             messagebox.showinfo('Error-
Heart Rate', "Enter valid Heart Rate")
524.         if h14.isdigit()==True and h15.isdigit()==True and h17.isdigit()
==True and h16_check==True:
525.             return True
526.
527.         ##if len(ld)==4:
528.         if h14!="" and h15!="" and h16!="" and h17!="":
529.             if healthvalid()==True:
530.                 c=0
531.                 p=[]#point backend
532.                 l=[]#csv yes or no
533.                 #f=open
534.
535.                 h18=''
536.                 l.append(h18)
537.                 if h1==1:#cough
538.                     l.append("Yes")
539.                     p.append(4)
540.                 else:
541.                     l.append("No")
542.                 if h2==1:#fever
543.                     l.append("Yes")
544.                     p.append(2)
545.                 else:
546.                     l.append("No")
547.
548.                 if h3==1:#chest pain
549.                     l.append("Yes")
550.                     p.append(4)
551.                 else:
552.                     l.append("No")
553.                 if h4==1:#difficulty in breathing
554.                     l.append("Yes")
555.                     p.append(5)
556.                 else:
557.                     l.append("No")
558.                 if h5==1:#sore throat
559.                     l.append("Yes")
560.                     p.append(2)
561.                 else:
562.                     l.append("No")
563.                 if h6==1:#body pains and aches
564.                     l.append("Yes")
565.                     p.append(2)
566.                 else:
567.                     l.append("No")
568.                 if h7==1:#diarrhoea
569.                     l.append("Yes")
570.                     p.append(3)
571.                 else:
572.                     l.append("No")
573.                 if h8==1:#headache
574.                     l.append("Yes")

```

```

575.         p.append(3)
576.     else:
577.         l.append("No")
578.     #
579.     if h9==1:#loss of taste or smell
580.         l.append("Yes")
581.         p.append(3)
582.     else:
583.         l.append("No")
584.     if h10==1:#tiredness
585.         l.append("Yes")
586.         p.append(3)
587.     else:
588.         l.append("No")
589.     if h11==1:#conjunctivitis
590.         l.append("Yes")
591.         p.append(2)
592.     else:
593.         l.append("No")
594.     if h12==1:#rash or skin discoloration
595.         l.append("Yes")
596.         p.append(3)
597.     else:
598.         l.append("No")
599.     if h13==1:#loss of speech or movement
600.         l.append("Yes")
601.         p.append(5)
602.     else:
603.         l.append("No")
604.
605.
606.     #temp
607.     l.append(h14)
608.     #weight
609.     l.append(h15)
610.     #BP
611.     l.append(h16)
612.     #Heart rate
613.     l.append(h17)
614.
615.     if vart==1: #international travel
616.         p.append(6)
617.         '''l.append("Yes")
618.     else:
619.         l.append("No")'''
620.     for j in p:
621.         c+=j
622.         l.append(100-((c/47)*100))
623.     #columns
624.     fields=["Day", "Dry Cough", "Fever", "Chest Pain", "Difficult
y in breathing", "Sore throat", "Body Pains and Aches", "Diarrhoea", "Headache"
, "Loss of taste or smell", "Tiredness", "Conjunctivits", "Rash or skin discolou
ration", "Loss of speech or locomotion", "Temperature", "Weight", "Blood Pressur
e", "Heart Rate", "Health Score"]
625.     rows=[]
626.     for i in l:
627.         rows.append(i)

```

```

628.
629.
630.         # name of csv file
631.         # writing to csv file
632.
633.
634.
635.         with open(filename, 'a') as csvfile:
636.             csvwriter = csv.writer(csvfile) # creating a csv write
r object
637.             csvwriter.writerow(fields)      # writing the fields
638.             csvwriter.writerow(rows)        # writing the data rows
639.
640.         global stats_rows
641.         stats_rows=[]
642.         with open(filename, 'r') as file:
643.             reader = csv.reader(file)
644.             for row in reader:
645.                 stats_rows.append(row)
646.
647.         #checking if health is bad after day 3
648.         if len(stats_rows)==12:
649.
650.             print('d3 got')
651.
652.             #print (stats_rows)
653.             print ('following are the scores extracted and averaged
from first 3 days')
654.
655.             stats_d1=[]
656.             stats_d1=stats_rows[2]
657.             #print (stats_d1[19])
658.
659.             stats_d2=[]
660.             stats_d2=stats_rows[6]
661.             #print (stats_d2[19])
662.
663.             stats_d3=[]
664.             stats_d3=stats_rows[10]
665.             #print (stats_d3[19])
666.
667.             #global avg_d3
668.             avg_d3=((float(stats_d1[19])+float(stats_d2[19])+float(s
tats_d3[19]))/3)
669.             print (avg_d3)
670.
671.
672.             if (avg_d3<=45):
673.
674.                 stats_rows[2][0]=str(1)
675.                 stats_rows[6][0]=str(2)
676.                 stats_rows[10][0]=str(3)
677.
678.                 with open(filename, 'w') as file:
679.                     file.truncate()
680.
681.                 with open(filename, 'a+', newline='') as file:

```



```

682.             with file:
683.                 write = csv.writer(file)
684.                 write.writerows(stats_rows)
685.
686.
687.                 scoretext=tk.Label(page3,bg="white", fg="red",text="
Your Health Score is: "+ str(round(avg_d3,2))+'/100',font="TimesNewRoman 15 italic").place(x=500, y=131)
688.                 select3()
689.
690.                 b=[]
691.                 data=[]
692.                 mycursor.execute("select name, gender, contact_no, email, age, address, med_disorder, bld_grp, algy, treat_speci, pricon1_email, pricon2_email, pricon3_email from users where username='"+ t1 + "' ") ##gives form 't1'
693.                 tab2=mycursor.fetchall()
694.                 for z in tab2:
695.                     b.append(z)
696.                 for i in b:
697.                     data.append(i)
698.
699.                 #print(emails)
700.                 name=data[0]
701.                 gender=data[1]
702.                 if gender=='1':
703.                     gender='M'
704.                 elif gender=='2':
705.                     gender='F'
706.                 else:
707.                     gender='other'
708.                 contact_no=data[2]
709.                 age=data[4]
710.                 address=data[5]
711.                 med_disorder=data[6]
712.                 if med_disorder=='':
713.                     med_disorder='N/a'
714.                 bld_grp=data[7]
715.                 algy=data[8]
716.                 if algy=='':
717.                     algy='N/a'
718.                 treat_speci=data[9]
719.                 if treat_speci=='':
720.                     treat_speci='N/a'
721.                 email=[data[3], data[10], data[11], data[12]]
722.
723.
724.                 #print(email)
725.
726.                 for i in email:
727.
728.                     fromaddr = "EMAIL address of the sender"
729.                     toaddr = "EMAIL address of the receiver"
730.
731.
732.                 # instance of MIME multipart
733.                 msg = MIME multipart()

```

```

734.
735.                                     fromaddr = 'covidquarantineassistance@gmail.com'

736.                                     my_password="gocoronago"
737.                                     # storing the receivers email address
738.                                     toaddr= i
739.
740.                                     msg['Subject'] = "COVID Quarantine Assistance Pr
ogram Report"
741.                                     msg["From"]=fromaddr
742.                                     msg['To']=toaddr
743.
744.                                     #l=["Thank you for entering your data!","Please
come back tomorrow.","You may close the application."]
745.                                     #messagebox.showinfo('Covid Quarantine Assistan
ce',"").join(l))
746.                                     body = "Dear Sir/ma'am\n\nPlease find attached a
copy of Mr/Mrs "+str(name)+ "'s COVID Quarantine assistance program health sta
ts report.\n\nFollowing are the user's details:\n\nName: "+str(name)+"\nGender:
"+str(gender)+"\nContact No: "+str(contact_no)+"\nEmail: "+str(email[0])+"\nAge
: "+str(age)+"\nAddress: "+str(address)+"Medical Disorder: "+str(med_disorder)
+"\nBlood Group: "+str(bld_grp)+"\nAllergy: "+str(algy)+"\nSpecial Treatment: "
+str(treat_speci)+"\n\nThe users health has not been looking good for the past
few days. He/she should seek professional help as soon as possible\n\nThank you
for using our programme\nRegards\nCQAP team"
747.                                     msg.attach(MIMEText(body, 'plain'))
748.
749.
750.                                     # open the file to be sent
751.                                     #filename = "sardarth_daily_stats.csv.csv"
752.                                     attachment = open(filename, "rb")
753.
754.                                     # instance of MIMEBase and named as p
755.                                     p = MIMEBase('application', 'octet-stream')
756.
757.                                     # To change the payload into encoded form
758.                                     p.set_payload((attachment).read())
759.
760.                                     # encode into base64
761.                                     encoders.encode_base64(p)
762.
763.                                     p.add_header('Content-
Disposition', "attachment; filename= %s" % filename)
764.
765.                                     # attach the instance 'p' to instance 'msg'
766.                                     msg.attach(p)
767.
768.                                     s = smtplib.SMTP_SSL('smtp.gmail.com')
769.                                     s.login(fromaddr,my_password)
770.                                     s.sendmail(fromaddr,toaddr, msg.as_string())
771.                                     s.quit()
772.
773.
774.                                     #checking if health is bad after day 5
775.                                     if len(stats_rows)==20:
776.
777.                                     #print (stats_rows)

```

```

778.         print ('following are the scores extracted and averaged
from first 5 days')
779.
780.         stats_d1=[]
781.         stats_d1=stats_rows[2]
782.         #print (stats_d1[19])
783.
784.         stats_d2=[]
785.         stats_d2=stats_rows[6]
786.         #print (stats_d2[19])
787.
788.         stats_d3=[]
789.         stats_d3=stats_rows[10]
790.         #print (stats_d3[19])
791.
792.
793.         stats_d4=[]
794.         stats_d4=stats_rows[14]
795.
796.         stats_d5=[]
797.         stats_d5=stats_rows[18]
798.
799.
800.         avg_d5=(float(stats_d1[19])+float(stats_d2[19])+float(st
ats_d3[19])+float(stats_d4[19])+float(stats_d5[19]))/5
801.         print (avg_d5)
802.
803.         if (avg_d5<=45):
804.
805.             stats_rows[2][0]=str(1)
806.             stats_rows[6][0]=str(2)
807.             stats_rows[10][0]=str(3)
808.             stats_rows[14][0]=str(4)
809.             stats_rows[18][0]=str(5)
810.
811.             with open(filename, 'w') as file:
812.                 file.truncate()
813.
814.             with open(filename, 'a+', newline='') as file:
815.                 with file:
816.                     write = csv.writer(file)
817.                     write.writerows(stats_rows)
818.
819.
820.             scoretext=tk.Label(page4,bg="white", fg="red",text="
Your Health Score is: "+ str(round(avg_d5,2))+'/100',font="TimesNewRoman 15 ita
lic").place(x=500, y=131)
821.             select4() ## put button there, to view avg score aft
er 5 days
822.
823.             b=[]
824.             data=[]
825.             mycursor.execute("select name, gender, contact_no, e
mail, age, address, med_disorder, bld_grp, algy, treat_speci, pricon1_email, pr
icon2_email, pricon3_email from users where username='"+ t1 + "' ") ##gives fo
rm 't1'
826.             tab2=mycursor.fetchall()

```

```

827.         for z in tab2:
828.             b.append(z)
829.         for i in z:
830.             data.append(i)
831.
832.         #print(emails)
833.         name=data[0]
834.         gender=data[1]
835.         if gender=='1':
836.             gender='M'
837.         elif gender=='2':
838.             gender='F'
839.         else:
840.             gender='other'
841.         contact_no=data[2]
842.         age=data[4]
843.         address=data[5]
844.         med_disorder=data[6]
845.         if med_disorder=='':
846.             med_disorder='N/a'
847.         bld_grp=data[7]
848.         algy=data[8]
849.         if algy=='':
850.             algy='N/a'
851.         treat_speci=data[9]
852.         if treat_speci=='':
853.             treat_speci='N/a'
854.         email=['goicovid@gmail.com',data[3], data[10], data[
11], data[12]]
855.
856.         for i in email:
857.
858.             fromaddr = "EMAIL address of the sender"
859.             toaddr = "EMAIL address of the receiver"
860.
861.
862.             # instance of MIMEMultipart
863.             msg = MIMEMultipart()
864.
865.             fromaddr = 'covidquarantineassistance@gmail.com'
866.
867.             my_password="gocoronago"
868.             # storing the receivers email address
869.             toaddr= i
870.
871.
872.             msg['Subject'] = "COVID Quarantine Assistance Pr
ogram Report"
873.
874.             msg["From"]=fromaddr
875.             msg['To']=toaddr
876.
877.             body = "Dear Sir/ma'am\n\nPlease find attached a
copy of Mr/Mrs "+str(name)+ "'s COVID Quarantine assistance program health sta
ts report.\n\nFollowing are the user's details:\n\nName: "+str(name)+"\nGender:
"+str(gender)+"\nContact No: "+str(contact_no)+"\nEmail: "+str(email[1])+"\nAg
e: "+str(age)+"\nAddress: "+str(address)+"Medical Disorder: "+str(med_disorder)

```

```

+ "\nBlood Group: "+str(bld_grp)+"\nAllergy: "+str(algy)+"\nSpecial Treatment: "
+str(treat_speci)+"\n\nThe users health has not been looking good for the past
few days. He/she should seek professional help as soon as possible\n\nThank you
for using our programme\nRegards\nCQAP team"
877.         msg.attach(MIMEText(body, 'plain'))
878.
879.
880.         # open the file to be sent
881.         attachment = open(filename, "rb")
882.
883.
884.
885.         # instance of MIMEBase and named as p
886.         p = MIMEBase('application', 'octet-stream')
887.
888.         # To change the payload into encoded form
889.         p.set_payload((attachment).read())
890.
891.         # encode into base64
892.         encoders.encode_base64(p)
893.
894.         p.add_header('Content-
Disposition', "attachment; filename= %s" % filename)
895.
896.         # attach the instance 'p' to instance 'msg'
897.         msg.attach(p)
898.         s = smtplib.SMTP_SSL('smtp.gmail.com')
899.         s.login(fromaddr,my_password)
900.         s.sendmail(fromaddr,toaddr, msg.as_string())
901.         s.quit()
902.
903.
904.         elif (avg_d5>45):
905.
906.             stats_rows[2][0]=str(1)
907.             stats_rows[6][0]=str(2)
908.             stats_rows[10][0]=str(3)
909.             stats_rows[14][0]=str(4)
910.             stats_rows[18][0]=str(5)
911.
912.             with open(filename, 'w') as file:
913.                 file.truncate()
914.
915.             with open(filename, 'a+', newline='') as file:
916.                 with file:
917.                     write = csv.writer(file)
918.                     write.writerows(stats_rows)
919.
920.             scoretext=tk.Label(page5,bg="white", fg="green",text
="Your Health Score is: "+ str(round(avg_d5,2))+ '/100',font="TimesNewRoman 15 i
talic").place(x=500, y=105)
921.             select5()
922.
923.             b=[]
924.             data=[]
925.             mycursor.execute("select name, gender, contact_no, e
mail, age, address, med_disorder, bld_grp, algy, treat_speci, pricon1_email, pr

```

```

    icon2_email, pricon3_email from users where username='" + t1 + "' ") ##gives fo
rm 't1'
926.         tab2=mycursor.fetchall()
927.         for z in tab2:
928.             b.append(z)
929.         for i in z:
930.             data.append(i)
931.
932.         #print(emails)
933.         name=data[0]
934.         gender=data[1]
935.         if gender=='1':
936.             gender='M'
937.         elif gender=='2':
938.             gender='F'
939.         else:
940.             gender='other'
941.         contact_no=data[2]
942.         age=data[4]
943.         address=data[5]
944.         med_disorder=data[6]
945.         if med_disorder=='':
946.             med_disorder='N/a'
947.         bld_grp=data[7]
948.         algy=data[8]
949.         if algy=='':
950.             algy='N/a'
951.         treat_speci=data[9]
952.         if treat_speci=='':
953.             treat_speci='N/a'
954.         email=[data[3]]
955.
956.
957.         #print(email)
958.
959.         for i in email:
960.
961.             fromaddr = "EMAIL address of the sender"
962.             toaddr = "EMAIL address of the receiver"
963.
964.
965.             # instance of MIMEMultipart
966.             msg = MIMEMultipart()
967.
968.             fromaddr ='covidquarantineassistance@gmail.com'
969.
970.             my_password="gocoronago"
971.             # storing the receivers email address
972.             toaddr= i
973.
974.
975.             msg['Subject'] = "COVID Quarantine Assistance Pr
ogram Report"
976.             msg["From"]=fromaddr
977.             msg['To']=toaddr
978.

```

```

979.
980.         body = "Mr/Mrs "+str(name)+"\n\nPlease find attached a copy of your COVID Quarantine assistance program health stats report.\n\nUser details:\n\nName: "+str(name)+"\nGender: "+str(gender)+"\nContact No: "+str(contact_no)+"\nEmail: "+str(email[0])+"\nAge: "+str(age)+"\nAddress: "+str(address)+"Medical Disorder: "+str(med_disorder)+"\nBlood Group: "+str(bld_grp)+"\nAllergy: "+str(algy)+"\nSpecial Treatment: "+str(treat_speci)+"\n\nYour health has been looking fine for the past few days. There is no need urgent to seek professional help.\n\nThank you for using our programme\nRegards\nCQAP team"
981.         msg.attach(MIMEText(body, 'plain'))
982.
983.
984.         # open the file to be sent
985.         attachment = open(filename, "rb")
986.         # instance of MIMEBase and named as p
987.         p = MIMEBase('application', 'octet-stream')
988.
989.         # To change the payload into encoded form
990.         p.set_payload((attachment).read())
991.
992.         # encode into base64
993.         encoders.encode_base64(p)
994.
995.         p.add_header('Content-Disposition', "attachment; filename= %s" % filename)
996.
997.         # attach the instance 'p' to instance 'msg'
998.         msg.attach(p)
999.
1000.
1001.         s = smtplib.SMTP_SSL('smtp.gmail.com')
1002.         s.login(fromaddr,my_password)
1003.         s.sendmail(fromaddr,toaddr, msg.as_string())
1004.         s.quit()
1005.
1006.         def clicked():
1007.             if (len(stats_rows)==4 or len(stats_rows)==8 or (len(stats_rows)==12 and avg_d3>45) or (len(stats_rows)==16 and ((float(stats_rows[2][19])+float(stats_rows[6][19])+float(stats_rows[10][19]))/3)>45)):
1008.                 l1=["Thank you for entering your data!","Please come back tomorrow.","You may close the application."]
1009.                 messagebox.showinfo('Covid Quarantine Assistance',"
\n".join(l1))
1010.                 root.destroy()
1011.             elif (len(stats_rows)>20) or ((len(stats_rows)>12) and (len(stats_rows)<17) and ((float(stats_rows[2][19])+float(stats_rows[6][19])+float(stats_rows[10][19]))/3)<=45):
1012.                 l1=["You have already entered your health stats for the maximum number of days and the program has been terminated for your account.", "Please login from a different ID or create a new account.", "Thank you. We hope you liked using COVID Quarantine Assistance"]
1013.                 messagebox.showinfo('Covid Quarantine Assistance',"
\n".join(l1))
1014.                 root.destroy()
1015.
1016.
1017.         clicked()

```

```

1018.
1019.
1020.
1021.         else:
1022.             messagebox.showinfo('Error',"Please fill all mandatory fields")
1023.
1024.
1025.         #####
1026.         #####
1027.         #####
1028.         var1= IntVar()
1029.         c1=Checkbutton(page2,text="Dry Cough",variable=var1,bg="white",font="Geo
1030.             rgia 11")
1031.         c1.place(x=60,y=150)
1032.         var2= IntVar()
1033.         c2=Checkbutton(page2,text="Fever",variable=var2,bg="white",font="Georgia
1034.             11")
1035.         c2.place(x=60,y=190)
1036.
1037.         var3= IntVar()
1038.         c3=Checkbutton(page2,text="Chest Pain",variable=var3,bg="white",font="Ge
1039.             orgia 11")
1040.         c3.place(x=60,y=230)
1041.
1042.         var4= IntVar()
1043.         c4=Checkbutton(page2,text="Difficulty in breathing",variable=var4,bg="wh
1044.             ite",font="Georgia 11")
1045.         c4.place(x=60,y=270)
1046.
1047.         myimgsave=Image.open("save.png")
1048.         resizesav=myimgsave.resize((90,50),Image.ANTIALIAS)
1049.         ne21=ImageTk.PhotoImage(resizesav)
1050.         save_button=tk.Button(page2,bg="white",image=ne21,command=save,borderwid
1051.             th=0)
1052.         save_button.place(x=1110,y=640)
1053.         var5= IntVar()
1054.         c5=Checkbutton(page2,text="Sore throat",variable=var5,bg="white",font="G
1055.             eorgia 11")
1056.         c5.place(x=60,y=310)
1057.         var6= IntVar()
1058.         c6=Checkbutton(page2,text="Body pain and aches",variable=var6,bg="white"
1059.             ,font="Georgia 11")
1060.         c6.place(x=60,y=350)
1061.         var7= IntVar()
1062.         c7=Checkbutton(page2,text="Diarrhoea",variable=var7,bg="white",font="Geo
1063.             rgia 11")
1064.         c7.place(x=60,y=390)

```



```

1064.
1065.     var8= IntVar()
1066.     c8=Checkbutton(page2,text="Headache",variable=var8,bg="white",font="Geor
gia 11")
1067.     c8.place(x=60,y=430)
1068.
1069.     var9= IntVar()
1070.     c9=Checkbutton(page2,text="Loss of taste or smell",variable=var9,bg="whi
te",font="Georgia 11")
1071.     c9.place(x=60,y=470)
1072.
1073.     var10= IntVar()
1074.     c10=Checkbutton(page2,text="Tiredness",variable=var10,bg="white",font="G
eorgia 11")
1075.     c10.place(x=60,y=510)
1076.
1077.     var11= IntVar()
1078.     c11=Checkbutton(page2,text="Conjunctivitis",variable=var11,bg="white",fon
t="Georgia 11")
1079.     c11.place(x=60,y=550)
1080.
1081.     var12= IntVar()
1082.     c12=Checkbutton(page2,text="Rash or skin discolouration",variable=var12,
bg="white",font="Georgia 11")
1083.     c12.place(x=60,y=590)
1084.
1085.     var13= IntVar()
1086.     c13=Checkbutton(page2,text="Loss of speech or movement",variable=var13,b
g="white",font="Georgia 11")
1087.     c13.place(x=60,y=630)
1088.
1089.
1090.     ##temp
1091.     temp=tk.Label(page2,fg="black",text="Enter Current Temperature (°C):",bg
="white",font="Georgia 11",).place(x=660,y=150)
1092.     temp_e=tk.Entry(page2,width=21,font="Verdana 9 bold italic",relief="sunk
en",borderwidth=3)
1093.     temp_e.place(x=1000,y=150)
1094.
1095.
1096.     ##weight
1097.     wt=tk.Label(page2,fg="black",text="Enter Weight (Kgs):",bg="white",font=
"Georgia 11",).place(x=660,y=200)
1098.     wt_e=tk.Entry(page2,width=21,font="Verdana 9 bold italic",relief="sunken
",borderwidth=3)
1099.     wt_e.place(x=1000,y=200)
1100.
1101.
1102.     ##bloodpressure
1103.     bp=tk.Label(page2,fg="black",text="Enter Blood Pressure (mm Hg):",bg="wh
ite",font="Georgia 11",).place(x=660,y=300)
1104.     bpp=tk.Label(page2,fg="black",text="(Systolic/Diastolic)",bg="white",fon
t="Georgia 11",).place(x=660,y=330)
1105.     bp_e=tk.Entry(page2,width=21,font="Verdana 9 bold italic",relief="sunken
",borderwidth=3)
1106.     bp_e.place(x=1000,y=300)
1107.

```

```

1108.
1109.     ##heartrate
1110.     hr=tk.Label(page2,fg="black",text="Enter Heart Rate(Bpm):",bg="white",font="Georgia 11 ",).place(x=660,y=250)
1111.     hr_e=tk.Entry(page2,width=21,font="Verdana 9 bold italic",relief="sunken",borderwidth=3)
1112.     hr_e.place(x=1000,y=250)
1113.
1114.
1115.     def select3():
1116.         tc.select(3)
1117.         tc.hide(2)
1118.
1119.     #####
1120.     ##tab 4
1121.     canvass = tk.Canvas(page3, width=1280, height=700)
1122.     canvass.grid()
1123.     back11=Image.open("day 3 bad.jpeg")#program ends here
1124.     resizee=back11.resize((1200,700),Image.ANTIALIAS)
1125.     new11=ImageTk.PhotoImage(resizee)
1126.     canvass.create_image(640,360,image=new11)
1127.
1128.     def select4():
1129.         tc.select(4)
1130.         tc.hide(3)
1131.
1132.
1133.
1134.
1135.     ##tab5
1136.     ##bad day 5
1137.     canvass_w = tk.Canvas(page4, width=1280, height=700)
1138.     canvass_w.grid()
1139.     back15=Image.open("day 5 bad.jpeg")#program ends here
1140.     resizeea=back15.resize((1260,780),Image.ANTIALIAS)
1141.     new112=ImageTk.PhotoImage(resizeea)
1142.     canvass_w.create_image(640,360,image=new112)
1143.
1144.     def select5():
1145.         tc.select(5)
1146.         tc.hide(4)
1147.
1148.
1149.     ##tab6
1150.     ##good after 5 days
1151.     canvass_ww = tk.Canvas(page5, width=1280, height=700)
1152.     canvass_ww.grid()
1153.     back151=Image.open("day 5 good thank you.jpeg")#program ends here
1154.     resizeez=back151.resize((1260,780),Image.ANTIALIAS)
1155.     new113=ImageTk.PhotoImage(resizeez)
1156.     canvass_ww.create_image(640,360,image=new113)
1157.
1158.     root.mainloop()

```

Sample Output

WELCOME TO

COVID QUARANTINE ASSISTANCE PROGRAM

User ID :

Enter Password:

OR

CREATE NEW ACCOUNT

Name :

Username :

Contact number :

Email ID :

Age:

Address:

Gender : ☒ M ☐ F ☐ Other

Have you travelled internationally in last the 30 days : ☒ Yes ☐ No

Allergies to any specific medication? :

Under any specific treatment, if yes specify :

ENTER PASSWORD:

CONFIRM PASSWORD:

Primary contact 1 name:

Primary contact 1 email id:

Primary contact 2 name:

Primary contact 2 email id:

Primary contact 3 name:

Primary contact 3 email id:


Blood Group :

Medical Disorder (if any):

CQA-1
WELCOME REGISTER


WELCOME TO

COVID QUARANTINE ASSISTANCE PROGRAM




User ID :

Enter Password:

 LOGIN

OR

 REGISTER

CQA-1
GENERAL

HEALTH QUESTIONS

Are you facing any of the following symptoms?

☒ Dry Cough
☒ Fever
☒ Chest Pain
☒ Difficulty in breathing
☒ Sore throat
☒ Body pain and aches
☒ Diarrhoea
☒ Headache
☒ Loss of taste or smell
☒ Tiredness
☐ Conjunctivitis
☒ Rash or skin discolouration
☐ Loss of speech or movement

Enter Current Temperature (°C):

Enter Weight (Kgs):

Enter Heart Rate(Bpm):

Enter Blood Pressure (mm Hg):

(Systolic/Diastolic)

SAVE

CQA-1
GENERAL

Welcome to day 2

HEALTH QUESTIONS

Are you facing any of the following symptoms?

☒ Dry Cough
☐ Fever
☒ Chest Pain
☒ Difficulty in breathing
☒ Sore throat
☒ Body pain and aches
☐ Diarrhoea
☒ Headache
☒ Loss of taste or smell
☒ Tiredness
☐ Conjunctivitis
☒ Rash or skin discolouration
☐ Loss of speech or movement

Enter Current Temperature (°C): 33
Enter Weight (Kgs): 62
Enter Heart Rate(Bpm): 79
Enter Blood Pressure (mm Hg): 70/95

Covid Quarantine Assistance
Thank you for entering your data! Please come back tomorrow. You may close the application.
OK

SAVE

CQA-1
GENERAL

Welcome to day 3

HEALTH QUESTIONS

Are you facing any of the following symptoms?

☒ Dry Cough
☒ Fever
☒ Chest Pain
☒ Difficulty in breathing
☒ Sore throat
☒ Body pain and aches
☐ Diarrhoea
☒ Headache
☒ Loss of taste or smell
☒ Tiredness
☐ Conjunctivitis
☒ Rash or skin discolouration
☐ Loss of speech or movement

Enter Current Temperature (°C): 34
Enter Weight (Kgs): 61
Enter Heart Rate(Bpm): 75
Enter Blood Pressure (mm Hg): 80/94 (Systolic/Diastolic)

SAVE

Your health is not looking good for the past 3 days.

Your Health Score is: 31.91/100



You should consider seeing a doctor.
Please check your E-mail for your health report.
A copy of your health report has also been sent to your primary contacts.
This is the end of this program

Thank You for using COVID QUARANTINE ASSISTANCE PROGRAM™

COVID Quarantine Assistance Program Report Inbox X



covidquarantineassistance@gmail.com

7:22 PM (1 minute ago) ☆ ↶ ⋮

to me ▾

Dear Sir/mam

Please find attached a copy of Mr/Mrs Siddhartha's COVID Quarantine assistance program health stats report.

Following are the user's details:

Name: Siddhartha
Gender: M
Contact No: 968690545
Email: siddhartha.kayitha@gmail.com
Age: 32
Address: B-0231, Brigade Meadows
Medical Disorder: N/a
Blood Group: B+
Allergy: No
Special Treatment: N/a

The users health has not been looking good for the past few days. He/she should seek professional help as soon as possible

Thank you for using our programme
Regards
CQAP team



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Day	Dry Cough	Fever	Chest Pain	Difficulty in breathin	Sore throat	Body Pains and Ach	Diarhoea	Headache	Loss of taste or smell	Tiredness	Conjunctivitis	Rash or skin discolou	Loss of speech or loc	Temperature
2															
3		1 Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	No	
4															
5	Day	Dry Cough	Fever	Chest Pain	Difficulty in breathin	Sore throat	Body Pains and Ach	Diarhoea	Headache	Loss of taste or smell	Tiredness	Conjunctivitis	Rash or skin discolou	Loss of speech or loc	Temperature
6															
7		2 Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	No	
8															
9	Day	Dry Cough	Fever	Chest Pain	Difficulty in breathin	Sore throat	Body Pains and Ach	Diarhoea	Headache	Loss of taste or smell	Tiredness	Conjunctivitis	Rash or skin discolou	Loss of speech or loc	Temperature
10															
11		3 Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	No	

Constraints and Scope

The following is a list of the current shortcomings of our project and some ideas we hope to implement in the near future.

- The project runs based on the symptoms entered by the user and will not be to detect special asymptomatic cases.
- We created this project mainly for the elderly who aren't as self-sufficient as the youth. Yet, the successful operation of the project is based on installation and proper working of softwares such as mysql which may be rather complex for a senior citizen who isn't versed in it.
- Currently the project is restricted to only guiding a user and helping him determine if he may be infected . We aim at providing a live track of other cases nearby.
- The health stats entered are easily manipulatable and there is no system to detect a manually changed data stats file.
- The program is dependent on extrinsic factors such as a stable internet connection. Thus, it's prone to crashing.
- By basing the program on an online server database, we can eliminate the requirement of a local database such as mysql.
- Tasks such as calculating health score and sending emails are time consuming, especially for low specification machines. We are working on minimizing the run time of the program to make it run smooth and efficiently.

Bibliography

- <https://www.geeksforgeeks.org>
- <https://www.w3schools.com>
- <https://stackoverflow.com>
- <https://github.com>
- <https://www.youtube.com>
- Computer Science with Python - Sumita Arora