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

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

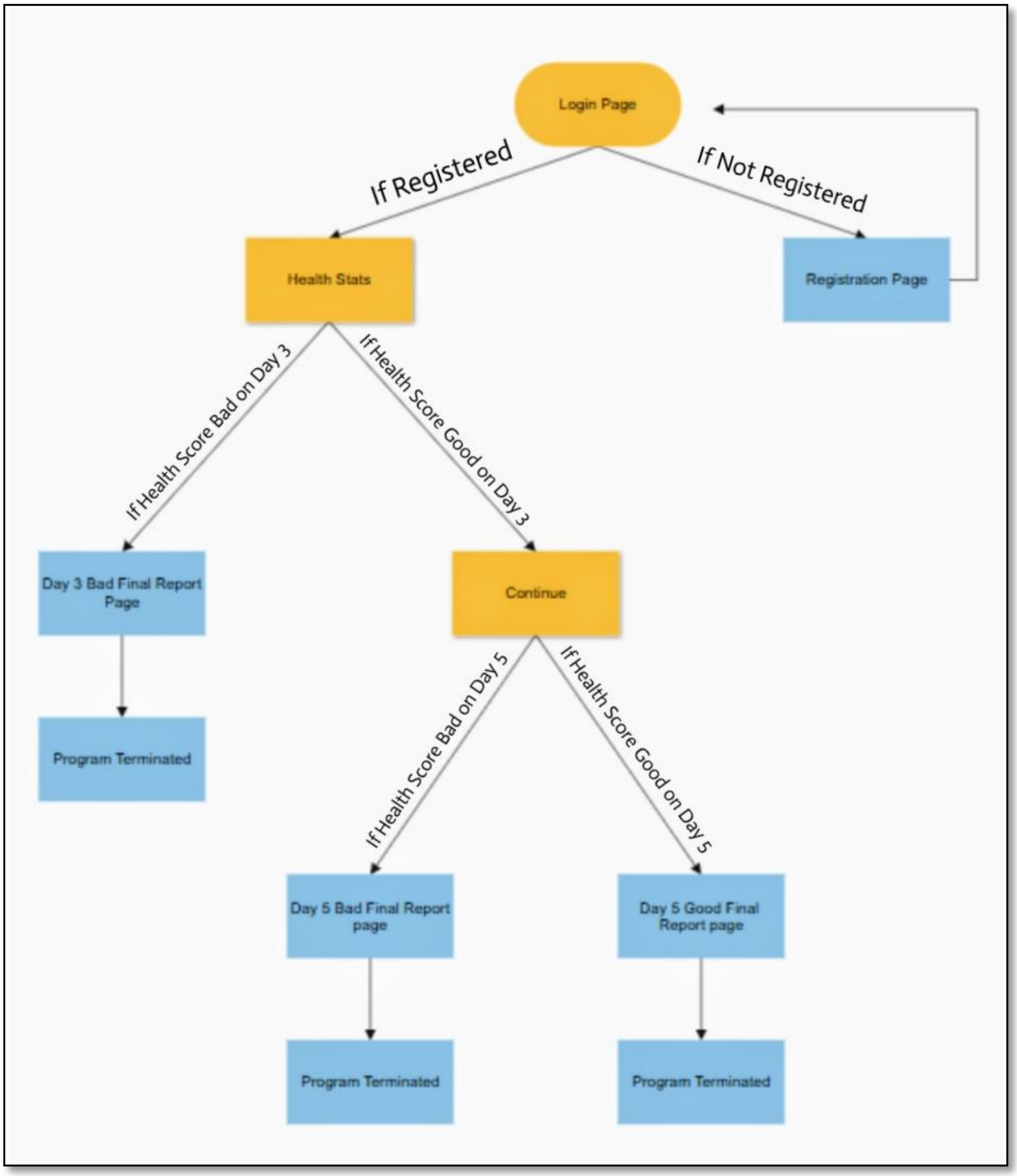
The following is a comprehensive list of the minimum system requirements to be able to execute this project successfully--

- 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


Login Page
If Registered
If Not Registered
Health Stats
Registration Page
If Health Score Bad on Day 3
If Health Score Good on Day 3
Day 3 Bad Final Report Page
Continue
Program Terminated
If Health Score Bad on Day 5
If Health Score Good on Day 5
Day 5 Bad Final Report page
Day 5 Good Final Report page
Program Terminated
Program Terminated

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 eail 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

```python
import tkinter as tk
from tkinter import *
from tkinter import ttk
from PIL import Image,ImageTk #PIL -> Pillow
from tkinter import messagebox
import random
import mysql.connector
import csv
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.mime.base import MIMEBase
from email import encoders
import os.path
from os import path

###########################################################################################################################################################################

mydb = mysql.connector.connect(
  host="localhost",
  user="root",
  passwd="user",
  database="cqap"
)
mycursor = mydb.cursor()
#############################################################################################################################################################################################

##tab1
root = Tk()
root.title("CQA-1")
root.configure(bg="white")
root.geometry("2000x1000")
tc=ttk.Notebook(root)
tc.pack(pady=0,padx=0)
page0 = Frame(tc,width=1280,height=720)
page1 = Frame(tc,width=1280,height=720 )
page2 = Frame(tc,width=1280,height=720 )
page3 = Frame(tc,width=1280,height=720 )
page4 = Frame(tc,width=1280,height=720 )
page5 = Frame(tc,width=1280,height=720 )
page0.pack(expand = 1,fill="both")
page1.pack(expand = 1,fill="both")
page2.pack(expand = 1,fill="both")
page3.pack(expand = 1,fill="both")
page5.pack(expand = 1,fill="both")
tc.add(page0 , text='WELCOME')
tc.add(page1 , text='REGISTER')
tc.add(page2 , text='GENERAL')
tc.add(page3 , text='RESULT-DAY3')
tc.add(page4 , text='RESULT-DAY5')
tc.add(page5 , text='RESULT-DAY5')
tc.hide(1)
tc.hide(2)
tc.hide(3)
tc.hide(4)
tc.hide(5)
canvas = tk.Canvas(page0, width=1280, height=720)
canvas.grid()
back=Image.open("welcome.jpeg")
resize=back.resize((800,690),Image.ANTIALIAS)
new=ImageTk.PhotoImage(resize)
canvas.create_image(400,360,image=new)
back1=Image.open("logo.png")
resize=back1.resize((400,450),Image.ANTIALIAS)
new1=ImageTk.PhotoImage(resize)
canvas.create_image(1100,160,image=new1)
noaccount=tk.Label(page0,fg="black",text="OR",bg="white",font="CaviarDreams 20 bold ").place(x=1080,y=550)
user=tk.Label(page0,fg="black",text=" User ID :",font="CaviarDreams 10 bold ").place(x=860,y=400)
user1_e=tk.Entry(page0,width=45,font="CaviarDreams 10 bold ",bg="white",borderwidth=3)
user1_e.place(x=950,y=400)
pwd=tk.Label(page0,fg="black",text="   Enter Password: ",font="CaviarDreams 10 bold").place(x=810,y=440)
pwd1_e=tk.Entry(page0,width=45,font="CaviarDreams 10 bold ",bg="white",borderwidth=3,show="*")
pwd1_e.place(x=950,y=440)


##############################################################################################################################################

global t1
t1=user1_e.get()
def select1():
    tc.select(1)
    tc.hide(0)
def select2():
    tc.select(2)
    tc.hide(1)
    tc.hide(0)
    def day_no_check():
        global day_count
        if path.exists(filename)==False:
            day_count='Welcome to day 1'
        else:
            global pre_stats_rows
            pre_stats_rows=[]
            with open(filename, 'r') as file:
                reader = csv.reader(file)
                for row in reader:
                    pre_stats_rows.append(row)
            if len(pre_stats_rows)==4:
                day_count='Welcome to day 2'
                  elif len(pre_stats_rows)==8:
                       day_count='Welcome to day 3'
                   elif len(pre_stats_rows)==12:
                       day_count='Welcome to day 4'
                   elif len(pre_stats_rows)==16:
                       day_count='Welcome to day 5'
                   else:
                       day_count= 'Day Count Limit Reached'

                   gnqu=tk.Label(page2,fg="red",text=day_count,bg="white",font="Georgia 15 bold italic").place(x=560,y=15)
           day_no_check()


       ##login
       def login():
           global t1
           t1=user1_e.get()
           global t2
           t2=pwd1_e.get()

           global filename
           filename = t1+"_"+"daily_stats.csv"

           if t1=="" or t2=="":
               messagebox.showinfo('Error',"Please fill all fields")
               return

           def username_check():
               mycursor.execute("select username from users")
               tab=mycursor.fetchall()
               u=[]
               for i in tab:
                   u.append(i[0])
               for z in u:
                   if z==t1:
                       return True
               else:
                   messagebox.showinfo('Error',"Incorrect UserID")

           pwd_check=True
           if username_check()==True:
               mycursor.execute("select pwd from users where username =  '" + t1 + "' ") ##gives form 't1'
               tab1=mycursor.fetchall()
               for j in tab1:
                   if j[0]==t2:
                       select2()
                   else:
                       pwd_check=False

           if pwd_check==False:
               messagebox.showinfo('Error',"Incorrect password")


       ### button from login to question
       ##need to add if condition to check if user id matches database
       weltoque_but1=Image.open("login.png")
       resize=weltoque_but1.resize((120,50),Image.ANTIALIAS)
       ne47=ImageTk.PhotoImage(resize)

       welqnext_but1=tk.Button(page0,bg="white",image=ne47,command=login,borderwidth=0)
       welqnext_but1.place(x=1050,y=474)


       #own create button
       myimgnext_but1=Image.open("register.png")
       resize=myimgnext_but1.resize((120,50),Image.ANTIALIAS)
       ne44=ImageTk.PhotoImage(resize)

       next_but1=tk.Button(page0,bg="white",image=ne44,command=select1,borderwidth=0)
       next_but1.place(x=1050,y=600)

       ###############################################################################################################################

       def variables():
           global n1,n2,n3,n4,n5,n6,n7,n8,n9,n10,n11,n12,n13,n14,n15,n16,n17,n18,n19,n20,lpg1,lpg2
           n1=name_e.get()
           n2=username_e.get()
           n3=pwd_e.get()
           n4=cpwd_e.get()
           n5=varg.get()
           n6=ph_e.get()
           n7=email_e.get()
           n8=age_e.get()
           n9=address_e.get('1.0',END)
           n10=pr1_e.get()
           n11=pr1e_e.get()
           n12=pr2_e.get()
           n13=pr2e_e.get()
           n14=pr3_e.get()
           n15=pr3e_e.get()
           n16=md_e.get()
           n17=click.get()
           n18=allr_e.get()
           n19=vart.get()
           n20=sptr_e.get()
           lpg1=[n1,n2,n3,n4,n5,n6,n7,n8,n9,n10,n11,n12,n13,n14,n15,n16,n17,n18,n19,n20]
           lpg2=[n1,n2,n3,n4,n5,n6,n7,n8,n9,n10,n11,n12,n13,n14,n15,n17,n19]



       def exists_id():
           mycursor.execute("select username from users")
           tab=mycursor.fetchall()
           u=[]
           for i in tab:
               u.append(i[0])
           if n2 in u:
               return True
           else:
               return False


       def exists_email():
           global n7
           mycursor.execute("select email from users")
           tab=mycursor.fetchall()
           u=[]
           for i in tab:
               u.append(i[0])
           if n7 in u:
               return True
           else:
               return False



       def exists_phone():
           mycursor.execute("select contact_no from users")
           tab=mycursor.fetchall()
           u=[]
           for i in tab:
               u.append(i[0])
           for z in u:
               if z==n6:
                   return True
               else:
                   return False

       def check(email):
           import re
           regex = '^[a-z0-9]+[.]?[a-z0-9]+[@]\w+[.]\w{2,3}$'
           if(re.search(regex,email)):
               return True
           else:
               return False


       def validate():
           duplicatemails=False
           e_mails=[n7,n11,n13,n15]
           e_mails_set=set(e_mails)

           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!="":
               if exists_id()==True:
                   messagebox.showinfo('Error-Username', "This username is already taken.")
               if exists_email()==True:
                   print(n7)
                   messagebox.showinfo('Error-Email', "The entered email-id aldreday exists. Please login.")
               if exists_phone()==True:
                   messagebox.showinfo('Error-Contact Number', "This contact number is already taken.")
               a=n1.split()
               b=n10.split()
               c=n12.split()
               d=n12.split()
               for i in a:
                   if i!=" " and i.isalpha()==False :
                           messagebox.showinfo('Error-Name',"Enter alphabets only")

               for i in b:
                   if i!=" " and i.isalpha()==False :
                           messagebox.showinfo('Error-Name',"Enter alphabets only")

               for i in c:
                   if i!=" " and i.isalpha()==False :
                           messagebox.showinfo('Error-Name',"Enter alphabets only")

               for i in d:
                   if i!=" " and i.isalpha()==False :
                           messagebox.showinfo('Error-Name',"Enter alphabets only")
               #if n1.isalpha()==False and n1.split()!=" "  or n10.isalpha()==False and n10.split()!=" " or n12.isalpha()==False and n12.split()!=" " or n14.isalpha()==False or n14.split()!=" " :
                #   messagebox.showinfo('Error-Name',"Enter alphabets only")
               if len(n3)< 8:
                   messagebox.showinfo('Error-login',"Password should be greater than 8 characters")
               if n3!=n4:
                   messagebox.showinfo('Error-password',"Passwords do not match")
               if len(n6)!=10 :
                   messagebox.showinfo('Error-contact',"contact number should be 10 digits only")
               if len(n9)< 10 :
                   messagebox.showinfo('Error-address',"Please enter valid address")
               if n6.isdigit()==False:
                   messagebox.showinfo('Error-contact',"Enter numeric characters only")
               if check(n7)==False or check(n11)==False or check(n13)==False or check(n15)==False:
                   messagebox.showinfo('Error-Email',"Invalid email, Please enter a valid email")
               if n8.isdigit()==False:
                   messagebox.showinfo('Error-age',"Enter a valid age")

               if len(e_mails)!=len(e_mails_set):
                    duplicatemails=True
                    messagebox.showinfo('Error-Email',"Emails can not be repeated")

               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:
                sql = "INSERT INTO users (name,username,pwd,cpwd,gender,contact_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)"
                val = (n1,n2,n3,n4,n5,n6,n7,n8,n9,n10,n11,n12,n13,n14,n15,n16,n17,n18,n20)
                mycursor.execute(sql, val)
                mydb.commit()
                messagebox.showinfo('Login',"account created")
                '''''with open(filename,'a') as csvfile:
                     csvwriter = csv.writer(csvfile)  # creating a csv writer object
                     csvwriter.writerow([n2])      # writing the fields
                    # csvwriter.writerow(rows)      # writing the data rows '''
                select0()
        else:
            messagebox.showinfo('Error-fields',"Fill all required fields (all except medical disorder, allergy, special treatment)")

    def details():
        variables()
        validate()


    def select0():
        tc.select(0)


    canvas = tk.Canvas(page1, width=1280, height=720)
    canvas.grid()
    f_back=Image.open("bgwhite2.jpg")
    resize=f_back.resize((1280,800),Image.ANTIALIAS)
    n=ImageTk.PhotoImage(resize)
    canvas.create_image(633,324, image=n)

    register=tk.Label(page1,fg="black",text="CREATE NEW ACCOUNT",bg="white",font="Georgia 20 bold").place(x=60,y=50)


    #name
    global name_e
    name=tk.Label(page1,fg="black",text="Name :",bg="White",font="Georgia 11").place(x=60,y=110)
    name_e=tk.Entry(page1,width=25,font="Georgia 11",bg="white",borderwidth=3)
    name_e.place(x=350,y=100)


    #username
    username=tk.Label(page1,fg="black",text="Username :",bg="White",font="Georgia 11").place(x=60,y=140)
    username_e=tk.Entry(page1,width=25,font="Georgia 11",bg="white",borderwidth=3)
```

```
        username_e.place(x=350,y=135)


        #phone
        ph=tk.Label(page1,fg="black",text=" Contact number :",bg="white",font="Georgia 11").place(x=60,y=170)
        ph_e=tk.Entry(page1,width=25,font="Georgia 11",bg="white",borderwidth=3)

        ph_e.place(x=350,y=165)


        #email
        email=tk.Label(page1,fg="black",text=" Email ID :",bg="white",font="Georgia 11").place(x=60,y=205)
        email_e=tk.Entry(page1,width=25,font="Georgia 11",bg="white",borderwidth=3)
        email_e.place(x=350,y=205)

        #age
        age=tk.Label(page1,fg="black",text=" Age:",bg="white",font="Georgia 11",).place(x=60,y=240)
        age_e=tk.Entry(page1,width=25,font="Georgia 11",relief="sunken",borderwidth=3)
        age_e.place(x=350,y=240)


        #address
        address=tk.Label(page1,fg="black",text=" Address:",bg="white",font="Georgia 11",).place(x=60,y=270)
        address_e=Text(page1,width=25,height=4,font="Georgia 11",relief="sunken",borderwidth=3)
        address_e.place(x=350,y=270)

        #primary1name
        pr1=tk.Label(page1,fg="black",text=" Primary contact 1 name:",bg="white",font="Georgia 11",).place(x=60,y=360)
        pr1_e=tk.Entry(page1,width=25,font="Georgia 11",relief="sunken",borderwidth=3)
        pr1_e.place(x=350,y=360)

        #primary1email
        pr1e=tk.Label(page1,fg="black",text=" Primary contact 1 email id:",bg="white",font="Georgia 11",).place(x=60,y=390)
        pr1e_e=tk.Entry(page1,width=25,font="Georgia 11",relief="sunken",borderwidth=3)
        pr1e_e.place(x=350,y=390)

        #primary2
        pr2=tk.Label(page1,fg="black",text=" Primary contact 2 name:",bg="white",font="Georgia 11",).place(x=60,y=420)
        pr2_e=tk.Entry(page1,width=25,font="Georgia 11",relief="sunken",borderwidth=3)
        pr2_e.place(x=350,y=420)


        #primary2email
```

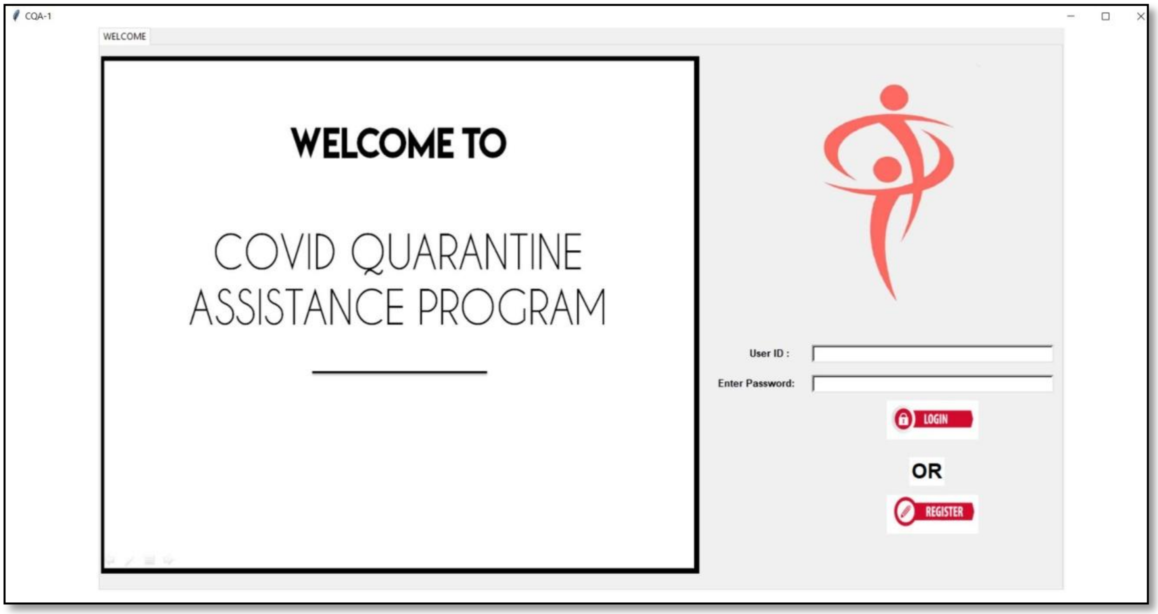
```
    pr2e=tk.Label(page1,fg="black",text=" Primary contact 2 email id:",bg="white",font="Georgia 11",).place(x=60,y=450)
    pr2e_e=tk.Entry(page1,width=25,font="Georgia 11",relief="sunken",borderwidth=3)
    pr2e_e.place(x=350,y=450)

    #primary3
    pr3=tk.Label(page1,fg="black",text=" Primary contact 3 name:",bg="white",font="Georgia 11",).place(x=60,y=480)
    pr3_e=tk.Entry(page1,width=25,font="Georgia 11",relief="sunken",borderwidth=3)
    pr3_e.place(x=350,y=480)
    #primary3email
    pr3e=tk.Label(page1,fg="black",text=" Primary contact 3 email id:",bg="white",font="Georgia 11",).place(x=60,y=510)
    pr3e_e=tk.Entry(page1,width=25,font="Georgia 11",relief="sunken",borderwidth=3)
    pr3e_e.place(x=350,y=510)

    #password
    pwd=tk.Label(page1,fg="black",text=" ENTER PASSWORD:",bg="white",font="Georgia 11").place(x=861,y=290)
    pwd_e=tk.Entry(page1,width=22,font="Georgia 10",bg="white",borderwidth=3,show="*")
    pwd_e.place(x=1050,y=290)

    #confirmpassword
    cpwd=tk.Label(page1,fg="black",text=" CONFIRM PASSWORD:",bg="white",font="Georgia 11").place(x=840,y=330)
    cpwd_e=tk.Entry(page1,width=22,font="Georgia 10",bg="white",borderwidth=3,show="*")
    cpwd_e.place(x=1050,y=330)


    #medical disorder
    md=tk.Label(page1,fg="black",text=" Medical Disorder (if any):",bg="white",font="Georgia 11").place(x=60,y=615)
    md_e=tk.Entry(page1,width=25,font="Georgia 11",bg="white",borderwidth=3)
    md_e.place(x=350,y=615)

    #bloofgroup
    bldgrp=tk.Label(page1,fg="black",text=" Blood Group :",bg="white",font="Georgia 11").place(x=60,y=540)
    click=StringVar()
    names=['SELECT',' A+',' B+',' O+', ' AB+',' A-',' B-',' O-',' AB-']
    click.set(names[0])
    drop=OptionMenu(page1,click,*names)
    drop.config(width=20,bg="white",font="Georgia 11",borderwidth=3)
    drop.place(x=350,y=540)


    #gender-radio button
    gen=tk.Label(page1,fg="black",text="Gender :",bg="white",font="Georgia 11").place(x=952,y=100)
    varg=IntVar()
```

```python
        R1 = Radiobutton(page1, text="M", variable=varg, value="1",command=variables)
        R1.place(x=1050,y=100)
        R2 = Radiobutton(page1, text="F", variable=varg, value="2",command=variables)
        R2.place(x=1120,y=100)
        R3 = Radiobutton(page1, text="Other", variable=varg, value="3",command=variables)
        R3.place(x=1190,y=100)


        #allergies
        allr=tk.Label(page1,fg="black",text="Allergies to any specific medication? :",bg="white",font="Georgia 11").place(x=761,y=190)
        allr_e=tk.Entry(page1,width=22,font="Georgia 11",bg="white",borderwidth=3)
        allr_e.place(x=1050,y=190)


        #travelhistory- raDIO BUTTON
        trvhis=tk.Label(page1,fg="black",text="Have you travelled internationally in last the 30 days :",bg="white",font="Georgia 11").place(x=647,y=140)
        vart=IntVar()
        R1 = Radiobutton(page1, text="Yes", variable=vart, value=1,command=variables)
        R1.place(x=1050,y=140)
        R2 = Radiobutton(page1, text="No", variable=vart, value=2,command=variables)
        R2.place(x=1120,y=140)


        #specifictreatment
        sptr=tk.Label(page1,fg="black",text="Under any specfic treatment, if yes specify :",bg="white",font="Georgia 11").place(x=720,y=235)
        sptr_e=tk.Entry(page1,width=22,font="Georgia 11",bg="white",borderwidth=3)
        sptr_e.place(x=1050,y=235)


        #created account successfully thing button
        myimgreg=Image.open("create_acc.png")
        resize=myimgreg.resize((180,100),Image.ANTIALIAS)
        ne41=ImageTk.PhotoImage(resize)

        reg_button=tk.Button(page1,bg="white",image=ne41,command=details,borderwidth=0)
        reg_button.place(x=1050,y=620)


        acc_created=tk.Label(page1,fg="black",bg="white",font="Georgia 11")
        acc_created.place(x=960,y=550)

        def select3():
            tc.select(3)
            tc.hide(2)

        ##############################################################################################################################################################################################
        #tab3 general questions
        Canvas = tk.Canvas(page2, width=1280, height=720)
        Canvas.grid()
        f_back=Image.open("bgwhite2.jpg")
        resize=f_back.resize((1280,850),Image.ANTIALIAS)
        no=ImageTk.PhotoImage(resize)
        Canvas.create_image(633,324, image=no)

        gnqu=tk.Label(page2,fg="black",text="HEALTH QUESTIONS",bg="white",font="Georgia 20 bold").place(x=60,y=60)


        ##input questions
        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)
        ####save function
        def save():
            global filename
            filename = t1+"_"+"daily_stats.csv"
            avg_d3=0
            avg_d5=0
            h1=var1.get()
            h2=var2.get()
            h3=var3.get()
            h4=var4.get()
            h5=var5.get()
            h6=var6.get()
            h7=var7.get()
            h8=var8.get()
            h9=var9.get()
            h10=var10.get()
            h11=var11.get()
            h12=var12.get()
            h13=var13.get()
            h14=temp_e.get()
            h15=wt_e.get()
            h16=bp_e.get()
            h17=hr_e.get()
        ###################################################################################################################################################################

            ##health validation
            def healthvalid():
                if h14.isdigit()==False:
                    messagebox.showinfo('Error-Temperature', "Enter valid Temperature.")
                if h15.isdigit()==False:
                    messagebox.showinfo('Error-Weight', "Enter valid Weight")
                if re.match('^[0-9]{2,3}/[0-9]{2,3}$', h16):
                    h16_check=True
                else:
                    h16_check=False
                    messagebox.showinfo('Error-Blood Pressure', "Enter valid Blood Pressure")
                if h17.isdigit()==False:
                    messagebox.showinfo('Error-Heart Rate', "Enter valid Heart Rate")
                if h14.isdigit()==True and h15.isdigit()==True and h17.isdigit()==True and h16_check==True:
                    return True

            ##if len(ld)==4:
            if h14!="" and h15!="" and h16!="" and h17!="":
                if healthvalid()==True:
                    c=0
                    p=[]#point backend
                    l=[]#csv yes or no
                    #f=open

                    h18=''
                    l.append(h18)
                    if h1==1:#cough
                        l.append("Yes")
                        p.append(4)
                    else:
                        l.append("No")
                    if h2==1:#fever
                        l.append("Yes")
                        p.append(2)
                    else:
                        l.append("No")

                    if h3==1:#chest pain
                        l.append("Yes")
                        p.append(4)
                    else:
                        l.append("No")
                    if h4==1:#difficulty in breathing
                        l.append("Yes")
                        p.append(5)
                    else:
                        l.append("No")
                    if h5==1:#sore throat
                        l.append("Yes")
                        p.append(2)
                    else:
                        l.append("No")
                    if h6==1:#body pains and aches
                        l.append("Yes")
                        p.append(2)
                    else:
                        l.append("No")
                    if h7==1:#diarrhoea
                        l.append("Yes")
                        p.append(3)
                    else:
                        l.append("No")
                    if h8==1:#headache
                        l.append("Yes")
                        p.append(3)
                    else:
                        l.append("No")
                    #
                    if h9==1:#loss of taste or smell
                        l.append("Yes")
                        p.append(3)
                    else:
                        l.append("No")
                    if h10==1:#tiredness
                        l.append("Yes")
                        p.append(3)
                    else:
                        l.append("No")
                    if h11==1:#conjuctivitis
                        l.append("Yes")
                        p.append(2)
                    else:
                        l.append("No")
                    if h12==1:#rash or skin discoloration
                        l.append("Yes")
                        p.append(3)
                    else:
                        l.append("No")
                    if h13==1:#loss of speech or movement
                        l.append("Yes")
                        p.append(5)
                    else:
                        l.append("No")


                    #temp
                    l.append(h14)
                    #weight
                    l.append(h15)
                    #BP
                    l.append(h16)
                    #Heart rate
                    l.append(h17)

                    if vart==1: #international travel
                        p.append(6)
                        '''''l.append("Yes")
                    else:
                        l.append("No")'''
                    for j in p:
                        c+=j
                    l.append(100-((c/47)*100))
                    ##columns
                    fields=["Day","Dry Cough", "Fever", "Chest Pain", "Difficulty in breathing", "Sore throat", "Body Pains and Aches", "Diarrhoea", "Headache", "Loss of taste or smell", "Tiredness", "Conjuctivits", "Rash or skin discoloration", "Loss of speech or locomotion","Temperature", "Weight", "Blood Pressure", "Heart Rate", "Health Score"]
                    rows=[]
                    for i in l:
                        rows.append(i)


                    # name of csv file
                    # writing to csv file



                    with open(filename,'a') as csvfile:
                         csvwriter = csv.writer(csvfile)  # creating a csv writer object
                         csvwriter.writerow(fields)      # writing the fields
                         csvwriter.writerow(rows)      # writing the data rows

                    global stats_rows
                    stats_rows=[]
                    with open(filename, 'r') as file:
                        reader = csv.reader(file)
                        for row in reader:
                            stats_rows.append(row)

                    #checking if health is bad after day 3
                    if len(stats_rows)==12:

                        print('d3 got')

                        #print (stats_rows)
                        print ('following are the scores extracted and averaged from first 3 days')

                        stats_d1=[]
                        stats_d1=stats_rows[2]
                        #print (stats_d1[19])

                        stats_d2=[]
                        stats_d2=stats_rows[6]
                        #print (stats_d2[19])

                        stats_d3=[]
                        stats_d3=stats_rows[10]
                        #print (stats_d3[19])

                        #global avg_d3
                        avg_d3=((float(stats_d1[19])+float(stats_d2[19])+float(stats_d3[19]))/3)
                        print (avg_d3)


                        if (avg_d3<=45):

                            stats_rows[2][0]=str(1)
                            stats_rows[6][0]=str(2)
                            stats_rows[10][0]=str(3)

                            with open(filename, 'w') as file:
                                file.truncate()

                            with open(filename, 'a+', newline ='') as file:
                                with file:
                                    write = csv.writer(file)
                                    write.writerows(stats_rows)


                            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)
                            select3()

                            b=[]
                            data=[]
                            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'
                            tab2=mycursor.fetchall()
                            for z in tab2:
                                b.append(z)
                            for i in z:
                                data.append(i)

                            #print(emails)
                            name=data[0]
                            gender=data[1]
                            if gender=='1':
                                gender='M'
                            elif gender=='2':
                                gender='F'
                            else:
                                gender='other'
                            contact_no=data[2]
                            age=data[4]
                            address=data[5]
                            med_disorder=data[6]
                            if med_disorder=='':
                                med_disorder='N/a'
                            bld_grp=data[7]
                            algy=data[8]
                            if algy=='':
                                algy='N/a'
                            treat_speci=data[9]
                            if treat_speci=='':
                                treat_speci='N/a'
                            email=[data[3], data[10], data[11], data[12]]


                            #print(email)

                            for i in email:

                                fromaddr = "EMAIL address of the sender"
                                toaddr = "EMAIL address of the receiver"


                                # instance of MIMEMultipart
                                msg = MIMEMultipart()

                                fromaddr ='covidquarantineassisstance@gmail.com'
                                my_password="gocoronago"
                                # storing the receivers email address
                                toaddr= i

                                msg['Subject'] = "COVID Quarantine Assistance Program Report"
                                msg["From"]=fromaddr
                                msg['To']=toaddr

                                #l=["Thank you for entering your data!","Please come back tomorrow.","You may close the application."]
                                #messagebox.showinfo('Covid Quarantine Assistance',"\n".join(l))
                                body = "Dear Sir/ma'am\n\nPlease find attached a copy of Mr/Mrs "+str(name)+ "'s COVID Quarantine assistance program health stats 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"
                                msg.attach(MIMEText(body, 'plain'))


                                # open the file to be sent
                                #filename = "sardarth_daily_stats.csv.csv"
                                attachment = open(filename, "rb")

                                # instance of MIMEBase and named as p
                                p = MIMEBase('application', 'octet-stream')

                                # To change the payload into encoded form
                                p.set_payload((attachment).read())

                                # encode into base64
                                encoders.encode_base64(p)

                                p.add_header('Content-Disposition', "attachment; filename= %s" % filename)

                                # attach the instance 'p' to instance 'msg'
                                msg.attach(p)

                                s = smtplib.SMTP_SSL('smtp.gmail.com')
                                s.login(fromaddr,my_password)
                                s.sendmail(fromaddr,toaddr, msg.as_string())
                                s.quit()


                    #checking if health is bad after day 5
                    if len(stats_rows)==20:

                        #print (stats_rows)
                        print ('following are the scores extracted and averaged from first 5 days')

                        stats_d1=[]
                        stats_d1=stats_rows[2]
                        #print (stats_d1[19])

                        stats_d2=[]
                        stats_d2=stats_rows[6]
                        #print (stats_d2[19])

                        stats_d3=[]
                        stats_d3=stats_rows[10]
                        #print (stats_d3[19])


                        stats_d4=[]
                        stats_d4=stats_rows[14]

                        stats_d5=[]
                        stats_d5=stats_rows[18]


                        avg_d5=(float(stats_d1[19])+float(stats_d2[19])+float(stats_d3[19])+float(stats_d4[19])+float(stats_d5[19]))/5
                        print (avg_d5)

                        if (avg_d5<=45):

                            stats_rows[2][0]=str(1)
                            stats_rows[6][0]=str(2)
                            stats_rows[10][0]=str(3)
                            stats_rows[14][0]=str(4)
                            stats_rows[18][0]=str(5)

                            with open(filename, 'w') as file:
                                file.truncate()

                            with open(filename, 'a+', newline ='') as file:
                                with file:
                                    write = csv.writer(file)
                                    write.writerows(stats_rows)


                            scoretext=tk.Label(page4,bg="white", fg="red",text="Your Health Score is: "+ str(round(avg_d5,2))+'/100',font="TimesNewRoman 15 italic").place(x=500, y=131)
                            select4() ## put button there, to view avg score after 5 days

                            b=[]
                            data=[]
                            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'
                            tab2=mycursor.fetchall()
                            for z in tab2:
                                b.append(z)
                            for i in z:
                                data.append(i)

                            #print(emails)
                            name=data[0]
                            gender=data[1]
                            if gender=='1':
                                gender='M'
                            elif gender=='2':
                                gender='F'
                            else:
                                gender='other'
                            contact_no=data[2]
                            age=data[4]
                            address=data[5]
                            med_disorder=data[6]
                            if med_disorder=='':
                                med_disorder='N/a'
                            bld_grp=data[7]
                            algy=data[8]
                            if algy=='':
                                algy='N/a'
                            treat_speci=data[9]
                            if treat_speci=='':
                                treat_speci='N/a'
                            email=['goicovid@gmail.com',data[3], data[10], data[11], data[12]]

                            for i in email:

                                fromaddr = "EMAIL address of the sender"
                                toaddr = "EMAIL address of the receiver"


                                # instance of MIMEMultipart
                                msg = MIMEMultipart()

                                fromaddr ='covidquarantineassisstance@gmail.com'

                                my_password="gocoronago"
                                # storing the receivers email address
                                toaddr= i



                                msg['Subject'] = "COVID Quarantine Assistance Program Report"
                                msg["From"]=fromaddr
                                msg['To']=toaddr

                                body = "Dear Sir/ma'am\n\nPlease find attached a copy of Mr/Mrs "+str(name)+ "'s COVID Quarantine assistance program health stats report.\n\nFollowing are the user's details:\n\nName: "+str(name)+"\nGender: "+str(gender)+"\nContact No: "+str(contact_no)+"\nEmail: "+str(email[1])+"\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"
                                msg.attach(MIMEText(body, 'plain'))


                                # open the file to be sent
                                attachment = open(filename, "rb")



                                # instance of MIMEBase and named as p
                                p = MIMEBase('application', 'octet-stream')

                                # To change the payload into encoded form
                                p.set_payload((attachment).read())

                                # encode into base64
                                encoders.encode_base64(p)

                                p.add_header('Content-Disposition', "attachment; filename= %s" % filename)

                                # attach the instance 'p' to instance 'msg'
                                msg.attach(p)
                                s = smtplib.SMTP_SSL('smtp.gmail.com')
                                s.login(fromaddr,my_password)
                                s.sendmail(fromaddr,toaddr, msg.as_string())
                                s.quit()


                        elif (avg_d5>45):

                            stats_rows[2][0]=str(1)
                            stats_rows[6][0]=str(2)
                            stats_rows[10][0]=str(3)
                            stats_rows[14][0]=str(4)
                            stats_rows[18][0]=str(5)

                            with open(filename, 'w') as file:
                                file.truncate()

                            with open(filename, 'a+', newline ='') as file:
                                with file:
                                    write = csv.writer(file)
                                    write.writerows(stats_rows)

                            scoretext=tk.Label(page5,bg="white", fg="green",text="Your Health Score is: "+ str(round(avg_d5,2))+'/100',font="TimesNewRoman 15 italic").place(x=500, y=105)
                            select5()

                            b=[]
                            data=[]
                            mycursor.execute("select name, gender, contact_no, email, age, address, med_disorder, bld_grp, algy, treat_speci, pricon1_email, pr
icon2_email, pricon3_email from users where username='" + t1 + "' ") ##gives form 't1'
                            tab2=mycursor.fetchall()
                            for z in tab2:
                                b.append(z)
                            for i in z:
                                data.append(i)

                            #print(emails)
                            name=data[0]
                            gender=data[1]
                            if gender=='1':
                                gender='M'
                            elif gender=='2':
                                gender='F'
                            else:
                                gender='other'
                            contact_no=data[2]
                            age=data[4]
                            address=data[5]
                            med_disorder=data[6]
                            if med_disorder=='':
                                med_disorder='N/a'
                            bld_grp=data[7]
                            algy=data[8]
                            if algy=='':
                                algy='N/a'
                            treat_speci=data[9]
                            if treat_speci=='':
                                treat_speci='N/a'
                            email=[data[3]]


                            #print(email)

                            for i in email:

                                fromaddr = "EMAIL address of the sender"
                                toaddr = "EMAIL address of the receiver"


                                # instance of MIMEMultipart
                                msg = MIMEMultipart()

                                fromaddr ='covidquarantineassisstance@gmail.com'
                                my_password="gocoronago"
                                # storing the receivers email address
                                toaddr= i



                                msg['Subject'] = "COVID Quarantine Assistance Program Report"
                                msg["From"]=fromaddr
                                msg['To']=toaddr


                                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"
                                msg.attach(MIMEText(body, 'plain'))


                                # open the file to be sent
                                attachment = open(filename, "rb")
                                # instance of MIMEBase and named as p
                                p = MIMEBase('application', 'octet-stream')

                                # To change the payload into encoded form
                                p.set_payload((attachment).read())

                                # encode into base64
                                encoders.encode_base64(p)

                                p.add_header('Content-Disposition', "attachment; filename= %s" % filename)

                                # attach the instance 'p' to instance 'msg'
                                msg.attach(p)


                                s = smtplib.SMTP_SSL('smtp.gmail.com')
                                s.login(fromaddr,my_password)
                                s.sendmail(fromaddr,toaddr, msg.as_string())
                                s.quit()

                    def clicked():
                        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)):
                            l=["Thank you for entering your data!","Please come back tomorrow.","You may close the application."]
                            messagebox.showinfo('Covid Quarantine Assisstance',"\n".join(l))
                            root.destroy()
                        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):
                            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"]
                            messagebox.showinfo('Covid Quarantine Assisstance',"\n".join(l1))
                            root.destroy()


                    clicked()
```

```



        else:
            messagebox.showinfo('Error',"Please fill all mandatory fields")


    #############################################################################################################################################################################

    var1= IntVar()
    c1=Checkbutton(page2,text="Dry Cough",variable=var1,bg="white",font="Georgia 11")
    c1.place(x=60,y=150)

    var2= IntVar()
    c2=Checkbutton(page2,text="Fever",variable=var2,bg="white",font="Georgia 11")
    c2.place(x=60,y=190)


    var3= IntVar()
    c3=Checkbutton(page2,text="Chest Pain",variable=var3,bg="white",font="Georgia 11")
    c3.place(x=60,y=230)


    var4= IntVar()
    c4=Checkbutton(page2,text="Difficulty in breathing",variable=var4,bg="white",font="Georgia 11")
    c4.place(x=60,y=270)


    myimgsave=Image.open("save.png")
    resizesav=myimgsave.resize((90,50),Image.ANTIALIAS)
    ne21=ImageTk.PhotoImage(resizesav)
    save_button=tk.Button(page2,bg="white",image=ne21,command=save,borderwidth=0)
    save_button.place(x=1110,y=640)

    var5= IntVar()
    c5=Checkbutton(page2,text="Sore throat",variable=var5,bg="white",font="Georgia 11")
    c5.place(x=60,y=310)

    var6= IntVar()
    c6=Checkbutton(page2,text="Body pain and aches",variable=var6,bg="white",font="Georgia 11")
    c6.place(x=60,y=350)

    var7= IntVar()
    c7=Checkbutton(page2,text="Diarrhoea",variable=var7,bg="white",font="Georgia 11")
    c7.place(x=60,y=390)
    var8= IntVar()
    c8=Checkbutton(page2,text="Headache",variable=var8,bg="white",font="Georgia 11")
    c8.place(x=60,y=430)

    var9= IntVar()
    c9=Checkbutton(page2,text="Loss of taste or smell",variable=var9,bg="white",font="Georgia 11")
    c9.place(x=60,y=470)

    var10= IntVar()
    c10=Checkbutton(page2,text="Tiredness",variable=var10,bg="white",font="Georgia 11")
    c10.place(x=60,y=510)

    var11= IntVar()
    c11=Checkbutton(page2,text="Conjuctivitis",variable=var11,bg="white",font="Georgia 11")
    c11.place(x=60,y=550)

    var12= IntVar()
    c12=Checkbutton(page2,text="Rash or skin discolouration",variable=var12,bg="white",font="Georgia 11")
    c12.place(x=60,y=590)

    var13= IntVar()
    c13=Checkbutton(page2,text="Loss of speech or movement",variable=var13,bg="white",font="Georgia 11")
    c13.place(x=60,y=630)


    ##temp
    temp=tk.Label(page2,fg="black",text="Enter Current Temperature (°C):",bg="white",font="Georgia 11",).place(x=660,y=150)
    temp_e=tk.Entry(page2,width=21,font="Verdana 9 bold italic",relief="sunken",borderwidth=3)
    temp_e.place(x=1000,y=150)


    ##weight
    wt=tk.Label(page2,fg="black",text="Enter Weight (Kgs):",bg="white",font="Georgia 11",).place(x=660,y=200)
    wt_e=tk.Entry(page2,width=21,font="Verdana 9 bold italic",relief="sunken",borderwidth=3)
    wt_e.place(x=1000,y=200)


    ##bloodpressure
    bp=tk.Label(page2,fg="black",text="Enter Blood Pressure (mm Hg):",bg="white",font="Georgia 11",).place(x=660,y=300)
    bpp=tk.Label(page2,fg="black",text="(Systolic/Diastolic)",bg="white",font="Georgia 11",).place(x=660,y=330)
    bp_e=tk.Entry(page2,width=21,font="Verdana 9 bold italic",relief="sunken",borderwidth=3)
    bp_e.place(x=1000,y=300)

    ##heartrate
    hr=tk.Label(page2,fg="black",text="Enter Heart Rate(Bpm):",bg="white",font="Georgia 11 ",).place(x=660,y=250)
    hr_e=tk.Entry(page2,width=21,font="Verdana 9 bold italic",relief="sunken",borderwidth=3)
    hr_e.place(x=1000,y=250)


    def select3():
        tc.select(3)
        tc.hide(2)

    ############################################################################################################################################################################################
    ##tab 4
    canvass = tk.Canvas(page3, width=1280, height=700)
    canvass.grid()
    back11=Image.open("day 3 bad.jpeg")#program ends here
    resizee=back11.resize((1200,700),Image.ANTIALIAS)
    new11=ImageTk.PhotoImage(resizee)
    canvass.create_image(640,360,image=new11)

    def select4():
        tc.select(4)
        tc.hide(3)




    ##tab5
    ##bad day 5
    canvass_w = tk.Canvas(page4, width=1280, height=700)
    canvass_w.grid()
    back15=Image.open("day 5 bad.jpeg")#program ends here
    resizeea=back15.resize((1260,780),Image.ANTIALIAS)
    new112=ImageTk.PhotoImage(resizeea)
    canvass_w.create_image(640,360,image=new112)

    def select5():
        tc.select(5)
        tc.hide(4)


    ##tab6
    ##good after 5 days
    canvass_ww = tk.Canvas(page5, width=1280, height=700)
    canvass_ww.grid()
    back151=Image.open("day 5 good thank you.jpeg")#program ends here
    resizeez=back151.resize((1260,780),Image.ANTIALIAS)
    new113=ImageTk.PhotoImage(resizeez)
    canvass_ww.create_image(640,360,image=new113)

    root.mainloop()
```

Sample Output

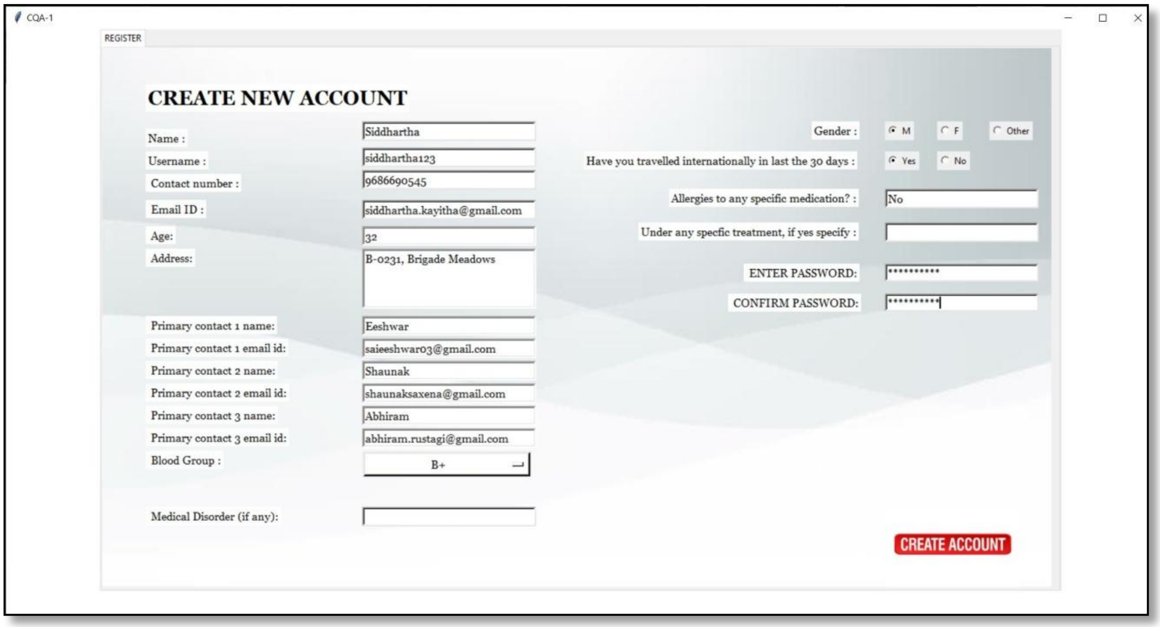


WELCOME TO

COVID QUARANTINE ASSISTANCE PROGRAM

User ID :

Enter Password:



CREATE NEW ACCOUNT

Name : Siddhartha

Username : siddhartha123

Contact number : 9686690545

Email ID : siddhartha.kayitha@gmail.com

Age: 32

Address: B-0231, Brigade Meadows

Primary contact 1 name: Eeshwar

Primary contact 1 email id: saieeshwar03@gmail.com

Primary contact 2 name: Shaunak

Primary contact 2 email id: shaunaksaxena@gmail.com

Primary contact 3 name: Abhiram

Primary contact 3 email id: abhiram.rustagi@gmail.com

Blood Group : B+

Medical Disorder (if any):

Gender : ● M ○ F ○ Other

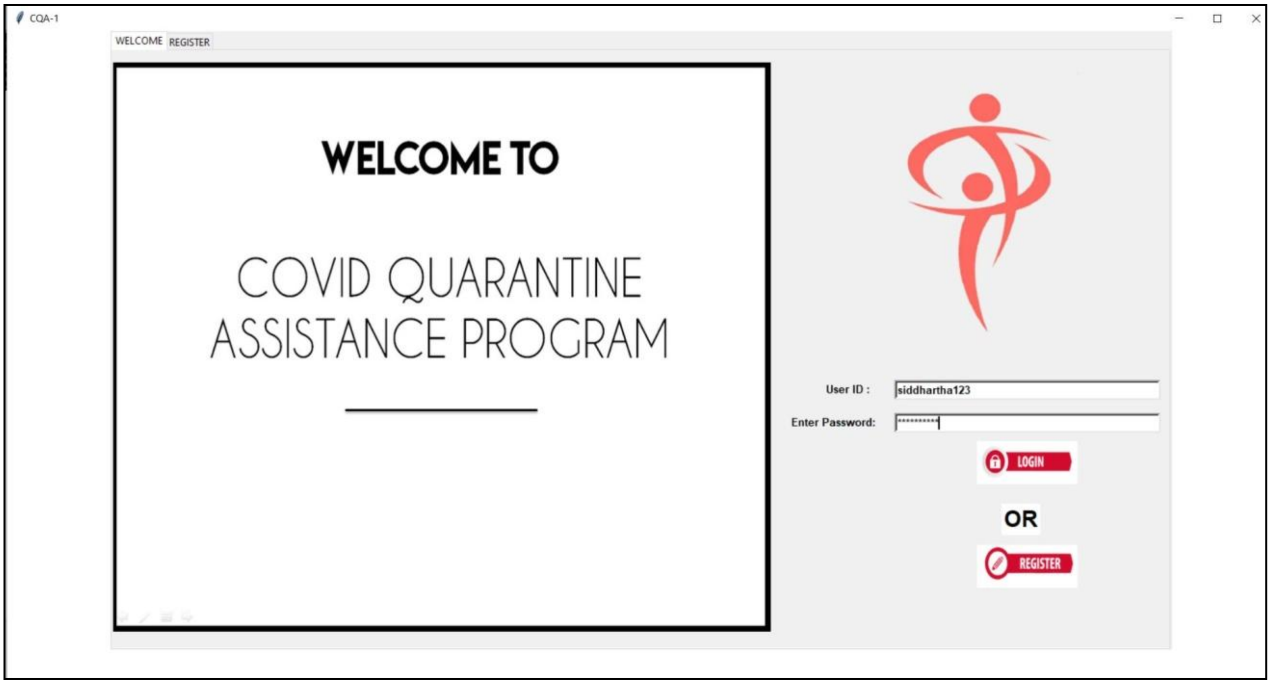
Have you travelled internationally in last the 30 days : ● Yes ○ No

Allergies to any specific medication? : No

Under any specfic treatment, if yes specify :

ENTER PASSWORD: **********

CONFIRM PASSWORD: **********

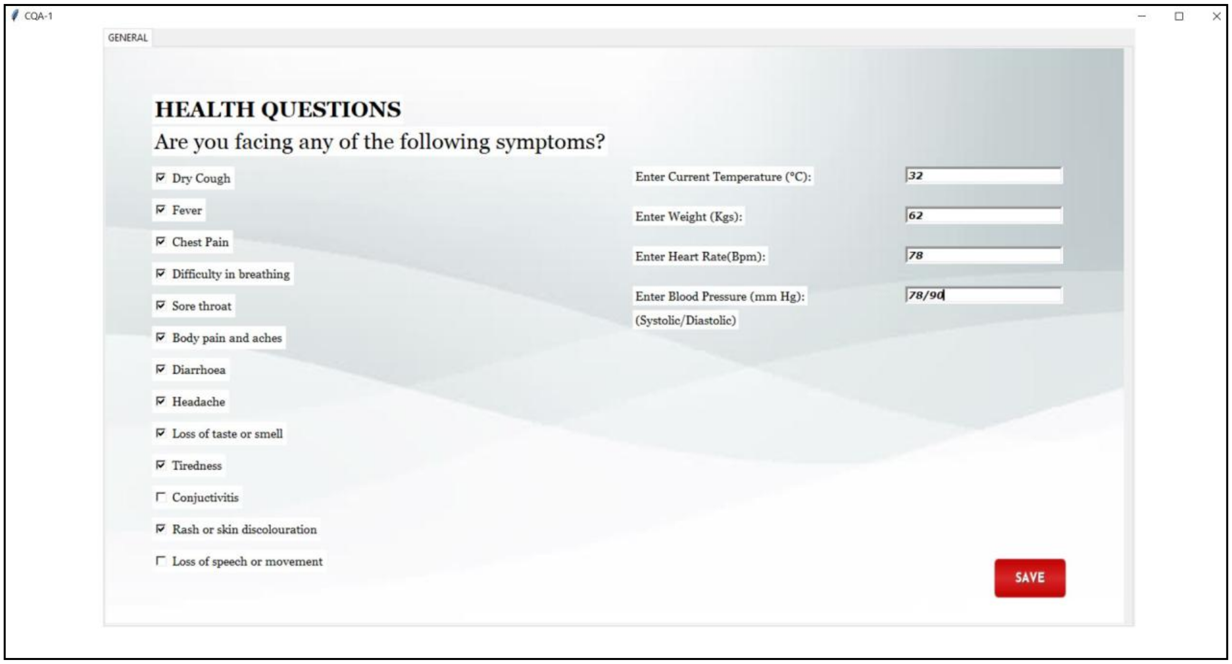


WELCOME TO

COVID QUARANTINE ASSISTANCE PROGRAM

User ID : siddhartha123

Enter Password: *********



HEALTH QUESTIONS

Are you facing any of the following symptoms?

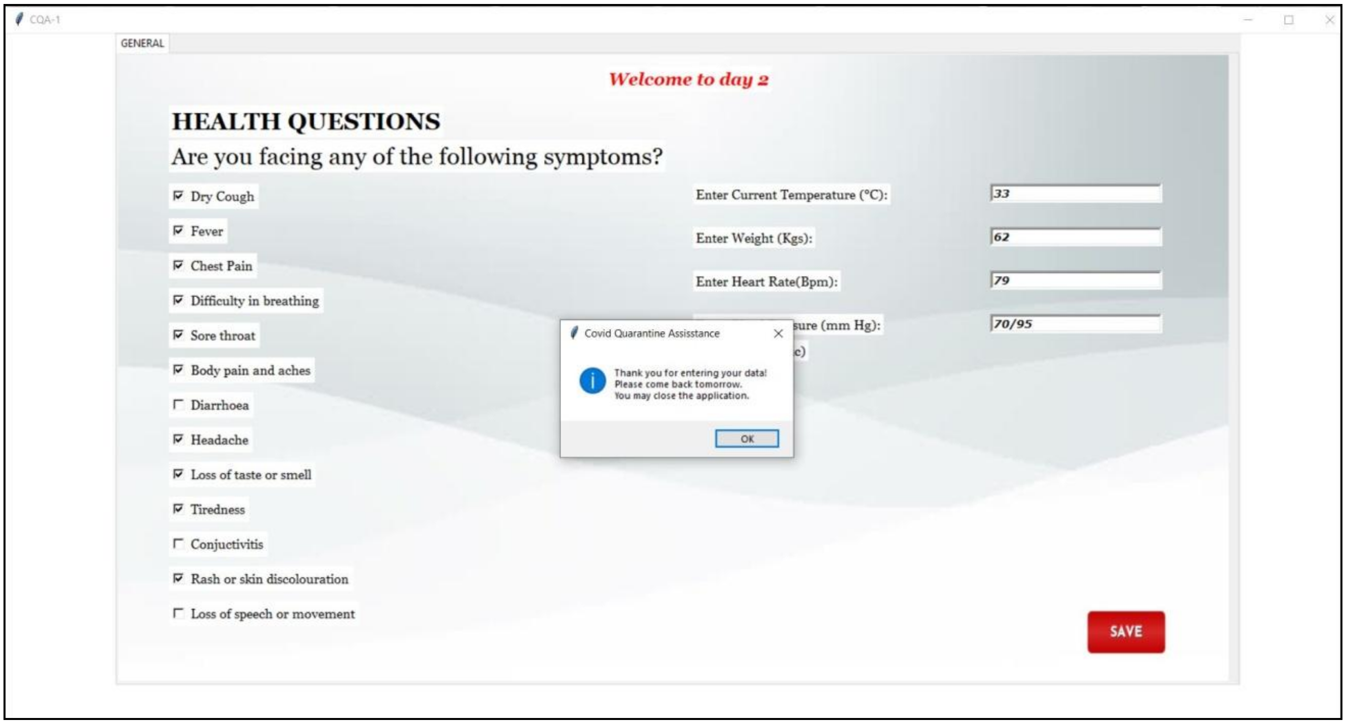
- ☑ Dry Cough
- ☑ Fever
- ☑ Chest Pain
- ☑ Difficulty in breathing
- ☑ Sore throat
- ☑ Body pain and aches
- ☑ Diarrhoea
- ☑ Headache
- ☑ Loss of taste or smell
- ☑ Tiredness
- ☐ Conjuctivitis
- ☑ Rash or skin discolouration
- ☐ Loss of speech or movement

Enter Current Temperature (°C): 32

Enter Weight (Kgs): 62

Enter Heart Rate(Bpm): 78

Enter Blood Pressure (mm Hg): (Systolic/Diastolic) 78/90



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
- □ Conjuctivitis
- ☑ Rash or skin discolouration
- □ Loss of speech or movement

Enter Current Temperature (°C): 33

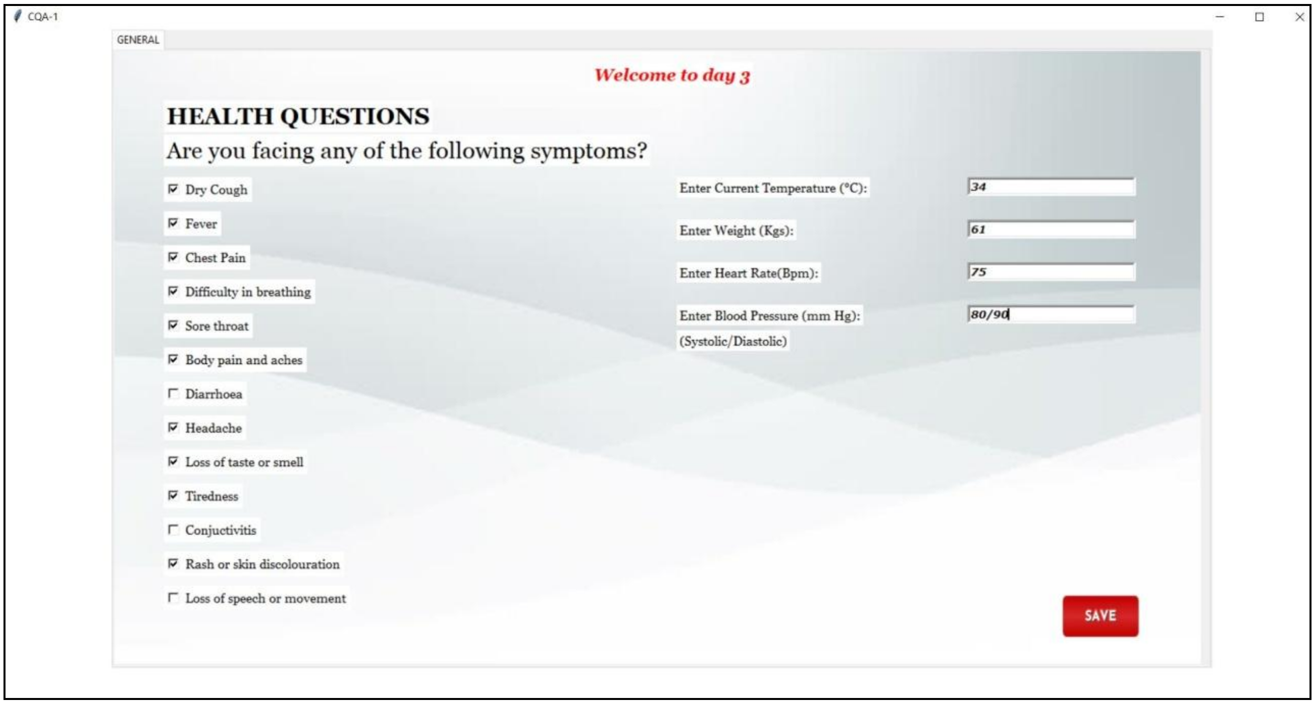
Enter Weight (Kgs): 62

Enter Heart Rate(Bpm): 79

sure (mm Hg): 70/95

c)

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



Welcome to day 3

HEALTH QUESTIONS

Are you facing any of the following symptoms?

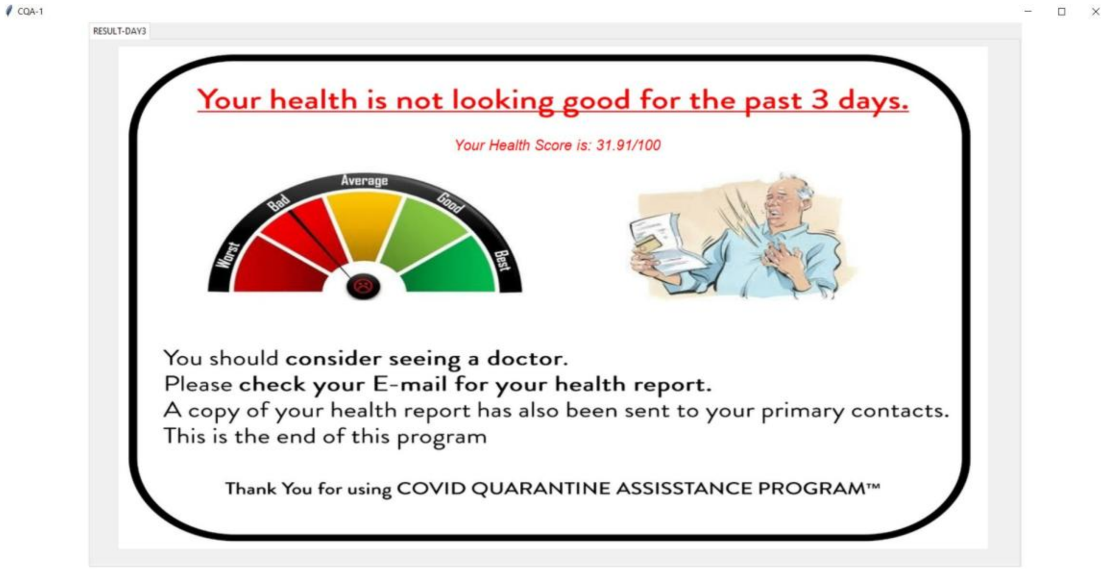
- ☑ Dry Cough
- ☑ Fever
- ☑ Chest Pain
- ☑ Difficulty in breathing
- ☑ Sore throat
- ☑ Body pain and aches
- □ Diarrhoea
- ☑ Headache
- ☑ Loss of taste or smell
- ☑ Tiredness
- □ Conjuctivitis
- ☑ Rash or skin discolouration
- □ Loss of speech or movement

Enter Current Temperature (°C): 34

Enter Weight (Kgs): 61

Enter Heart Rate(Bpm): 75

Enter Blood Pressure (mm Hg):
(Systolic/Diastolic) 80/90

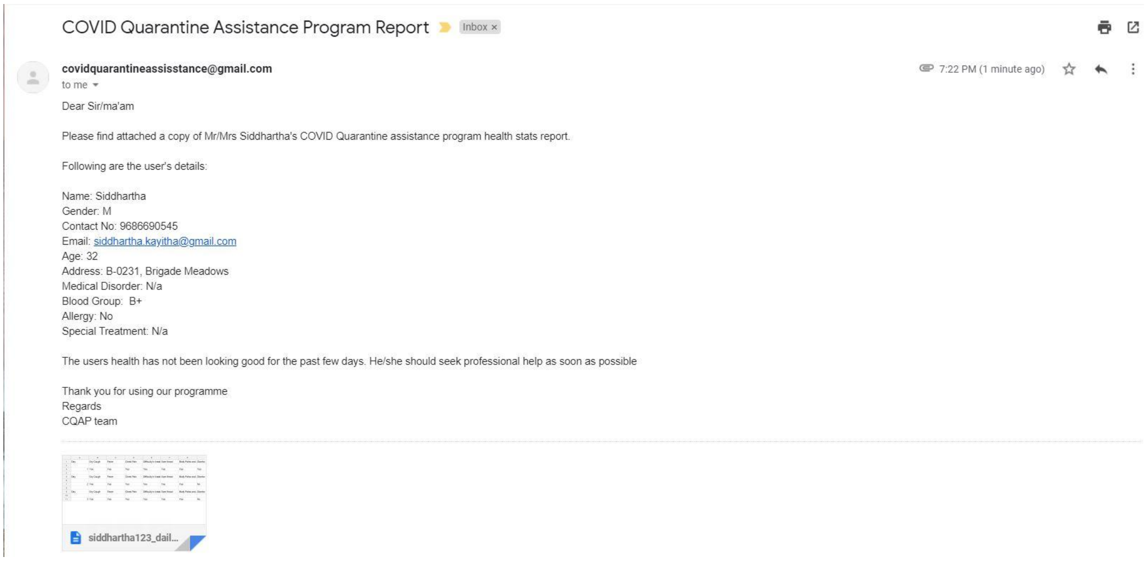


Your health is not looking good for the past 3 days.

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 ASSISSTANCE PROGRAM™



COVID Quarantine Assistance Program Report

covidquarantineassisstance@gmail.com
to me

Dear Sir/ma'am

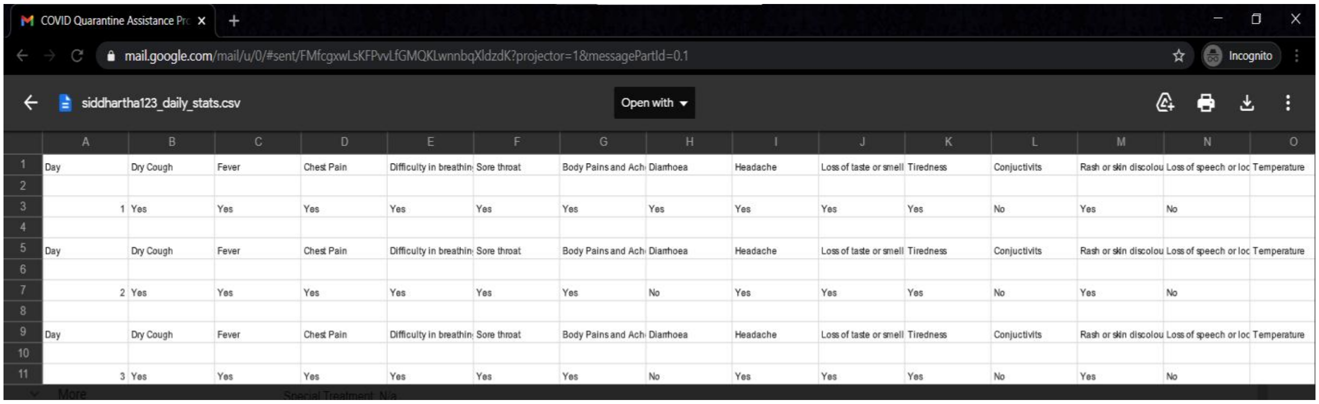
Please find attached a copy of Mr/Mrs Siddhartha's COVID Quarantine assistance program health stats report.

Following are the user's details:

Name: Siddhartha
Gender: M
Contact No: 9686690545
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


COVID Quarantine Assistance Pr
mail.google.com/mail/u/0/#sent/FMfcgxwLsKFPvvLfGMQKLwnnbqXldzdK?projector=1&messagePartId=0.1
Incognito
Open with

siddhartha123_daily_stats.csv

	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	Diarrhoea	Headache	Loss of taste or smell	Tiredness	Conjuctivits	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	Diarrhoea	Headache	Loss of taste or smell	Tiredness	Conjuctivits	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	Diarrhoea	Headache	Loss of taste or smell	Tiredness	Conjuctivits	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