

COMPUTER SCIENCE PROJECT

PYTHON HEALTH TRACKER

Jash Veragiwala

Roll No. 8747688



SINGAPORE • MALAYSIA • THAILAND • JAPAN • UAE • INDIA • VIETNAM

Certificate

Department of Information & Technology

**Global Indian International School
SMART Campus, Singapore**

This is to certify that the project entitled Python Health Tracker is a genuine work of Jash Veragiwala, Roll No. 8747688 undertaken as a part of fulfilment of Computer Science [083] practical conducted by C.B.S.E. and has been completed within stipulated time period under my guidance and supervision.

Signature

Ms. Radha Ganesh

Academic Coordinator

Computer Science Teacher

Date:

ACKNOWLEDGEMENT

I take this opportunity to express my acknowledgement and sincere gratitude to Ms. Radha Ganesh, for her valuable suggestions and able guidance required for this project. It is through her I have learnt efficient debugging skills which were helpful in completing the project on time.

Jash Veragiwala

CONTENTS

1. Introduction	5
2. About the programming tool used.....	6
3. Problem definition.....	7
4. Design requirements.....	8
5. System requirements.....	16
6. Algorithm and flowchart.....	17
7. Code listing.....	22
8. User documentation.....	32
9. Bibliography	40

1. INTRODUCTION

During the coronavirus pandemic, the need for a database to keep track of body temperatures and other symptoms has been on the rise and to cater to that demand, we have built the Python Health Tracker.

The Health Tracker has the following functions:

1. To maintain a record of the user's body temperatures.
2. To provide the user with statistics regarding the coronavirus.
3. To provide the user a graphical view of his/her body temperature over a period of time.
4. To provide the user with precautions he/she should take to keep themselves and their loved ones safe from the virus
5. To get the user contact details of the hospitals nearest to them.

2. ABOUT THE PROGRAMMING TOOL USED

“Programming languages are how programmers express and communicate ideas — and the audience for those ideas is other programmers, not computers.” – Guido van Rossum

Python is an interpreted, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it extremely attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms and can be freely distributed.

MySQL is a freely available open-source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL). SQL is the most popular language for adding, accessing, and managing content in a database. It is most noted for its quick processing, proven reliability, ease, and flexibility of use.

3. PROBLEM DEFINITION

During the Covid-19 pandemic, one of the biggest problems the world faced (apart from the virus, of course) was the management of misinformation. Web giants like Google scrambled to hunt every piece of information about the pandemic on the web, and verify if the source was valid, and then allow the users to access it.

But this problem can be easily solved if the user relies on a program like the Health Tracker that provides the user with updates regarding the pandemic and also tracks the user's health status in the process. The Health Tracker has an added benefit as well: because it keeps a record of the body temperature of the user, in the scenario that the user is infected by the virus, he/she can provide the data to the authorities so that they can track, when the user might have been infected by the virus, based on the temperatures recorded.

One of the key factors that pushed us towards the idea of making the Health Tracker was the ability of the program to scour the internet and collect information and organize it in the necessary manner, without the need for human intervention. We discovered the python module Selenium, during the pandemic and we used it for several applications (from whatsapp bots to programs to login to zoom meetings), but we felt that the best application of the program in the current scenario would be to allow the user to receive updates about the virus (in the form of statistics), from any country in the world, without having to search for it themselves.

This ties in to the first problem, as the user can be misguided by wrong information on the web, but our program ensures that the information received by the user is accurate and is not biased in any manner.

4. DESIGN REQUIREMENTS

The main goal of making a Health Tracker is to make the user experience better than any of the other solution present on the internet. In order to keep, the program interesting and interactive from a user perspective, we have made the implemented a few modules offered by Python.

The Health Tracker also needs to be useful incase the user has been infected by Covid-19. We have implemented MySQL databases and graphs to help the authorities trace when the user might have been infected by the virus, so that his close contacts can be informed and/or quarantined.

The user's details are stored in the form of a text file and a CSV file. The text file contains the details of the user, such as address and country of residence which are used in the program for various functions. The CSV file contains the temperatures of the user along with the dates when they were recorded.

1. FILE TYPES

The user's details are stored in the following form:

a. Text file



The user's verification details are stored in a text file, which is used by the program to verify the user.

In the case of a new user, a new text file is created.

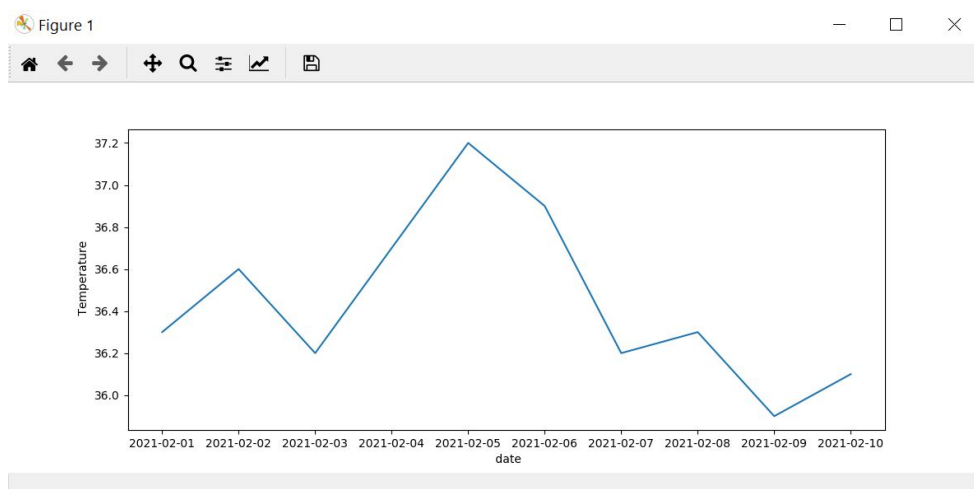
b. CSV file

Jash

date	temperature
2021-02-01	36.3
2021-02-02	36.6
2021-02-03	36.2
2021-02-06	36.9
2021-02-07	36.2
2021-02-08	36.3
2021-02-09	35.9
2021-02-10	36.1
2021-02-05	37.2

The user's temperature is stored in a CSV file which can be accessed by the user.

c. Graph



The user can access his/her temperatures in the form of a graph.

d. MySQL Database

date	temprature	
2021-02-03	36.2	
2021-02-06	36.9	
2021-02-07	36.2	
2021-02-08	36.3	
2021-02-09	35.9	
2021-02-10	36.1	
2021-02-05	37.2	

The user's temperature is also stored in a MySQL Database.

4.2 FUNCTIONS

The following are the functions used in the program:

a. dprint()

```
1. def dprint(s):
2.     for c in s:
3.         sys.stdout.write(c)
4.         sys.stdout.flush()
5.         time.sleep(0.10)
```

This function is used to create a 'typewriter' like effect, i.e., the text that is printed in the program, appears letter by letter, as if it is being typed out manually.

The function prints a line, letter by letter, with a time gap in between. It makes use of the time module.

The main reason this was implemented was to enhance the user experience.

b. name()

```
1. def name():
2.     global name_user
3.     name_user=input(" ENTER YOUR NAME: ")
```

This function is used to get the name of the user. It defines the name as a global variable so that is accessible by the rest of the program.

c. name_check():

```
1. def name_check():
2.     f = open(name_user+".txt","a")
3.     fh = open(name_user+".txt","r")
4.     line = fh.read()
5.     if len(line)>0:
6.         print(" ")
7.         dprint('' WE HAVE NOTICED THAT YOU HAVE USED THE PROGRAM
BEFORE.
8.     ARE THESE YOUR DETAILS? '')
9.         print()
10.        ch = open(name_user+".txt", "r")
11.        y = ch.readlines()
12.        for i in y:
13.            if "NAME: " in i:
14.                print(" ")
15.                n = i[6:-1]
16.                print("    NAME        |", n)
17.                time.sleep(0.5)
18.            if "AGE: " in i:
19.                n = i[5:-1]
20.                print("    AGE          |", n)
21.                time.sleep(0.5)
22.            if "GENDER: " in i:
23.                n = i[8:-1]
24.                if n.lower() == "m":
25.                    print("    GENDER    | Male")
26.                    time.sleep(0.5)
27.                else:
28.                    print("    GENDER    | Female")
29.                    time.sleep(0.5)
30.            if "ADDRESS: " in i:
31.                n = i[9:-1]
32.                print("    ADDRESS   |", n)
33.                time.sleep(0.5)
34.            if "COUNTRY: " in i:
35.                n = i[9:-1]
36.                print("    COUNTRY   |", n)
37.                print(" ")
38.        dprint(" IF YES, ENTER 'Y', ELSE ENTER 'N': ")
39.        s = input()
40.        if s.lower() == "y":
41.            global a
42.            a = 0
43.            hh = open(name_user+".txt", "r")
44.            l = hh.readlines()
45.            for i in l:
46.                if "AGE: " in i:
47.                    global age_user
48.                    age_user = i[5:-1]
49.                if "GENDER: " in i:
50.                    if i[8:-1] == "m":
51.                        global gender_user
52.                        gender_user = "m"
53.                    else:
```

```

54.             gender_user = "f"
55.         if "ADDRESS: " in i:
56.             global place_user
57.             place_user = i[9:-1]
58.         if "COUNTRY: " in i:
59.             global country_user
60.             country_user = i[9:-1]
61.         hh.close()
62.         ch.close()
63.     else:
64.         a = 1
65.         pass
66. else:
67.     f.close()
68.     fh.close()
69.     a = 1
70.     pass

```

This function is used to check whether the user has an account in the program. It does so by going through files, looking for a file with the user's name. If it finds a file with the user's name, it prints out the details of the file, asking the user to confirm if they are his/her details.

If the function does not find an existing file with the user's name, then it does nothing, and moves on.

d. age()

```

1. def age():
2.     global age_user
3.     age_user=int(input("  ENTER YOUR AGE: "))
4.     if age_user <110 or age_user>0:
5.         pass
6.     else:
7.         age()

```

This function takes the user's age. If the age is greater than 110 or less than 0, it runs the function again (using recursion).

e. gender()

```

1. def gender():
2.     global gender_user
3.     gender_user = input("  ENTER YOUR GENDER, M FOR MALE, F FOR FEMALE: ")
4.     if gender_user.lower() == "m" or gender_user == "f":
5.         pass
6.     else:
7.         print("  PLEASE ENTER YOUR GENDER CORRECTLY.")
8.         gender()

```

This function takes the gender of the user. If the user enters the gender incorrectly, the program prints an error, and runs the function again (using recursion).

f. place()

```
1. def place():
2.     global place_user
3.     place_user=input(" ENTER YOUR CURRENT ADDRESS: ")
```

This function takes the address of the user.

g. country()

```
1. def country():
2.     global country_user
3.     country_user=input(" ENTER THE COUNTRY YOU LIVE IN: ")
```

This function takes the country of the user.

h. date()

```
1. def date():
2.     dprint(" ENTER TODAY'S DATE [FORMAT: YYYY-MM-DD]: ")
3.     dat=input()
4.     if dat[4]!="-" and dat[7]!="-" and len(dat)!=10:
5.         print(" YOU ARE USING THE WRONG FORMAT. TRY AGAIN [FORMAT:
        YYYY-MM-DD].")
6.         date()
7.     else:
8.         return dat
```

This function takes the date from the user and checks it to ensure that is according to the given format.

i. hospital()

```
1. def hospital():
2.     import selenium
3.     from selenium import webdriver
4.     dprint("  YOUR CURRENT ADDRESS IS SAVED AS: " + place_user)
5.     print()
6.     dprint('"' DO YOU WANT TO FIND HOSPITALS NEAR A DIFFERENT
LOCATION OR THE ONE GIVEN ABOVE?
7.     TO ENTER A DIFFERENT LOCATION, ENTER 'Y' ELSE ENTER 'N': '"')
8.     hos = input()
9.     if hos.lower() == 'y':
10.         hospital = input("  ENTER YOUR DESIRED LOCATION: ")
11.     else:
12.         hospital = place_user
13.     print(" ")
14.     print(Fore.RED + "  YOU WILL BE REDIRECTED TO A WEBSITE.")
15.     print(Fore.RED + "  DO NOT CLICK ANYTHING ON THE POP-UP UNTIL
THE DESIRED INFORMATION HAS APPEARED.")
16.     print(" ")
17.     time.sleep(5)
18.     chrome = webdriver.Chrome("C:\\Users\\nagan\\Desktop\\Coding
Files\\chromedriver\\chromedriver") # change this if needed
19.     chrome.get("https://www.google.co.in/")
20.     time.sleep(2)
21.     bar = chrome.find_element_by_xpath('//*[@type="text"]')
22.     time.sleep(1)
23.     bar.send_keys("Hospitals near ", hospital)
24.     bar.submit()
```

This function makes use of the selenium module to get information of hospitals near the user. The selenium module scrapes the web for the necessary information and brings it to the user in the form of a webpage (in this case).

Note: In line 18 above, the information in the chrome variable needs to be changed according to where the chromedriver is located on the user's computer.

4.3 MODULES

The following modules are used in the program:

a. time

The time module is used to keep track of time and regulate the flow of some functions (dprint).

b. colorama

The colorama module allows us to add colors to program and make the user experience better.

Note: The color is only visible when the program is run on the Anaconda Prompt and not in the regular IDLE.

c. sys

The sys module is used in the dprint function to produce a 'typewriter' like effect while printing text.

d. selenium

The selenium module is used for web scraping, i.e., getting information from the internet to use in the python program. It is a module that is generally used for automation and testing, but in this case we have used it for scraping necessary data from the web and putting it together in a manner that is useful to the user.

e. mysql.connector

This module is used to connect MySQL Databases to the Python program, to store data and allow the user to access it in an easy way.

f. matplotlib.pyplot

This module is used to create the graph of the temperatures entered by the user. It produces a graph which can be downloaded by the user and stored.

5. SYSTEM REQUIREMENTS

The Health Tracker program requires the following to operate:

1. Computer:

The minimum requirements for the computer are:

- a. Windows 7
- b. 1 gigahertz (GHz) or faster 32-bit (x86) or 64-bit (x64) processor
- c. 2 GB RAM (64-bit)
- d. 16 GB available hard disk space (32-bit) or 20 GB (64-bit)
- e. Wi-Fi Card

2. Internet Connection:

Through Wi-Fi or Ethernet (for selenium module).

3. Software Requirements:

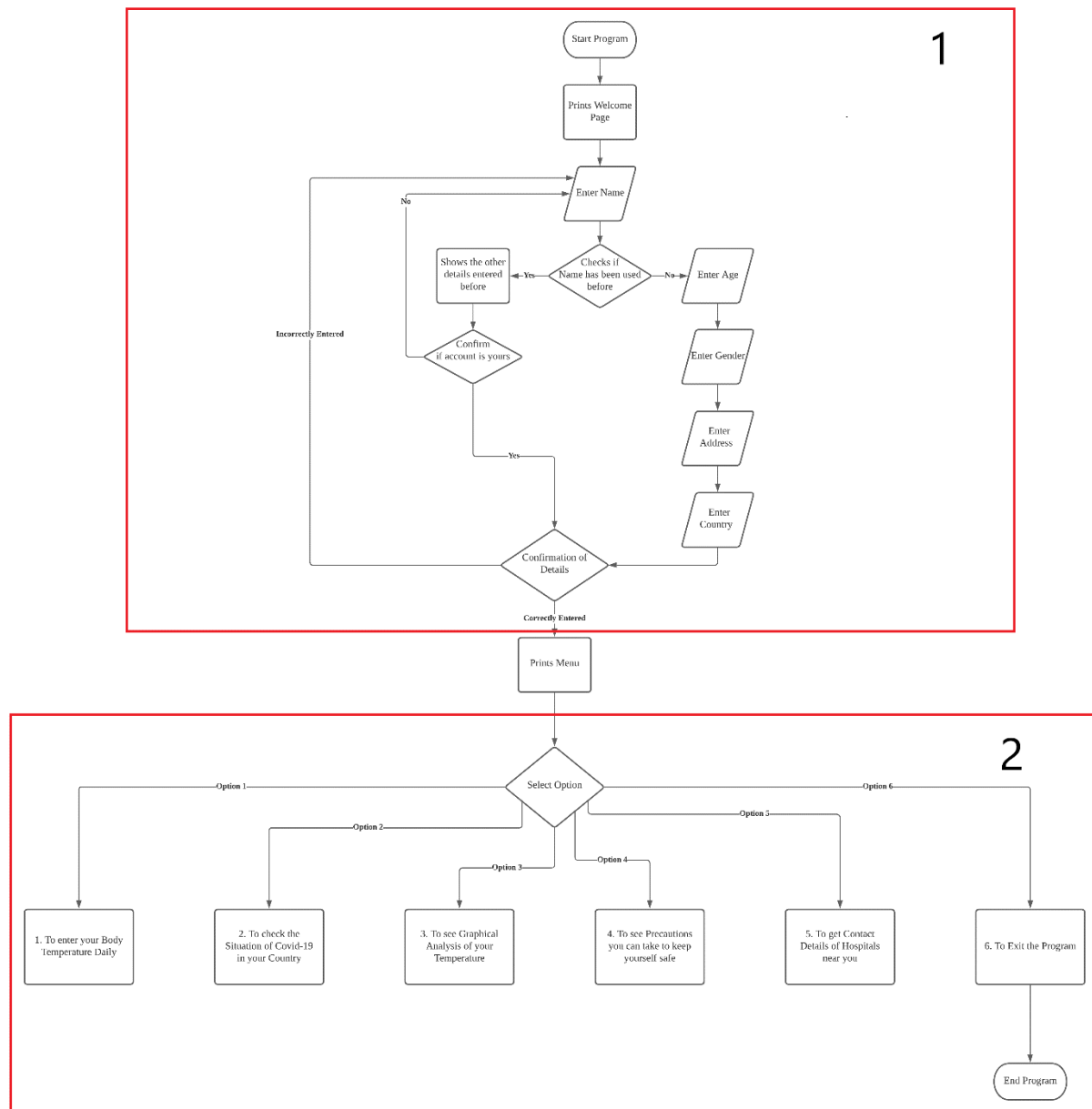
The following software is necessary to run the program:

- a. Anaconda3 (to run python, and it comes built in with the necessary modules)
- b. MySQL Workbench (to maintain a database of temperatures)
- c. Chrome Browser
- d. Chrome Webdriver (to automate Chrome Browser)

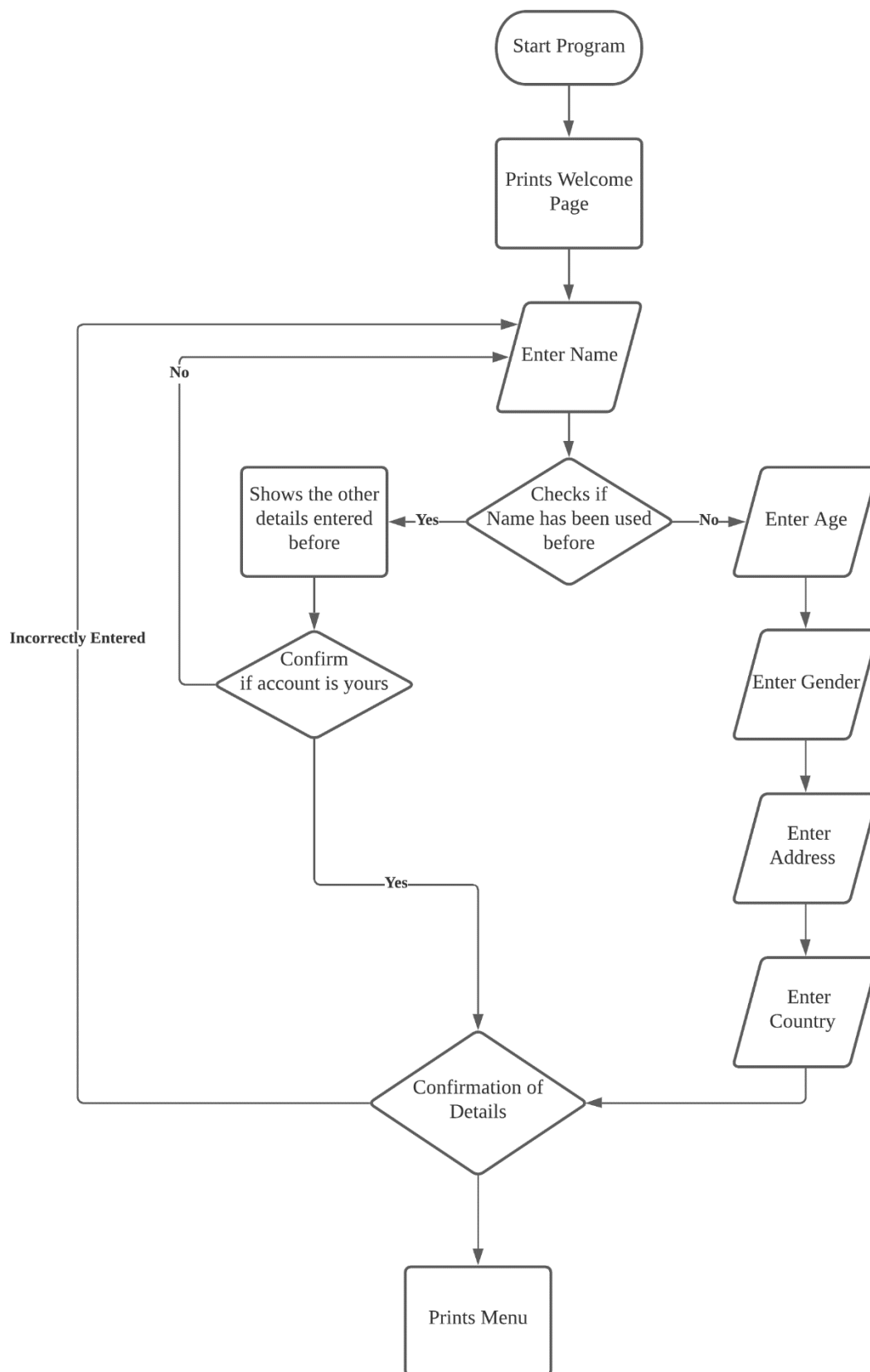
4. User:

The user should have the basic knowledge required to run a computer with Windows operating system. The user can (if needed) make the program run at a particular time of the day, with the help of Task Scheduler.

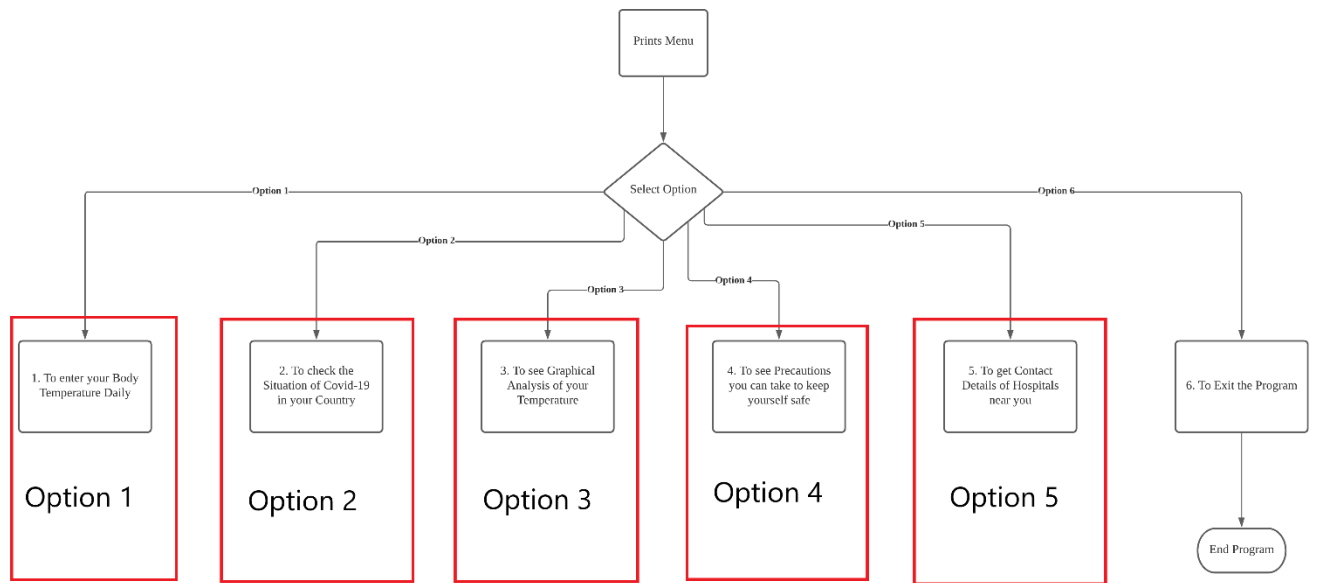
6. ALGORITHM AND FLOWCHART



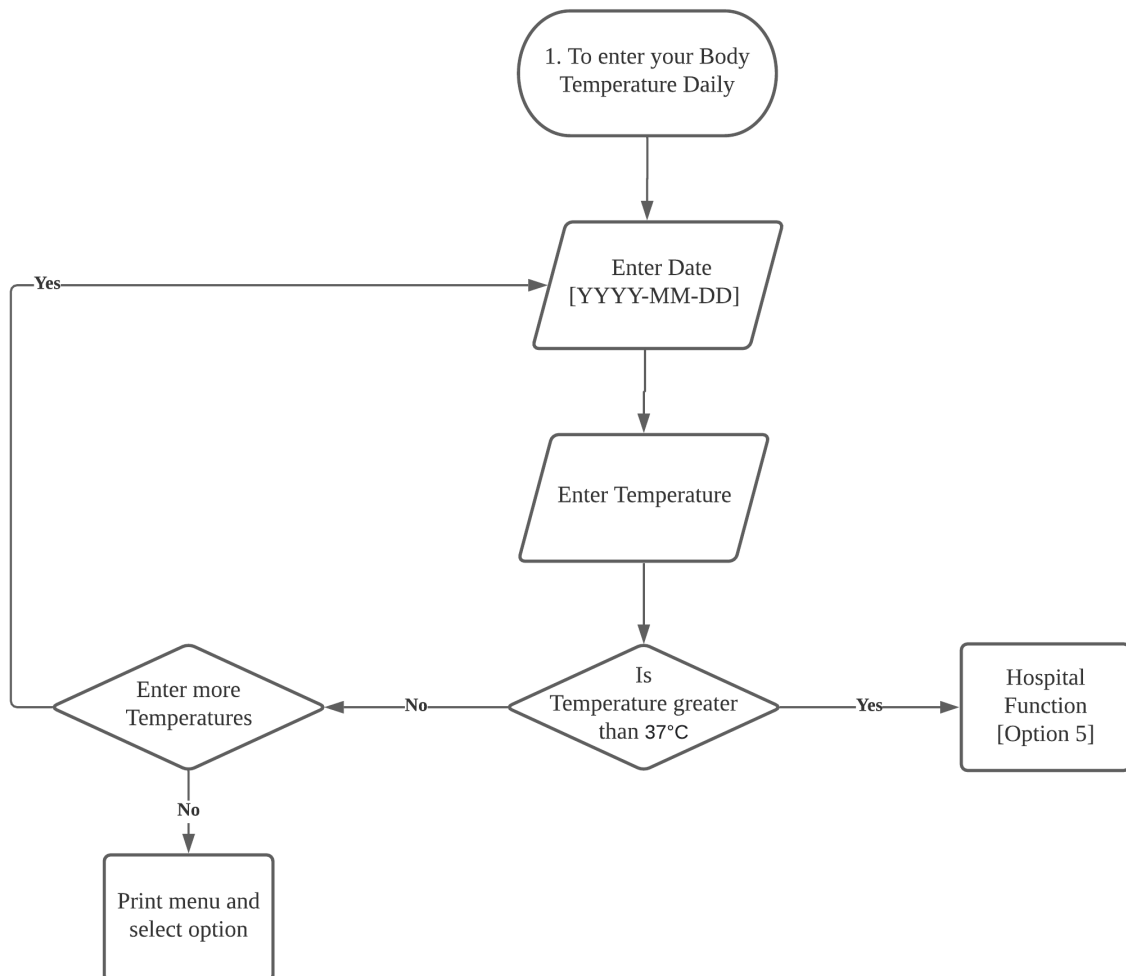
1:



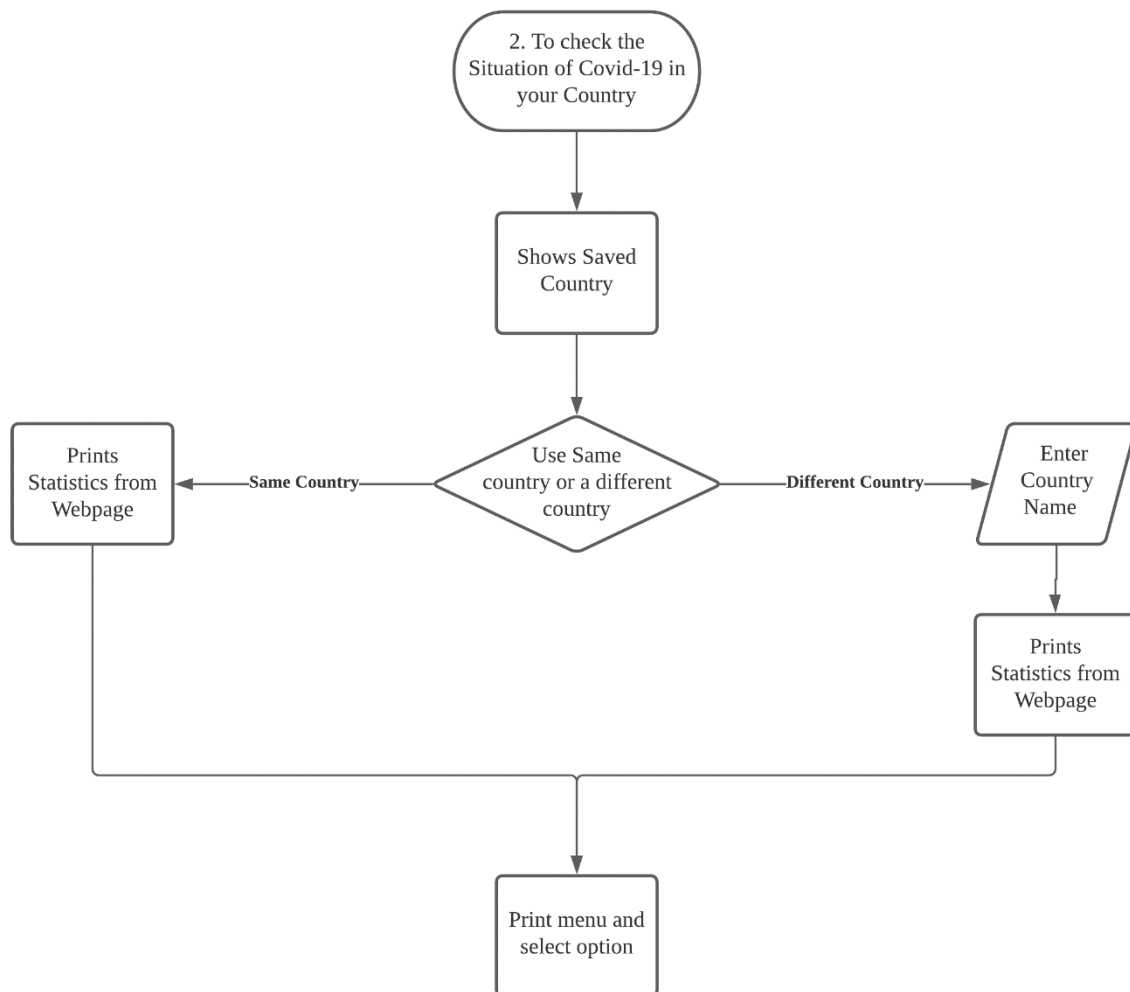
2:



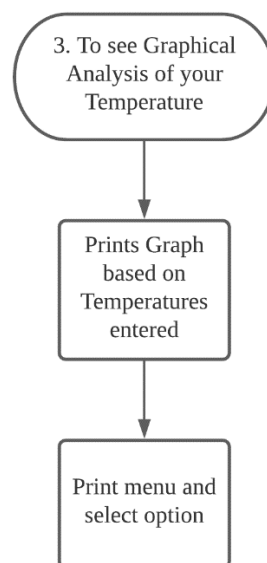
Option 1:



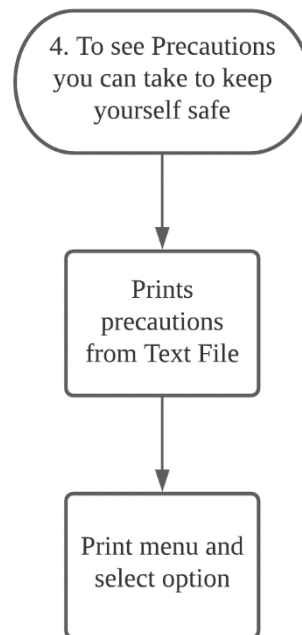
Option 2:



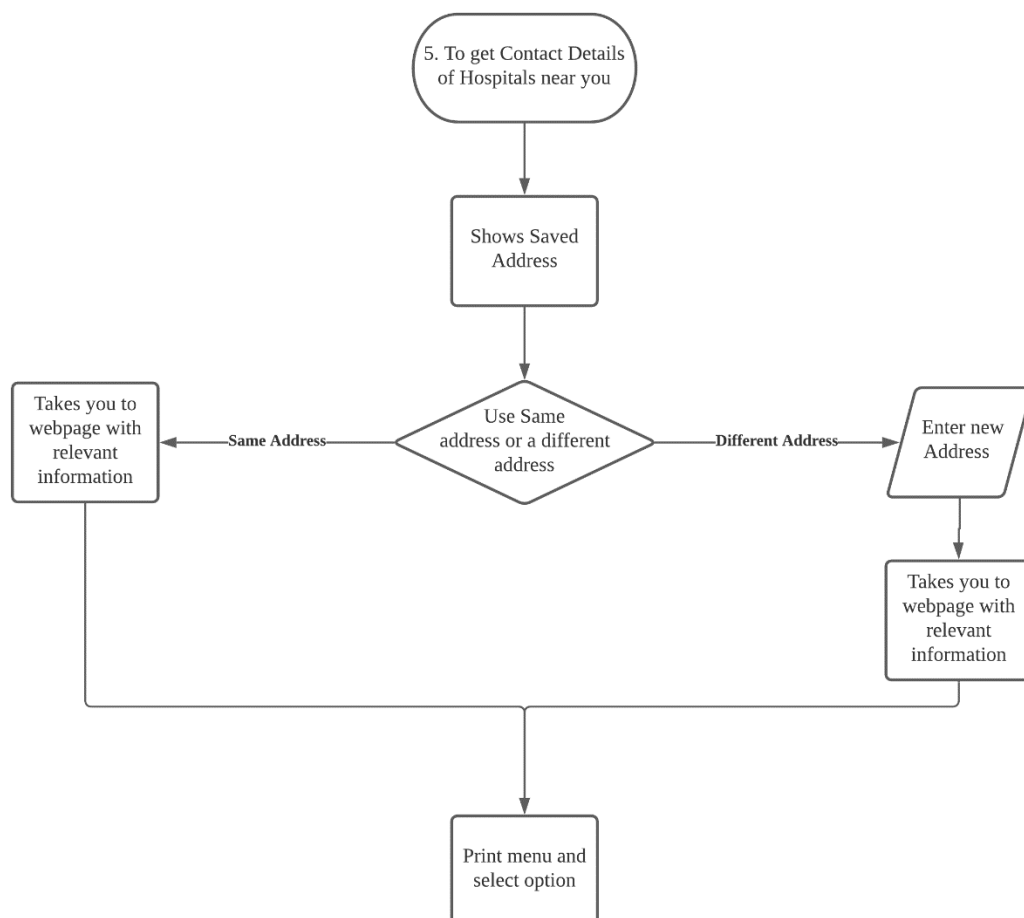
Option 3:



Option 4:



Option 5:



7. CODE LISTING

```

1.
2. ''' CS PROJECT - NAGANANDANA NAGENDRA AND JASH VERAGIWALA'''
3.
4. ''' PYTHON HEALTH TRACKER'''
5.
6. import time
7. import colorama
8. from colorama import Fore, Back, Style
9. colorama.init(autoreset=True)
10. import sys
11.
12.
13. def dprint(s):
14.     for c in s:
15.         sys.stdout.write(c)
16.         sys.stdout.flush()
17.         time.sleep(0.10)
18.
19. print(Fore.GREEN + "-----")
20.
21. img = [
22.     "
23.     "
24.     "
25.     "
26.     "
27.     "
28.     "
29.     "
30.     "
31.     "
32.     "

```



```

73.     MAKE SURE TO ENTER THE DETAILS EXACTLY AS GIVEN IN YOUR
PASSPORT. '')
74.     print()
75.     print(Fore.GREEN + "-----")
-----")
76.
77.
78.     all_name=[]
79.     all_age=[]
80.     all_gender=[]
81.     all_place=[]
82.     all_disability=[]
83.     all_country=[]
84.
85.
86.     def name():
87.         global name_user
88.         name_user=input("  ENTER YOUR NAME: ")
89.
90.
91.     def name_check():
92.         f = open(name_user+".txt","a")
93.         fh = open(name_user+".txt","r")
94.         line = fh.read()
95.         if len(line)>0:
96.             print(" ")
97.             dprint('' WE HAVE NOTICED THAT YOU HAVE USED THE PROGRAM
BEFORE.
98. ARE THESE YOUR DETAILS? '')
99.             print()
100.            ch = open(name_user+".txt", "r")
101.            y = ch.readlines()
102.            for i in y:
103.                if "NAME: " in i:
104.                    print(" ")
105.                    n = i[6:-1]
106.                    print("    NAME        |", n)
107.                    time.sleep(0.5)
108.                if "AGE: " in i:
109.                    n = i[5:-1]
110.                    print("    AGE          |", n)
111.                    time.sleep(0.5)
112.                if "GENDER: " in i:
113.                    n = i[8:-1]
114.                    if n.lower() == "m":
115.                        print("    GENDER      | Male")
116.                        time.sleep(0.5)
117.                    else:
118.                        print("    GENDER      | Female")
119.                        time.sleep(0.5)
120.                if "ADDRESS: " in i:
121.                    n = i[9:-1]
122.                    print("    ADDRESS     |", n)
123.                    time.sleep(0.5)
124.                if "COUNTRY: " in i:
125.                    n = i[9:-1]
126.                    print("    COUNTRY     |", n)
127.                    print(" ")
128.                dprint("  IF YES, ENTER 'Y', ELSE ENTER 'N': ")
129.                s = input()
130.                if s.lower() == "y":

```



```

131.         global a
132.         a = 0
133.         hh = open(name_user+".txt", "r")
134.         l = hh.readlines()
135.         for i in l:
136.             if "AGE: " in i:
137.                 global age_user
138.                 age_user = i[5:-1]
139.             if "GENDER: " in i:
140.                 if i[8:-1] == "m":
141.                     global gender_user
142.                     gender_user = "m"
143.                 else:
144.                     gender_user = "f"
145.             if "ADDRESS: " in i:
146.                 global place_user
147.                 place_user = i[9:-1]
148.             if "COUNTRY: " in i:
149.                 global country_user
150.                 country_user = i[9:-1]
151.         hh.close()
152.         ch.close()
153.     else:
154.         a = 1
155.         pass
156. else:
157.     f.close()
158.     fh.close()
159.     a = 1
160.     pass
161.
162.
163. def age():
164.     global age_user
165.     age_user=int(input("  ENTER YOUR AGE: "))
166.     if age_user <110 or age_user>0:
167.         pass
168.     else:
169.         age()
170.
171.
172. def gender():
173.     global gender_user
174.     gender_user = input("  ENTER YOUR GENDER, M FOR MALE, F FOR
FEMALE: ")
175.     if gender_user.lower() == "m" or gender_user == "f":
176.         pass
177.     else:
178.         print("  PLEASE ENTER YOUR GENDER CORRECTLY.")
179.         gender()
180.
181.
182. def place():
183.     global place_user
184.     place_user=input("  ENTER YOUR CURRENT ADDRESS: ")
185.
186.
187. def country():
188.     global country_user
189.     country_user=input("  ENTER THE COUNTRY YOU LIVE IN: ")
190.

```

```

191.
192. def date():
193.     dprint(" ENTER TODAY'S DATE [FORMAT: YYYY-MM-DD]: ")
194.     dat=input()
195.     if dat[4]!="-" and dat[7]!="-" and len(dat)!=10:
196.         print(" YOU ARE USING THE WRONG FORMAT. TRY AGAIN [FORMAT:
            YYYY-MM-DD].")
197.         date()
198.     else:
199.         return dat
200.
201.
202. def hospital():
203.     import selenium
204.     from selenium import webdriver
205.     dprint(" YOUR CURRENT ADDRESS IS SAVED AS: " + place_user)
206.     print()
207.     dprint('' DO YOU WANT TO FIND HOSPITALS NEAR A DIFFERENT
            LOCATION OR THE ONE GIVEN ABOVE?
208. TO ENTER A DIFFERENT LOCATION, ENTER 'Y' ELSE ENTER 'N': '')
209.     hos = input()
210.     if hos.lower() == 'y':
211.         hospital = input(" ENTER YOUR DESIRED LOCATION: ")
212.     else:
213.         hospital = place_user
214.     print(" ")
215.     print(Fore.RED + " YOU WILL BE REDIRECTED TO A WEBSITE.")
216.     print(Fore.RED + " DO NOT CLICK ANYTHING ON THE POP-UP UNTIL
            THE DESIRED INFORMATION HAS APPEARED.")
217.     print(" ")
218.     time.sleep(5)
219.     chrome = webdriver.Chrome("C:\\Users\\nagan\\Desktop\\Coding
            Files\\chromedriver\\chromedriver") # change this if needed
220.     chrome.get("https://www.google.co.in/")
221.     time.sleep(2)
222.     bar = chrome.find_element_by_xpath('//*[@type="text"]')
223.     time.sleep(1)
224.     bar.send_keys("Hospitals near ", hospital)
225.     bar.submit()
226.
227.
228. name()
229. name_check()
230.
231. if a == 1:
232.     print("")
233.     age()
234.     print("")
235.     gender()
236.     print("")
237.     place()
238.     print("")
239.     country()
240.     all_name.append(name_user)
241.     all_age.append(age_user)
242.     all_gender.append(gender_user)
243.     all_place.append(place_user)
244.     all_country.append(country_user)
245.     fileopen=open(name_user+".txt","w")
246.     age=str(age_user)
247.     fileopen.writelines(["NAME: "+name_user+"\n"])

```

```

248.     fileopen.writelines(["AGE: "+age+"\n"])
249.     fileopen.writelines(["GENDER: "+gender_user+"\n"])
250.     fileopen.writelines(["ADDRESS: "+place_user+"\n"])
251.     fileopen.writelines(["COUNTRY: "+country_user+"\n"])
252.     fh=open(name_user+".csv","a")
253.     fh.writelines(["date\t","temperature","\n"])
254.     fh.close()
255.     fileopen.close()
256.
257. if gender_user.lower() == "m":
258.     v = "Male"
259. else:
260.     v = "Female"
261. print(Fore.GREEN + "-----")
    print("-----")
262. dprint("  THANK YOU FOR FILLING UP YOUR DETAILS")
263. print()
264. dprint("  TO CONFIRM, YOUR DETAILS ARE AS FOLLOWS:")
265. print()
266. print(" ")
267. print("    NAME      |", name_user)
268. time.sleep(0.5)
269. print("    AGE       |", age_user)
270. time.sleep(0.5)
271. print("    GENDER    |", v)
272. time.sleep(0.5)
273. print("    ADDRESS   |", place_user)
274. time.sleep(0.5)
275. print("    COUNTRY   |", country_user)
276. print(" ")
277. dprint("  IF YOUR DETAILS ARE CORRECT ENTER 'Y', ELSE 'N': ")
278. ans = input()
279. if ans.lower() == "y":
280.     pass
281. else:
282.     main_function()
283.
284. x=0
285. while x==0:
286.     print(Fore.GREEN + "-----")
        print("-----")
287.     print("
        _____")
        print("
        _____")
288.     time.sleep(0.5)
289.     print("      |  SERIAL NO.  |  TASK
        |")
290.     time.sleep(0.5)
291.     print("      |              |
        |")
292.     time.sleep(0.5)
293.     print("      |          1      |  TO ENTER YOUR BODY TEMPERATURE
        DAILY.          |")
294.     time.sleep(0.5)
295.     print("      |              |
        |")
296.     time.sleep(0.5)
297.     print("      |          2      |  TO CHECK THE SITUATION OF COVID-19
        IN YOUR COUNTRY.  |")
298.     time.sleep(0.5)

```

```

299.     print("      |      |
      |")
300.     time.sleep(0.5)
301.     print("      |      3      | TO SEE GRAPHICAL ANALYSIS OF YOUR
TEMPERATURE.      |")
302.     time.sleep(0.5)
303.     print("      |      |
      |")
304.     time.sleep(0.5)
305.     print("      |      4      | TO SEE PRECAUTIONS THAT YOU CAN
TAKE TO KEEP YOURSELF SAFE      |")
306.     time.sleep(0.5)
307.     print("      |      |
      |")
308.     time.sleep(0.5)
309.     print("      |      5      | TO GET CONTACT DETAILS OF
HOSPITALS NEAR YOU      |")
310.     time.sleep(0.5)
311.     print("      |      |
      |")
312.     time.sleep(0.5)
313.     print("      |      6      | TO EXIT THE PROGRAM
      |")
314.     time.sleep(0.5)
315.     print("
      |")
-----
316.     print(" ")
317.     dprint('''' FOLLOWING ARE THE TASKS THIS PROGRAM PERFORMS.
318. TO SELECT A TASK, ENTER THE SERIAL NO. OF THE TASK: ''')
319.     b = int(input())
320.     print(Fore.GREEN + "-----
-----")
321.     if b==1:
322.         z=1
323.         while z==1:
324.             fh=open(name_user+".csv","a")
325.             d= date()
326.             dprint(" ENTER YOUR TEMPERATURE IN CELSIUS [°C]: ")
327.             temp = input()
328.             float_temp=float(temp)
329.             if float_temp > 37:
330.                 import colorama
331.                 from colorama import Fore, Back, Style
332.                 colorama.init(autoreset=True)
333.                 import mysql.connector as sqltor
334.
mycon=sqltor.connect(host="localhost",user="root",password="12345678"
,database="project") # change this if needed
335.                 cursor=mycon.cursor()
336.                 st="insert into tempdet
values('{}',{})".format(d,temp)
337.                 cursor.execute(st)
338.                 mycon.commit()
339.                 fh.writelines([d,"\t",temp,"\n"])
340.                 print(" ")
341.                 dprint(" ATTENTION!")
342.                 print()
343.                 print(" ")
344.                 print(Fore.RED + " YOU HAVE A FEVER.")

```

```

345.             print(Fore.RED + "  THIS IS A SYMPTOM SHOWN BY
COVID-19 PATIENTS.")
346.             print(Fore.RED + "  PLEASE VISIT A DOCTOR
IMMEDIATELY.")
347.             print(" ")
348.             dprint("  THE DETAILS OF THE NEAREST HOSPITAL AND
DIRECTIONS TO IT WILL BE SHOWN TO YOUR SHORTLY.")
349.             print()
350.             hospital()
351.             z=0
352.         else:
353.             import mysql.connector as sqltor
354.             mycon=sqltor.connect(host="localhost",user="root",password="12345678"
,database="project") # change this if needed
355.             cursor=mycon.cursor()
356.             st="insert into tempdet
values('{}',{})".format(d,temp)
357.             cursor.execute(st)
358.             mycon.commit()
359.             fh.writelines([d,"\t",temp,"\n"])
360.             dprint('''  WOULD YOU LIKE TO ENTER TEMPERATURES
FOR A DIFFERENT DAY?
361.             IF YES, ENTER 'Y' ELSE ENTER 'N': ''')
362.             h = input()
363.             if h.lower() == "y":
364.                 z=1
365.             else:
366.                 z=0
367.
368.
369.             mycon.close()
370.             fh.close()
371.             x=0
372.
373.
374.         if b==3:
375.             import numpy as np
376.             import matplotlib.pyplot as plot
377.             import datetime
378.             all_dates=[]
379.             all_temps=[]
380.             import mysql.connector as sqltor
381.             mycon=sqltor.connect(host="localhost",user="root",password="12345678"
,database="project") # change this if needed
382.             cursor=mycon.cursor()
383.             cursor.execute("select * from tempdet")
384.             data=cursor.fetchall()
385.             for i in data:
386.                 all_temps+=[i[0]]
387.                 all_dates+=[i[1]]
388.             mycon.close()
389.             plot.xlabel(' date')
390.             plot.ylabel(' Temperature')
391.             plot.plot(all_temps,all_dates)
392.             dprint('''  YOUR TEMPERATURES WILL BE AVAILABLE IN THE FORM
OF A GRAPH.
393.             YOU CAN DOWNLOAD THE GRAPH IF NEEDED.'''')
394.             print()
395.             time.sleep(5)

```

```

396.         plot.show()
397.         x=0
398.
399.
400.     if b==5:
401.         hospital()
402.         x=0
403.
404.
405.     if b==4:
406.         fh=open("precautions.txt","r")
407.         Str=fh.readlines()
408.         for i in Str:
409.             print(i,end=" ")
410.         print()
411.         time.sleep(10)
412.         fh.close()
413.         x=0
414.
415.
416.     if b==2:
417.         from selenium import webdriver
418.         import datetime
419.         import time
420.         import colorama
421.         from colorama import Fore, Back, Style
422.         colorama.init(autoreset=True)
423.         now = datetime.datetime.now()
424.
425.         dprint("  YOUR COUNTRY IS SAVED AS: " + country_user)
426.         print()
427.         dprint('''' DO YOU WANT TO KNOW THE COVID-19 STATISTICS IN
YOUR COUNTRY
428. OR IN A DIFFERENT COUNTRY?
429. TO ENTER A DIFFERENT COUNTRY, ENTER 'Y' ELSE ENTER 'N': ''')
430.         hos = input()
431.         if hos.lower() == 'y':
432.             country = input("  ENTER YOUR DESIRED COUNTRY: ")
433.         else:
434.             country = country_user
435.         print(" ")
436.         print(Fore.RED + "  YOU WILL BE REDIRECTED TO A WEBSITE,
KINDLY CLICK BACK ONTO THE PROGRAM.")
437.         print(Fore.RED + "  DO NOT CLOSE THE POP-UP. KINDLY
'MINIMIZE' THE POP-UP OPENED.")
438.         print(" ")
439.         time.sleep(5)
440.         dprint("  PLEASE WAIT FOR A FEW SECONDS AS WE GET YOU THE
LATEST STATISTICS.")
441.         print()
442.
443.
444.         chrome =
webdriver.Chrome("C:\\Users\\nagan\\Desktop\\Coding
Files\\chromedriver\\chromedriver") # change this if needed
445.
446.         chrome.get("https://www.worldometers.info/coronavirus/")
447.         time.sleep(5)
448.         chrome.execute_script("window.scrollTo(0,
document.body.scrollHeight/9);")
449.         time.sleep(5)

```

```

450.         search =
chrome.find_element_by_xpath('//input[@type="search"]')
451.         search.click()
452.         search.send_keys(country)
453.
454.         country_name =
chrome.find_element_by_css_selector('a.mt_a')
455.         total_cases = chrome.find_element_by_xpath('//td[3]')
456.         new_cases = chrome.find_element_by_xpath('//td[4]')
457.         total_deaths = chrome.find_element_by_xpath('//td[5]')
458.         new_deaths = chrome.find_element_by_xpath('//td[6]')
459.         total_recovered = chrome.find_element_by_xpath('//td[7]')
460.         active_cases = chrome.find_element_by_xpath('//td[8]')
461.         critical_cases = chrome.find_element_by_xpath('//td[9]')
462.
463.         print(" ")
464.         print(" _____ ")
465.         print(" | Country Name |",
country_name.text)
466.         print(" | |")
467.         print(" | Total cases |",
total_cases.text)
468.         print(" | |")
469.         print(" | New Cases in last 24 hours |",
new_cases.text)
470.         print(" | |")
471.         print(" | Total Deaths |",
total_deaths.text)
472.         print(" | |")
473.         print(" | New Deaths in last 24 hours |",
new_deaths.text)
474.         print(" | |")
475.         print(" | Total Recovered |",
total_recovered.text)
476.         print(" | |")
477.         print(" | Active Cases |",
active_cases.text)
478.         print(" | |")
479.         print(" | Critical Cases |",
critical_cases.text)
480.         print(" _____ ")
481.         time.sleep(10)
482.         chrome.close()
483.         x=0
484.
485.
486.         if b==6:
487.             dprint('' THANK YOU FOR USING OUR PROGRAM.
488. YOUR HEALTH IS OUR NUMBER ONE PRIORITY.
489. STAY SAFE, HEALTHY, AND ALWAYS WEAR A MASK.'')
490.             x=1
491.
492.
493.

```

8. USER DOCUMENTATION

The program prints the Welcome Page:



If the Name is not used before, the program goes as follows:

```
ENTER YOUR NAME: Jash Veragiwala
ENTER YOUR AGE: 17
ENTER YOUR GENDER, M FOR MALE, F FOR FEMALE: m
ENTER YOUR CURRENT ADDRESS: 172 Racecourse Road, Farrer Park
ENTER THE COUNTRY YOU LIVE IN: Singapore
-----
THANK YOU FOR FILLING UP YOUR DETAILS
TO CONFIRM, YOUR DETAILS ARE AS FOLLOWS:

NAME      | Jash Veragiwala
AGE       | 17
GENDER    | Male
ADDRESS   | 172 Racecourse Road, Farrer Park
COUNTRY   | Singapore

IF YOUR DETAILS ARE CORRECT ENTER 'Y', ELSE 'N':
```

If the Name is used before to create an account, the program goes as follows:

```
ENTER YOUR NAME: Jash Veragiwala

WE HAVE NOTICED THAT YOU HAVE USED THE PROGRAM BEFORE.
ARE THESE YOUR DETAILS?

NAME      | Jash Veragiwala
AGE       | 17
GENDER    | Male
ADDRESS   | 172 Racecourse Road, Farrer Park
COUNTRY   | Singapore

IF YES, ENTER 'Y', ELSE ENTER 'N': y
-----
THANK YOU FOR FILLING UP YOUR DETAILS
TO CONFIRM, YOUR DETAILS ARE AS FOLLOWS:

NAME      | Jash Veragiwala
AGE       | 17
GENDER    | Male
ADDRESS   | 172 Racecourse Road, Farrer Park
COUNTRY   | Singapore

IF YOUR DETAILS ARE CORRECT ENTER 'Y', ELSE 'N':
```

Once you confirm the details, the program prints the Menu (Note: If you enter 'no' during the confirmation of details, the program goes through the above steps again):

```
IF YOUR DETAILS ARE CORRECT ENTER 'Y', ELSE 'N': y
-----
  SERIAL NO. | TASK
  ---
  1          | TO ENTER YOUR BODY TEMPERATURE DAILY.
  2          | TO CHECK THE SITUATION OF COVID-19 IN YOUR COUNTRY.
  3          | TO SEE GRAPHICAL ANALYSIS OF YOUR TEMPERATURE.
  4          | TO SEE PRECAUTIONS THAT YOU CAN TAKE TO KEEP YOURSELF SAFE
  5          | TO GET CONTACT DETAILS OF HOSPITALS NEAR YOU
  6          | TO EXIT THE PROGRAM
  ---
FOLLOWING ARE THE TASKS THIS PROGRAM PERFORMS.
TO SELECT A TASK, ENTER THE SERIAL NO. OF THE TASK:
```

Task 1: To enter your body temperature daily:

```
FOLLOWING ARE THE TASKS THIS PROGRAM PERFORMS.
TO SELECT A TASK, ENTER THE SERIAL NO. OF THE TASK: 1
-----
ENTER TODAY'S DATE [FORMAT: YYYY-MM-DD]: 2021-02-07
ENTER YOUR TEMPERATURE IN CELSIUS [°C]: 36.2
WOULD YOU LIKE TO ENTER TEMPERATURES FOR A DIFFERENT DAY?
IF YES, ENTER 'Y' ELSE ENTER 'N':
```

If you enter 'Y' (to enter temperatures for a different day) the program prints the above lines again, so that the user can input the new data.

If the temperature entered by the user is above 37°C, the following happens:

```
IF YES, ENTER 'Y' ELSE ENTER 'N': y
ENTER TODAY'S DATE [FORMAT: YYYY-MM-DD]: 2021-02-08
ENTER YOUR TEMPERATURE IN CELSIUS [°C]: 37.2

ATTENTION!

YOU HAVE A FEVER.
THIS IS A SYMPTOM SHOWN BY COVID-19 PATIENTS.
PLEASE VISIT A DOCTOR IMMEDIATELY.

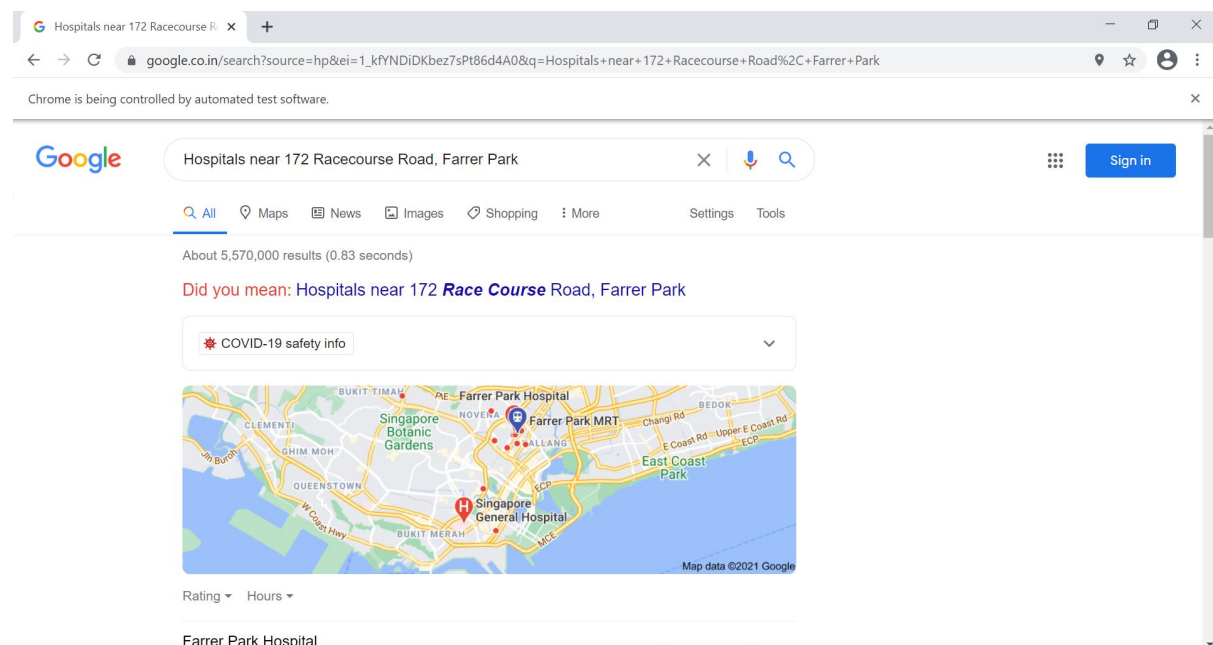
THE DETAILS OF THE NEAREST HOSPITAL AND DIRECTIONS TO IT WILL BE SHOWN TO YOUR SHORTLY.
YOUR CURRENT ADDRESS IS SAVED AS: 172 Racecourse Road, Farrer Park
DO YOU WANT TO FIND HOSPITALS NEAR A DIFFERENT LOCATION OR THE ONE GIVEN ABOVE?
TO ENTER A DIFFERENT LOCATION, ENTER 'Y' ELSE ENTER 'N': n

YOU WILL BE REDIRECTED TO A WEBSITE.
DO NOT CLICK ANYTHING ON THE POP-UP UNTIL THE DESIRED INFORMATION HAS APPEARED.

DevTools listening on ws://127.0.0.1:52402/devtools/browser/9f11958e-f290-4d86-9cbb-34473db3328d
```

The last line is the selenium module contacting the web browser. (Not applicable to the user).

The webpage the user receives is as follows:



The Hospital module is covered in detail later.

Task 2: To check the situation of Covid-19 in your country:

```
FOLLOWING ARE THE TASKS THIS PROGRAM PERFORMS.  
TO SELECT A TASK, ENTER THE SERIAL NO. OF THE TASK: 2  
-----  
YOUR COUNTRY IS SAVED AS: Singapore  
DO YOU WANT TO KNOW THE COVID-19 STATISTICS IN YOUR COUNTRY  
OR IN A DIFFERENT COUNTRY?  
TO ENTER A DIFFERENT COUNTRY, ENTER 'Y' ELSE ENTER 'N':
```

If you enter 'N' (to go with the same country as you entered earlier), the program prints the statistics of that country.

If you enter 'Y', the following happens:

```
TO ENTER A DIFFERENT COUNTRY, ENTER 'Y' ELSE ENTER 'N': y  
ENTER YOUR DESIRED COUNTRY: India  
  
YOU WILL BE REDIRECTED TO A WEBSITE, KINDLY CLICK BACK ONTO THE PROGRAM.  
DO NOT CLOSE THE POP-UP. KINDLY 'MINIMIZE' THE POP-UP OPENED.  
  
PLEASE WAIT FOR A FEW SECONDS AS WE GET YOU THE LATEST STATISTICS.  
  
DevTools listening on ws://127.0.0.1:52589/devtools/browser/ad836840-0f60-404e-bafb-900c5cd370  
Be  
[18936:24664:0207/224008.735:ERROR:device_event_log_impl.cc(211)] [22:40:08.735] USB: usb_devi  
ce_handle_win.cc:1049 Failed to read descriptor from node connection: A device attached to the  
system is not functioning. (0x1F)  
[25580:7312:0207/224013.720:ERROR:ssl_client_socket_impl.cc(962)] handshake failed; returned -  
1, SSL error code 1, net_error -201  
[25580:7312:0207/224013.721:ERROR:ssl_client_socket_impl.cc(962)] handshake failed; returned -  
1, SSL error code 1, net_error -201  
  
| Country Name | India |  
| Total cases | 10,828,167 |  
| New Cases in last 24 hours | +997 |  
| Total Deaths | 155,042 |  
| New Deaths in last 24 hours | +14 |  
| Total Recovered | 10,523,077 |  
| Active Cases | 150,048 |  
| Critical Cases | 8,944 |
```

Note: The lines in the middle (DevTools listening on... error code 1, net_error -201) are not applicable to the user. They are printed by the selenium module while it scrapes the webpage for information.

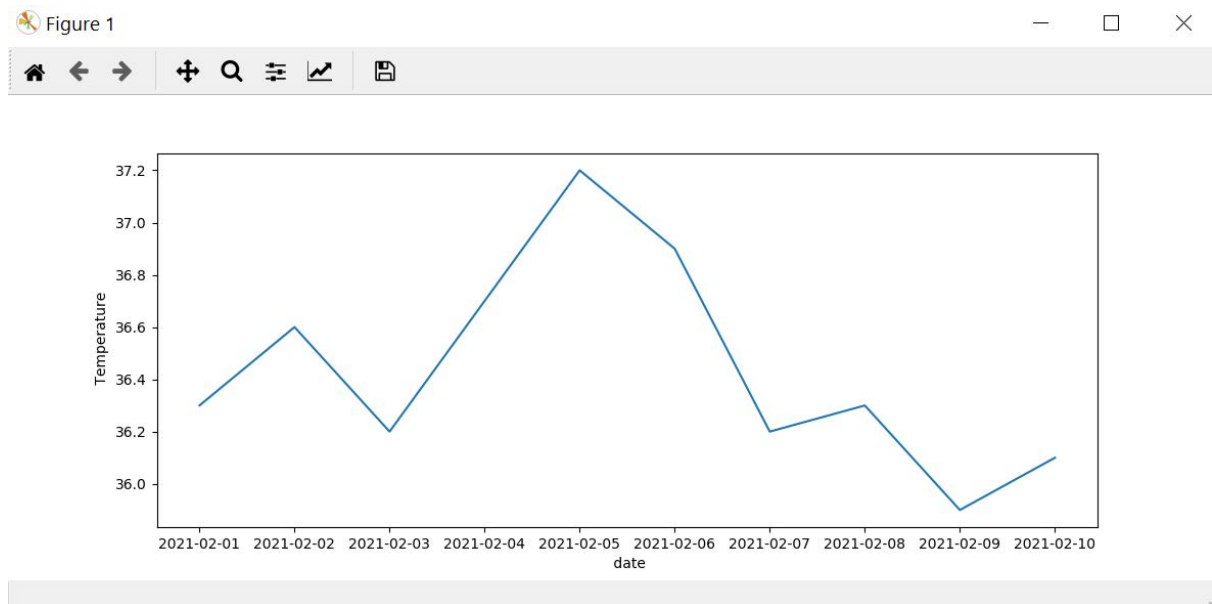
The user is asked to 'MINIMIZE' the webpage pop-up. Closing the pop-up will result in the program failing.

Task 3: To see graphical analysis of your temperature:

FOLLOWING ARE THE TASKS THIS PROGRAM PERFORMS.
TO SELECT A TASK, ENTER THE SERIAL NO. OF THE TASK: 3

YOUR TEMPERATURES WILL BE AVAILABLE IN THE FORM OF A GRAPH.
YOU CAN DOWNLOAD THE GRAPH IF NEEDED.

The Graph produced:



Task 4: To see precautions that you can take to keep yourself safe:

```
FOLLOWING ARE THE TASKS THIS PROGRAM PERFORMS.
TO SELECT A TASK, ENTER THE SERIAL NO. OF THE TASK: 4
-----
CAUSE OF COVID_19:

COVID-19 SPREADS PRIMARILY WHEN PEOPLE ARE IN CLOSE CONTACT AND ONE PERSON INHALES SMALL DROP
LETS PRODUCED BY AN INFECTED PERSON (SYMPTOMATIC OR NOT) COUGHING, SNEEZING, TALKING, OR SINGI
NG.

HOW TO PREVENT COVID-19:

1)HAND WASHING IS RECOMMENDED WITH SOAP AND WATER FOR AT LEAST TWENTY SECONDS BEFORE EATING;
AND AFTER BLOWING ONE'S NOSE, COUGHING, OR SNEEZING. OR WASHING HAND WITH ALCOHOL-BASED HAND S
ANITISER WITH AT LEAST 60 PERCENT ALCOHOL BY VOLUME WHEN SOAP AND WATER ARE NOT READILY AVAILA
BLE.

2)SURFACE CLEANING IS NECESSARY ON SURFACES LIKE ELECTRONIC EQUIPMENT LIKE TABLETS, TOUCH SCR
EENS, KEYBOARDS, REMOTE CONTROLS, AND ATM MACHINES.

3)USE FACE MASK IS YOU NEED TO GO OUT OF YOUR HOME.

4)SOCIAL DISTANCING IS NECESSARY AT ALL PUBLIC PLACES LIKE SCHOOLS, WORKPLACES, STADIUMS, THE
ATRES, OR SHOPPING CENTRES. IT CAN BE DONE BY STAYING AT HOME, LIMITING TRAVEL, AVOIDING CROWD
ED AREAS, USING NO-CONTACT GREETINGS, AND PHYSICALLY DISTANCING THEMSELVES FROM OTHERS.

5)SELF-ISOLATION AT HOME HAS BEEN RECOMMENDED FOR THOSE DIAGNOSED WITH COVID-19 AND THOSE WHO
SUSPECT THEY HAVE BEEN INFECTED.

6)TRY TO BOOST YOUR IMMUNITY BY EATING HOME FOOD AND AVOIDING OUTSIDE FOOD.
```

Task 5: To get contact details of hospitals near you:

```
FOLLOWING ARE THE TASKS THIS PROGRAM PERFORMS.
TO SELECT A TASK, ENTER THE SERIAL NO. OF THE TASK: 5
-----
YOUR CURRENT ADDRESS IS SAVED AS: 172 Racecourse Road, Farrer Park
DO YOU WANT TO FIND HOSPITALS NEAR A DIFFERENT LOCATION OR THE ONE GIVEN ABOVE?
TO ENTER A DIFFERENT LOCATION, ENTER 'Y' ELSE ENTER 'N':
```

If the user enters 'N' (to use same address) the program provides the webpage with information relevant to that address.

If the user enters 'Y', the following happens:

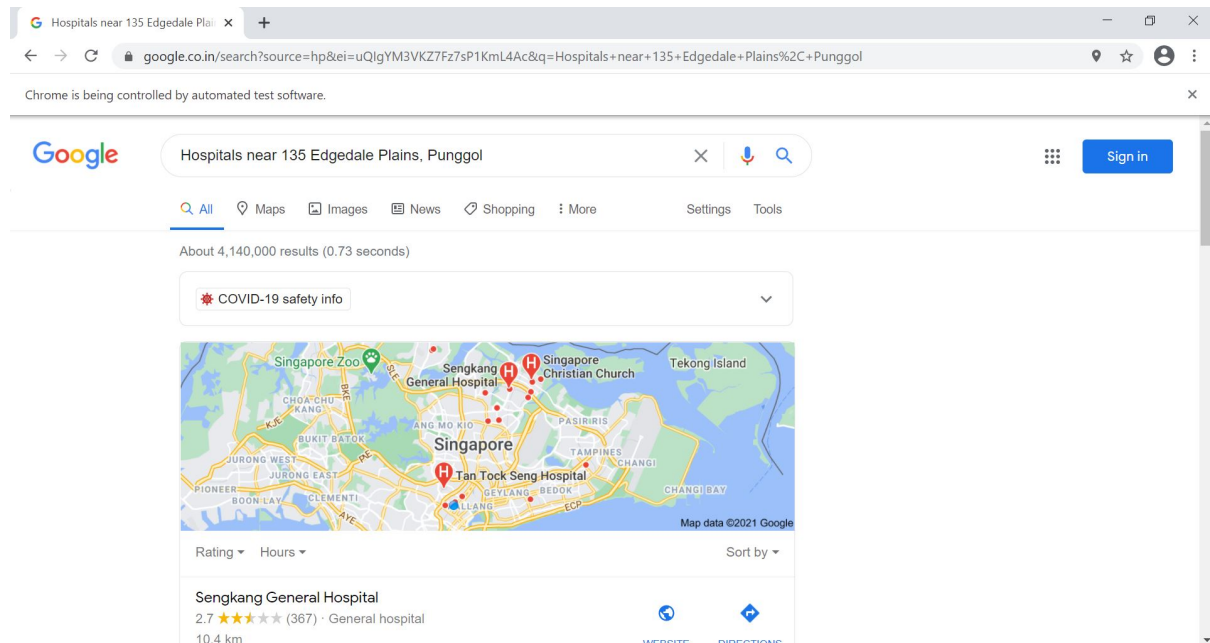
```
TO ENTER A DIFFERENT LOCATION, ENTER 'Y' ELSE ENTER 'N': y
ENTER YOUR DESIRED LOCATION: 135 Edgedale Plains, Punggol

YOU WILL BE REDIRECTED TO A WEBSITE.
DO NOT CLICK ANYTHING ON THE POP-UP UNTIL THE DESIRED INFORMATION HAS APPEARED.

DevTools listening on ws://127.0.0.1:52794/devtools/browser/26656775-e14a-4166-9cf3-ae2d2261b8
54
[14516:25104:0207/230948.921:ERROR:device_event_log_impl.cc(211)] [23:09:48.921] USB: usb_devi
ce_handle_win.cc:1049 Failed to read descriptor from node connection: A device attached to the
system is not functioning. (0x1F)
```

Note: The lines in the end (DevTools listening on... not functioning) are not applicable to the user. They are printed by the selenium module while it scrapes the webpage for information.

The webpage brought up to the user:



Task 6: To exit the program:

```
FOLLOWING ARE THE TASKS THIS PROGRAM PERFORMS.  
TO SELECT A TASK, ENTER THE SERIAL NO. OF THE TASK: 6
```

```
-----  
THANK YOU FOR USING OUR PROGRAM.  
YOUR HEALTH IS OUR NUMBER ONE PRIORITY.  
STAY SAFE, HEALTHY, AND ALWAYS WEAR A MASK.
```

9. BIBLIOGRAPHY

1. Automate the Boring Stuff with Python, by Al Sweigart.