CORONA VIRUS LIVE TRACKER

Python project

Submitting To,

Mr. Mayur Dev Sewak
General Manager, Operations
Eisystems Services
&

Ms. Mallika Srivastava
Trainer, Data Science & Analytics Domain
Eisystems Services

By,

SAI CHANDANA JAMPALA

BATCH: 21PHY-078

2. Table of Contents

3. List of images/figures

4. Nomenclature / Notations used in the project are

- We have made this project in Python and other than python other technologies we have used are Desktop GUI.
- We have also used variables which gives data to the computer for processing.

2.1 Variables Used are:-

- Title
- App-icon
- Timeout
- Res
- Soup
- Body
- Abs
- Country notification
- Header
- Totalcase1
- Total death
- New cases
- New death
- Path
- Flits
- Path
- Label1
- Label2
- Cantata
- Entry1
- Relief
- Intel
- In son
- Submit

5. Abstract of Project

- The overall project is about corona live tracker .
- In this project , by using the python programming language we have designed a code which shows corona live status of different countries such as Poland ,India etc .
- In the output we need to give the country name ,thereby it shows overall live cases ,deaths ,present cases in that country etc .
- In this project folder we will be having only 3 main things one which is mainfile.py ,the python file which is having the overall code .
- The second one is corona .ico which is having a particular corona virus image which we given as input it should be particularly in ico format .
- The third one is coronadata.csv file which is having each and every data of corona of each individual country which shows live status ,the file which is mainly help full in showing the corona live trace.
- So here in this project we have imported ,requests ,beautifulsoap and plyer packages .
- The beautiful soap package is for passing the html document and to read the file as well.
- The plyer is to get notification ,which is displayed in output .
- The main package tinker, in which we use only ico format .
- We have used cntdata.get() ; to get new updates of corona .
- We have included strip, which is used to remove spaces for a string .
- And also we have used append function here because append will increase the list order by 1.
- Even ,the data frame for proper order of data .
- Indeed , we have used button system here normal -> button state ; we need to use it with mouse and key board .
- The “bg” for normal baground colour and placed a label for displaying box for test and images .
- The relief=ridge is for certain simulated 3d figure .
- Therefore these are the main parts in the code .
- There by in the output we need to give any country name as input, therefore we get a notification about overall cases in the country and you can even download the overall report of each and every country in html,jsn or excel .

6. Project Summary

- Live tracker for corona virus cases ,deaths and everything .
- We will scrap our data from worldometer.com and to download icon ico images.
- There for the libraries used are .

- **Plyer.**
- **Bs4----for BeautifulSoup**
- **Tk---- from tkinter import.**
- **Pandas to arrange the data .**
- **Requests.**

- Now, we will make label buttons and text spaces,we have to scrap data from worldometer.
- To get text in html form we will import bs4.
- Here we want data of table so it will be named as t.body.
- For getting the data properity we will do .text.
- now, we will make list of all the columns like country ,total cases ,new cases etc.
- here we are using header because to name the column in our downloaded file .
- now we will see how to bring notifications for that you need to download .
- we will keep all our data in a function so it will be easy .
- we will keep world as default when no country is entered ,after that we make function disabled when it is clicked.

- Now we create function to download all the files .
- We can give path wherever we want to download our required file .
- Now to get our data in ascending order we will ue pandas to sort that .
- Now here we will use zip function ,so that we can store all the data to gether .
- Now to sort we will use sortfn and here we will sort according to total cases in the world .
- Now we are creating a message box .
- So now we can download all the data in html ,json,and csv file.

7. Objectives of Project

7.1 Introduction

- Corona virus widely known as COVID-19 are a large family of viruses that are known to cause illness ranging from the common cold to more severe diseases such as Middle East Respiratory Syndrome (MERS) and Severe Acute Respiratory Syndrome (SARS).
- The virus out-braked on Nov-2019 from Wuhan city of china. The virus affected south China for about a month and then it spread throughout the world making the path through Europe.
- The champion's league game between Atlanta and Valencia became the poison for spreading the virus in Europe as more than 80,000 spectators were in stadium. Till date, the virus had spread in about 235 territories affecting more than 55 million of population.
- The large number of people are being infected and killed on daily basis all over the world. And the data of death, infected and recovered cases are being provided on2 different websites, Google, WebPages etc. But these data contain the data from all over the world.
- Therefore, with the necessity to bring the data of local level with effective analysis, and to make a project work possible through virtual classes we decide to make this project.

7.2 Motivation

- National Informatics Centre posted on LinkedIn to encourage the engineers of India to build tools and services that can help provide accurate information from official sources in user-friendly manner and help spread awareness of gravity of situation.
- To develop the idea and concept of teamwork, field project and analytical skills to us.
- To study timely trend of Covid-19 in the world along with the comparative study of situations.

8. Details of Process / Project developed

In this python project, we will implement a dashboard and a coved cases report for COVID 19 spread analysis. This dashboard and repost will provide much insightful visualization for the study of corona virus spread.

So we started by building a web scrapper using Beautiful Soup, which is a very popular library for parsing simple HTML and XML documents. Python when combined with tinder provides a fast and easy way to create GUI application.

There are two main methods, which we should remember while creating the python application With GUI.

- **TK** (Sereen Name =None, base Name = None class Name =[1]TX",USE TK = 1
- **mainloop**() → mainloop() is used when our application is ready to run.

Tkinter also offers access to the geometric configuration of the widgets which can organize the Widgets in the parent Windows.

There is mainly three geometry manager class's class.

- Pack () method: - It organizes the widgets in blocks before placing in the parent widget.
- Grid () method! - It organizes the widgets in grid (table-like structure) before placing in the parent Widget.
- Place () method .

We have also used

- **Active background:** to set the background color when button is under the cursors out.
- **Active for ground**: to set the fore ground color button is under the cursor.
- **Big**: to set the normal background color.
- **Command**: to call a function.
- **Font**: to set the font on the button Label.
- **Image**: to set image on the button.
- **Width**: to set the width of the button .
- **Height**: to set the height of the button.
- **bad**: to set the border width in pixels .
- **Label** →which is responsible for implementing box-section for text & images.
- **Icon bitmap** (bitmap) sets the icon of the window/ frame widget to bitmap. The bitmap must be an icon type.
- We have also used modules and variables. When the final output is come there display a window where we have to enter the name of the country and it will display the cases details. Also we can download the files either in html,json,excel and the data from all over the country will be displayed in the files.

9. Apparatus / Components / System

- **Pc-** For creating project.
- **IDLE** - IDLE is Python's Integrated Development and Learning Environment. It allows programmers to easily write Python code.
- **Compiler** - A computer program that translates code written in one programming language into another is called a compiler. We need compiler for the smooth execution of the code.

9.1 Modules used in our project:-

- **TK-**Tk widgets can be used to construct buttons, menus, data fields, etc. in a Python application.

- **BeautifulSoup :-** Beautiful Soup is a Python library that is used for web scraping purposes to pull the data out of HTML and XML files.

- **Requests:-** The requests module allows you to send HTTP requests using Python. The HTTP request returns a Response Object with all the response data (content, encoding, status, etc).

- **Plyer:-** Plyer module is used to access the features of the hardware. This module does not comes built-in with Python.

- **Pandas:-**Pandas is an open source Python package that is most widely used for data science/data analysis and machine learning tasks. It is built on top of another package named Numpy, which provides support for multi-dimensional arrays.

10. Input and output screen shots

- code

mainfile.py

```
import requests
from bs4 import BeautifulSoup
import plyer


def datacollected():
    def notification(title, message):
        plyer.notification.notify (
            title=title,
            message=message,
            app_icon='corona.ico',
            timeout=15
        )

    url = "https://www.worldometers.info/coronavirus/"
    res = requests.get (url)
    soup = BeautifulSoup (res.content, 'html.parser')
    tbody = soup.find ('tbody')
    abc = tbody.find_all ('tr')
    countrynotification = cntdata.get ()

    if (countrynotification == ""):
        countrynotification = "world"
```

Fig 1.1 Importing requests and beautiful soap

mainfile.py

```
if (countrynotification == ""):
    countrynotification = "world"

serial_number, countries, total_cases, new_cases, total_death, new_deaths, total_recovered, active_cases = [], [], [], [], [], [], [], []
serious_critical, total_cases_permn, total_deaths_permn, total_tests, total_test_permillion, total_pop = [], [], [], [], [], []

header = ['serial_number', 'countries', 'total_cases', 'new_cases', 'total_death', 'new_deaths', 'total_recovered',
          'active_cases',
          'serious_critical', 'total_cases_permn', 'total_deaths_permn', 'total_tests', 'total_test_permillion',
          'total_pop']

for i in abc:
    id = i.find_all ('td')
    if (id[1].text.strip ().lower () == countrynotification):
        totalcases1 = int (id[2].text.strip ().replace (',', ""))
        totaldeath = id[4].text.strip ()
        newcases = id[3].text.strip ()
        newdeaths = id[5].text.strip ()
        notification ("CORONA RECENT UPDATES OF  {}".format (countrynotification),
                      "Total Cases : {}\nTotal Deaths : {}\nNew Cases : {}\nNew Deaths : {}".format (
                          totalcases1, totaldeath, newcases, newdeaths
                      ))
    serial_number.append (id[0].text.strip ())
```

Fig 1.2 Using if and for in code

mainfile.py

```
            notification ("CORONA RECENT UPDATES OF  {}".format (countrynotification),
                          "Total Cases : {}\nTotal Deaths : {}\nNew Cases : {}\nNew Deaths : {}".format (
                              totalcases1, totaldeath, newcases, newdeaths
                          ))
        serial_number.append (id[0].text.strip ())
        countries.append (id[1].text.strip ())
        total_cases.append (id[2].text.strip ().replace (',', ""))
        new_cases.append (id[3].text.strip ())
        new_deaths.append (id[5].text.strip ())
        total_death.append (id[4].text.strip ())
        total_recovered.append (id[6].text.strip ())
        active_cases.append (id[7].text.strip ())
        serious_critical.append (id[8].text.strip ())
        total_cases_permn.append (id[9].text.strip ())
        total_deaths_permn.append (id[10].text.strip ())
        total_tests.append (id[11].text.strip ())
        total_test_permillion.append (id[12].text.strip ())
        total_pop.append (id[13].text.strip ())

    dataframe = pd.DataFrame (list (zip (serial_number, countries, total_cases, new_cases, total_death,
                                         new_deaths, total_recovered, active_cases,
                                         serious_critical, total_cases_permn, total_deaths_permn, total_tests,
                                         total_test_permillion
```

Fig1.3 Using data frame and append

mainfile.py

```
    sorts = dataframe.sort_values ('total_cases', ascending=False)
    for a in flist:
        if (a == 'html'):
            path2 = '{}/coronadata.html'.format (path)
            sorts.to_html (r'{}'.format (path2))

        if (a == 'json'):
            path2 = '{}/coronadata.json'.format (path)
            sorts.to_json (r'{}'.format (path2))

        if (a == 'csv'):
            path2 = '{}/coronadata.csv'.format (path)
            sorts.to_csv (r'{}'.format (path2))

        if (len (flist) != 0):
            messagebox.showinfo ("Notification", "Corona Record is saved {}".format (path2), parent=coro)


def downloaddata():
    global path9
    if (len (flist) != 0):
        path = filedialog.askdirectory ()
```

Fig 1.4 Using if

mainfile.py

```
    if (len (flist) != 0):
        path = filedialog.askdirectory ()
    else:
        pass
    datacollected ()
    flist.clear ()
    Inhtml.configure (state='normal')
    Injson.configure (state='normal')
    Inexcel.configure (state='normal')


def inhtmldownload():
    flist.append ('html')
    Inhtml.configure (state='disabled')


def injsondownload():
    flist.append ('json')
    Injson.configure (state='disabled')


def inexceldownload():
    flist.append ('csv')
```

Fig1.5 Using if else and def

mainfile.py

```
def inexceldownload():
    flist.append ('csv')
    Inexcel.configure (state='disabled')


import pandas as pd
from tkinter import *
from tkinter import messagebox, filedialog

coro = Tk ()
coro.title ("Corona Virus Information")
coro.geometry ('800x500+200+80')
coro.configure (bg='#046173')
coro.iconbitmap ('corona.ico')
flist = []
path = ''
mainlabel = Label (coro, text="Corona Virus Live Tracker", font=("new roman", 30, "italic bold"), bg="#05897A", wid
                   , fg="black", bd=5)
mainlabel.place (x=0, y=0)

label1 = Label (coro, text="Country Name", font=("arial", 20, "italic bold"), bg="#046173")
label1.place (x=15, y=100)
```

Fig 1.6 Importing pd

mainfile.py

```
label1.place (x=15, y=100)

label2 = Label (coro, text="Download File in ", font=("arial", 20, "italic bold"), bg="#046173")
label2.place (x=15, y=200)

cntdata = StringVar ()
entry1 = Entry (coro, textvariable=cntdata, font=("arial", 20, "italic bold"), relief=RIDGE, bd=2, width=32)
entry1.place (x=280, y=100)

Inhtml = Button (coro, text="Html", bg="#2DAE9A", font=("arial", 15, "italic bold"), relief=RIDGE,
                 activebackground="#05945B",
                 activeforeground="white", bd=5, width=5, command=inhtmldownload)
Inhtml.place (x=300, y=200)

Injson = Button (coro, text="json", bg="#2DAE9A", font=("arial", 15, "italic bold"), relief=RIDGE,
                 activebackground="#05945B",
                 activeforeground="white", bd=5, width=5, command=injsondownload)
Injson.place (x=300, y=260)

Inexcel = Button (coro, text="Excel", bg="#2DAE9A", font=("arial", 15, "italic bold"), relief=RIDGE,
                  activebackground="#05945B",
                  activeforeground="white", bd=5, width=5, command=inexceldownload)
Inexcel.place (x=300, y=320)
```

Fig 1.7 Using html

mainfile.py

```
Injson = Button (coro, text="json", bg="#2DAE9A", font=("arial", 15, "italic bold"), relief=RIDGE,
                 activebackground="#05945B",
                 activeforeground="white", bd=5, width=5, command=injsondownload)
Injson.place (x=300, y=260)

Inexcel = Button (coro, text="Excel", bg="#2DAE9A", font=("arial", 15, "italic bold"), relief=RIDGE,
                  activebackground="#05945B",
                  activeforeground="white", bd=5, width=5, command=inexceldownload)
Inexcel.place (x=300, y=320)

Submit = Button (coro, text="Submit", bg="#CB054A", font=("arial", 15, "italic bold"), relief=RIDGE,
                 activebackground="#7B0519",
                 activeforeground="white", bd=5, width=25, command=downloaddata)
Submit.place (x=450, y=260)

coro.mainloop ()
```

Fig1.8 Submit button (code)

- **The image which we given as input ,it should be particulary in ico format**

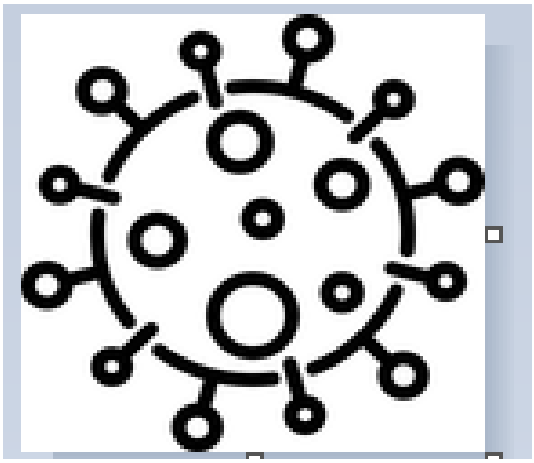

Fig 1.9 Corona.ico

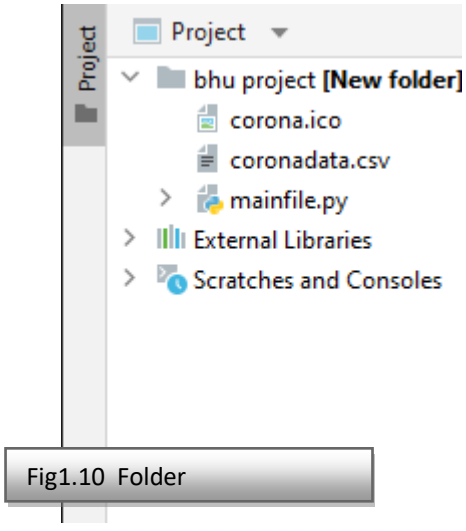

Fig1.10 Folder

Total folder

- Corona.ico represents above picture
- Coronadata.csv is the file with overall data about corona
- Mailfile.py is the python file with overall code

10.1 Output:-

- After we run the code in the pycharm, this is the output that displays ,which shows country name ,downloading file and the title as corona live tracker .
- Therefore ,There is a box or space beside country name which we need to give any country name for instance india ,Pakistan etc and u can even download overall report of all corona cases in html or json or excel format.
- After giving a country name and selecting the type of file you want to download ,click on the submit button.then you will get notification with all the updates of corona about the country you given.

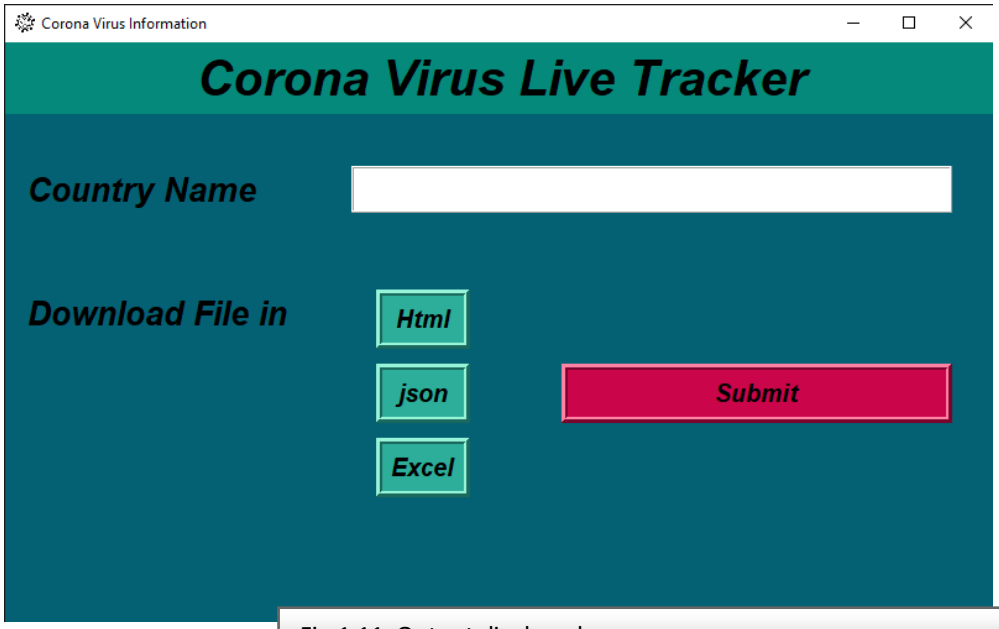

Fig 1.11 Output displayed

Here we have given the country name as Poland :-

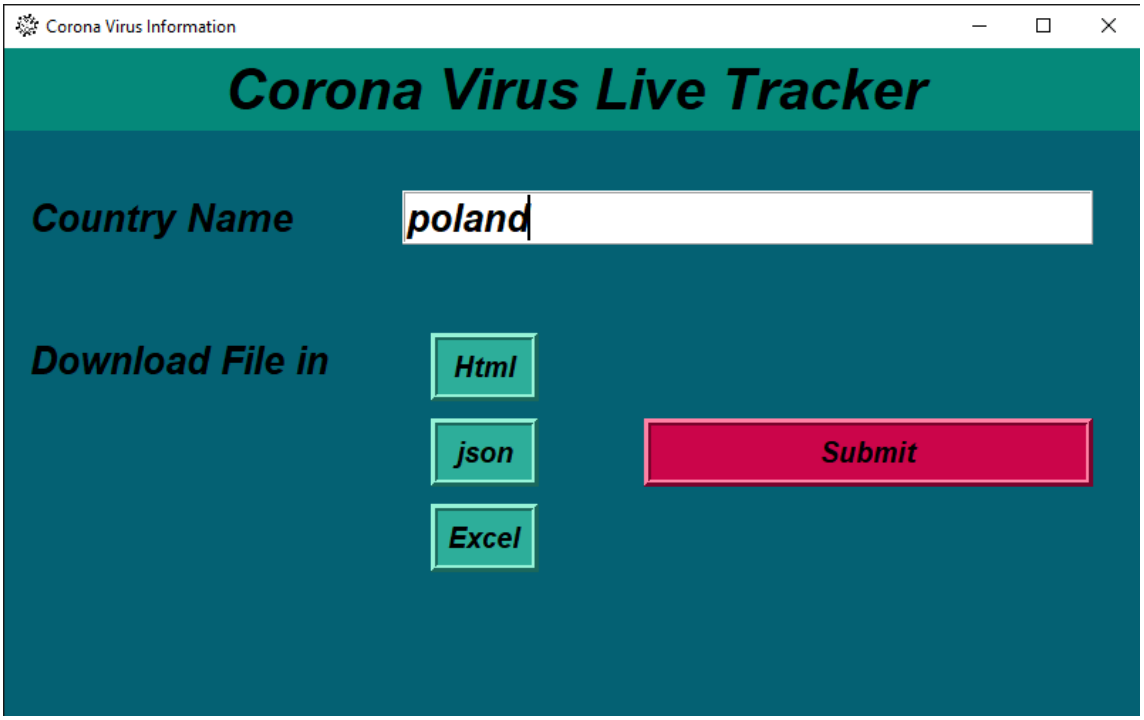

Fig 1.12 Giving input as poland

10.2 The displayed notification:-

- So as we given the input as country Poland , the notification displays the recent trends about Poland, it consists of total no of cases, total deaths ,new cases ,new deaths.

- Important

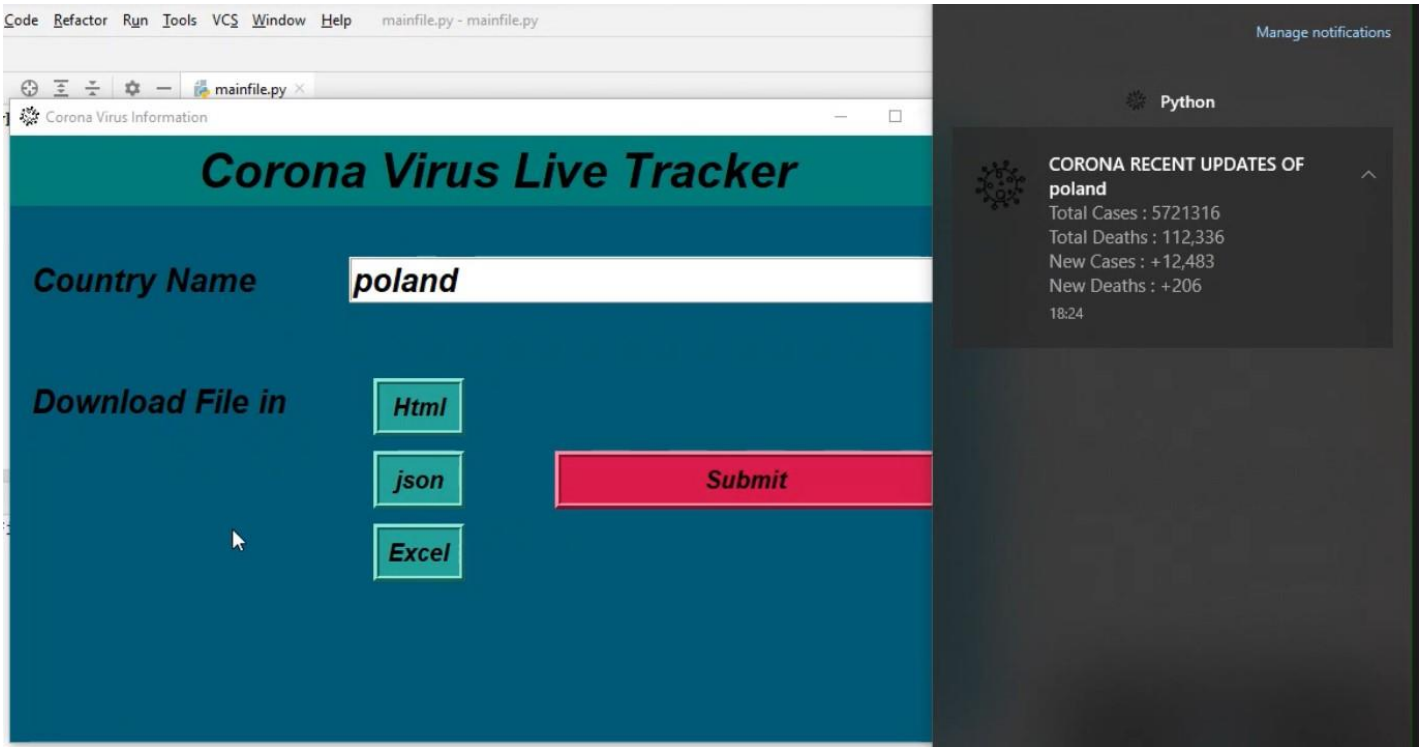

Fig 1.13 Displaying notification

11. text/code/program if used

```
import requests
from bs4 import Beautiful Soup
import plyer


def datacollected():
    def notification(title, message):
        plyer.notification.notify (
            title=title,
            message=message,
            app_icon='corona.ico',
            timeout=15
        )

    url = "https://www.worldometers.info/coronavirus/"
    res = requests.get (url)
    soup = BeautifulSoup (res.content, 'html.parser')
    tbody = soup.find ('tbody')
    abc = tbody.find_all ('tr')
    countrynotification = cntdata.get ()

    if (countrynotification == ""):
        countrynotification = "world"

    serial_number, countries, total_cases, new_cases, total_death, new_deaths,
total_recovered, active_cases = [], [], [], [], [], [], [], []
    serious_critical, total_cases_permn, total_deaths_permn, total_tests,
total_test_permillion, total_pop = [], [], [], [], [], []

    header = ['serial_number', 'countries', 'total_cases', 'new_cases', 'total_death',
'new_deaths', 'total_recovered',
              'active_cases',
              'serious_critical', 'total_cases_permn', 'total_deaths_permn',
'total_tests', 'total_test_permillion',
              'total_pop']

    for i in abc:
        id = i.find_all ('td')
        if (id[1].text.strip ().lower () == countrynotification):
            totalcases1 = int (id[2].text.strip ().replace (',', ""))
            totaldeath = id[4].text.strip ()
            newcases = id[3].text.strip ()
            newdeaths = id[5].text.strip ()
            notification ("CORONA RECENT UPDATES OF  {}".format (countrynotification),
                          "Total Cases : {}\nTotal Deaths : {}\nNew Cases : {}\nNew
Deaths : {}".format (
                              totalcases1, totaldeath, newcases, newdeaths
                          ))
        serial_number.append (id[0].text.strip ())
        countries.append (id[1].text.strip ())
        total_cases.append (id[2].text.strip ().replace (',', ""))
        new_cases.append (id[3].text.strip ())
        new_deaths.append (id[5].text.strip ())
        total_death.append (id[4].text.strip ())
        total_recovered.append (id[6].text.strip ())
        active_cases.append (id[7].text.strip ())
        serious_critical.append (id[8].text.strip ())
        total_cases_permn.append (id[9].text.strip ())
        total_deaths_permn.append (id[10].text.strip ())
        total_tests.append (id[11].text.strip ())
        total_test_permillion.append (id[12].text.strip ())
        total_pop.append (id[13].text.strip ())
```

```
dataframe = pd.DataFrame (list (zip (serial_number, countries, total_cases, new_cases, total_death,
                                     new_deaths, total_recovered, active_cases,
                                     serious_critical, total_cases_permn, total_deaths_permn, total_tests,
                                     total_test_permillion,
                                     total_pop)), columns=header)
  sorts = dataframe.sort_values ('total_cases', ascending=False)
  for a in flist:
    if (a == 'html'):
      path2 = '{}/coronadata.html'.format (path)
      sorts.to_html (r'{}'.format (path2))
    if (a == 'json'):
      path2 = '{}/coronadata.json'.format (path)
      sorts.to_json (r'{}'.format (path2))
    if (a == 'csv'):
      path2 = '{}/coronadata.csv'.format (path)
      sorts.to_csv (r'{}'.format (path2))
    if (len (flist) != 0):
      messagebox.showinfo ("Notification", "Corona Record is saved {}".format (path2), parent=coro)
def downloaddata():
  global path9
  if (len (flist) != 0):
    path = filedialog.askdirectory ()
  else:
    pass
  datacollected ()
  flist.clear ()
  Inhtml.configure (state='normal')
  Injson.configure (state='normal')
  Inexcel.configure (state='normal')
def inhtmldownload():
  flist.append ('html')
  Inhtml.configure (state='disabled')
def injsondownload():
  flist.append ('json')
  Injson.configure (state='disabled')
def inexceldownload():
  flist.append ('csv')
  Inexcel.configure (state='disabled')
import pandas as pd
from tkinter import *
from tkinter import messagebox, filedialog
coro = Tk ()
```

```
coro.title ("Corona Virus Information")



coro.geometry ('800x500+200+80')
coro.configure (bg='#046173')
coro.iconbitmap ('corona.ico')
flist = []
path = ''
mainlabel = Label (coro, text="Corona Virus Live Tracker", font=("new roman", 30, "italic
bold"), bg="#05897A", width=33
                    , fg="black", bd=5)
mainlabel.place (x=0, y=0)

label1 = Label (coro, text="Country Name", font=("arial", 20, "italic bold"),
bg="#046173")
label1.place (x=15, y=100)

label2 = Label (coro, text="Download File in ", font=("arial", 20, "italic bold"),
bg="#046173")
label2.place (x=15, y=200)

cntdata = StringVar ()
entry1 = Entry (coro, textvariable=cntdata, font=("arial", 20, "italic bold"),
relief=RIDGE, bd=2, width=32)
entry1.place (x=280, y=100)

Inhtml = Button (coro, text="Html", bg="#2DAE9A", font=("arial", 15, "italic bold"),
relief=RIDGE,
                 activebackground="#05945B",
                 activeforeground="white", bd=5, width=5, command=inhtmldownload)
Inhtml.place (x=300, y=200)

Injson = Button (coro, text="json", bg="#2DAE9A", font=("arial", 15, "italic bold"),
relief=RIDGE,
                 activebackground="#05945B",
                 activeforeground="white", bd=5, width=5, command=injsondownload)
Injson.place (x=300, y=260)

Inexcel = Button (coro, text="Excel", bg="#2DAE9A", font=("arial", 15, "italic bold"),
relief=RIDGE,
                  activebackground="#05945B",
                  activeforeground="white", bd=5, width=5, command=inexceldownload)
Inexcel.place (x=300, y=320)

Submit = Button (coro, text="Submit", bg="#CB054A", font=("arial", 15, "italic bold"),
relief=RIDGE,
                 activebackground="#7B0519",
                 activeforeground="white", bd=5, width=25, command=downloaddata)
Submit.place (x=450, y=260)

coro.mainloop ()
```

12. References

- https://pypi.org/project/tk/ (importing tk)
- https://pypi.org/project/beautifulsoup4/ (Importing beautifulsoap)
- https://www.worldometers.info/coronavirus/ (main reference)
- https://pypi.org/project/pandas/ (Importing pandas)
- https://www.geeksforgeeks.org/python-gui-tkinter/ (brief about tk)

Thank you ☺