A

Minor Project Report

On

Web Scraping (Covid-19 Tracker)

In partial fulfillment of the requirement for the award of the degree of

SUBMITTEDBY

Rahul Shakya (0901IT203D03)

Trapti Kaushal (0901IT203D06)

SUBMITTED TO

Prof. Vikas Sejwar

Dept. Of CSE & IT

Department of Information Technology

Madhav Institute of Technology and Science, Gwalior

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

Session: 2020-21

Madhav Institute of Technology and Science, Gwalior (M.P.)

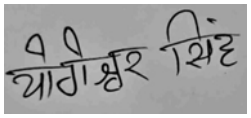
(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

CERTIFICATE

This is to certify that Rahul Shakya (0901IT203D03) and Trapti Kaushal (0901IT203D06) minor project, "Web Scraping (Covid-19 Tracker)" is a genuine record of a project completed under our supervision and guidance in partial fulfilment of the requirements for the award of a Bachelor of Technology in Information Technology in the Department of Information Technology, Madhav Institute of Technology and Science, Gwalior.

(Prof.Vikas Sejwar)

Mentor



(Prof. YOGESHWAR SINGH)

Mentor

CANDIDATE'S DECLARATION

I hereby declare that the Project entitled "**Web Scraping (Covid-19 Tracker)"** which is being submitted in the partial fulfilment of the requirement for the award of **Bachelor of Technology** in **Information Technology**.

All information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I have fully cited and referenced all material and results that are original to this work.

To the best of my knowledge the material presented in this Project has not been submitted elsewhere for the award of any other degree/diploma.

Date: 03/12/2021

Place: Gwalior **Rahul Shakya (0901IT203D03)**

Trapti Kaushal (0901IT203D06)

Content

1. Introduction

a. Introduction:-

Corona virus pandemic has been recognized as a global threat across the world and many methods were adopted for the prevention of this disease. This pandemic has caused global and economic disruption which resulted in numerous Covid-19 cases across the world. To know the number of cases and keep a track of this pandemic situation we need to collect the live data sets from the worldwide corona virus records.

This can be achieved by the technique of Web Scraping which enables the extraction of live data sets from a specific platform. It facilitates the user to access the World Wide Web wherein specific data is gathered, copied from the web and then it is stored in a central local database then provides ways to retrieve and analyse the data. This estimate is to design a platform where you can obtain the live data sets and have a compact knowledge about the present scenario.

This is an elementary approach to scrap the live data sets through a user interface from the Worldometer Covid-19 data set with the Tkinter. To implement this scheme, we use the Python programming language. To effectuate this task, we acquire the process of making GUI calls to the Worldometer Covid-19 website and simultaneously we will make use of the regular expressions to extract the data from the web page. However, this action includes a series of tactics that has to be recognized analysed and accomplished sequentially. Initially the input is given by the user in the form of speech. Then the required contents are searched and matched with the user's input. If the contents match then with the help of a Google voice assistant result is obtained which is the output in turn.

b. Problem Statement:-

- Bot access
- Web page structures
- IP blocking
- Captcha
- Honeypot traps
- Slow/unstable load speed
- Dynamic content
- Login requirement
- Real-time data scraping

c. Overview:-

Coronavirus, one of the biggest pandemic has brought all the world to Danger. Along with this, it is one of the trending News, everyone has this day. In this Project, we will be scraping data and showing Covid-19 statistics in human-readable form. The data will be scraped from "Worldometer" website.

2. System Requirement

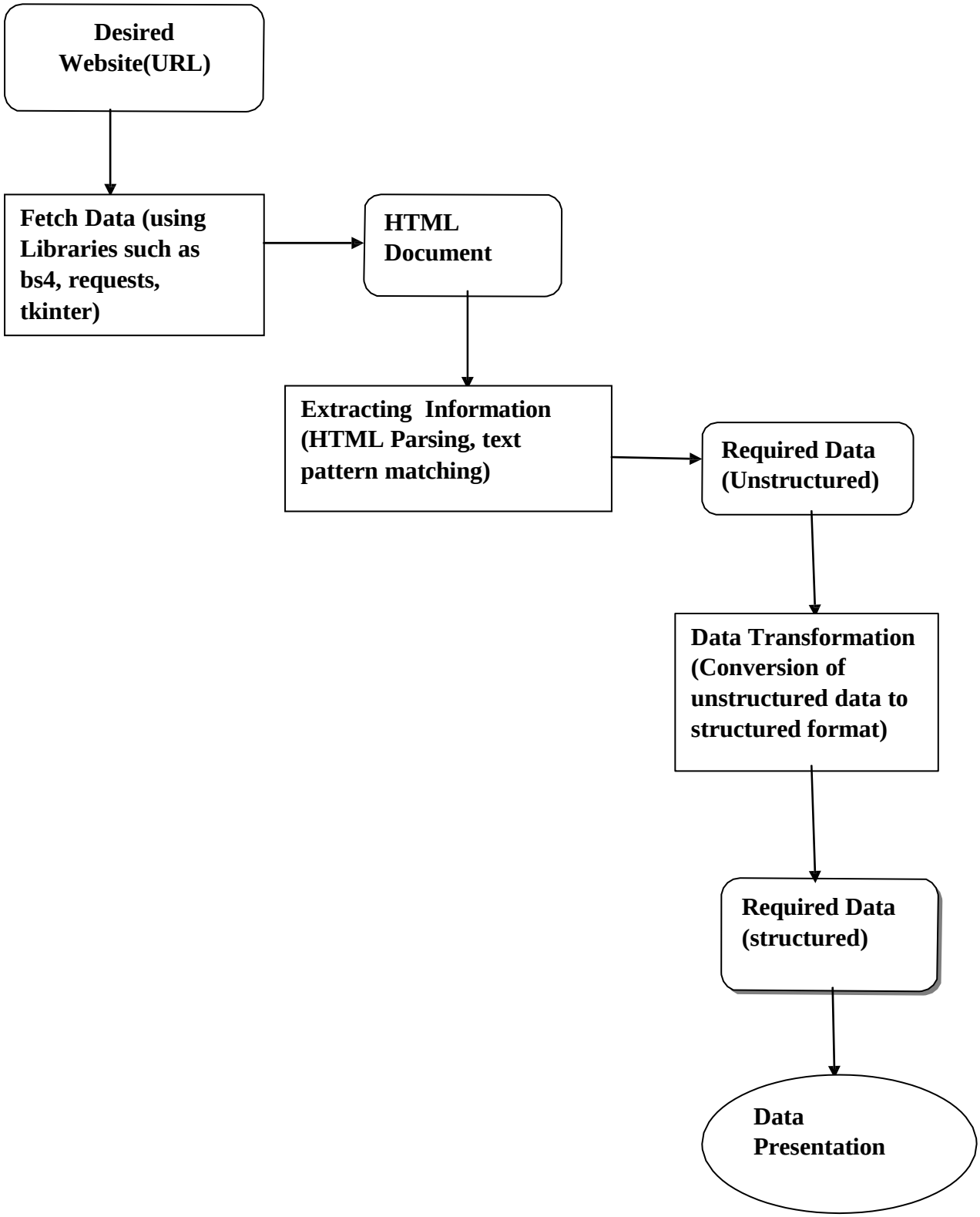
Hardware Requirement:

- System : Multimedia PC
- Processor : Pentium 4 or above
- Memory : 512MB RAM
- Hard Disk : 80GB or above
- Keyboard : 104 standards
- Monitor : SVGA
- Internet

Software Requirement:

- Python IDLE
- Chrome Browser

3. System Architecture



4. Implementation

Web scraping is a term used to describe the use of a program or algorithm to extract and process large amounts of data from the web. Whether you are a data scientist, engineer, or anybody who analyzes large amounts of datasets, the ability to scrape data from the web is a useful skill to have.

Required Modules:

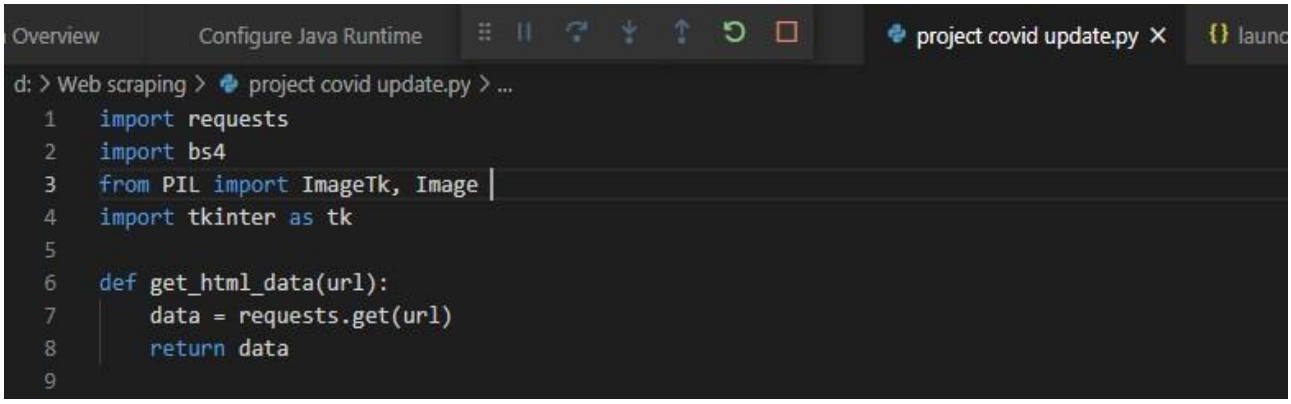
I. **Requests:-**The requests module allows you to send HTTP requests using Python. The HTTP request returns a Response Object with all the response data (content, encoding, status, etc).

II. **BeautifulSoup4:-** Beautiful Soup is a Python library that is used for web scraping purposes to pull the data out of HTML and XML files. It creates a parse tree from page source code that can be used to extract data in a hierarchical and more readable manner.

III. **Tkinter** :- Tkinter is a Python Package for creating GUI applications. Python has a lot of GUI frameworks, but Tkinter is the only framework that's built into the Python standard library.

IV. **Pillow:-** Python Imaging Library (expansion of PIL) is the de facto image processing package for Python language. It incorporates lightweight image processing tools that aids in editing, creating and saving images.

Code Design:



```
10
11  def get_covid_data():
12      url="https://www.worldometers.info/coronavirus/"
13      html_data = get_html_data(url)
14      bs = bs4.BeautifulSoup(html_data.text, 'html.parser')
15      info_div = bs.find("div", class_ = "col-md-8").findAll("div", id = "maincounter-wrap")
16      all_data=""
17
18      for block in info_div:
19          text = block.find("h1", class_=None).get_text()
20
21          count = block.find("span", class_=None).get_text()
22
23          all_data = all_data + text + count + "\n"
24      return all_data
25
26
27  def get_country_data():
28      name = textfield.get()
29      url = "https://www.worldometers.info/coronavirus/country/"+name
30      html_data = get_html_data(url)
31      bs = bs4.BeautifulSoup(html_data.text, 'html.parser')
32      info_div = bs.find("div", class_ = "col-md-8").findAll("div", id = "maincounter-wrap")
33      all_data=""
34
35      for block in info_div:
36
37          try:
38              text = block.find("h1", class_=None).get_text()
39              count = block.find("span", class_=None).get_text()
40              all_data = all_data + text + count + "\n"
41          except AttributeError:
42              continue
43      mainlabel['text']=all_data
44
45
46  def reload():
47      new_data = get_covid_data()
48      mainlabel['text']=new_data
49
50
51  get_covid_data()
52
53
54  root = tk.Tk()
55  root.geometry("600x500")
56  root.title("Covid Tracker")
57  f = ("poppins", 15, "bold",)
```

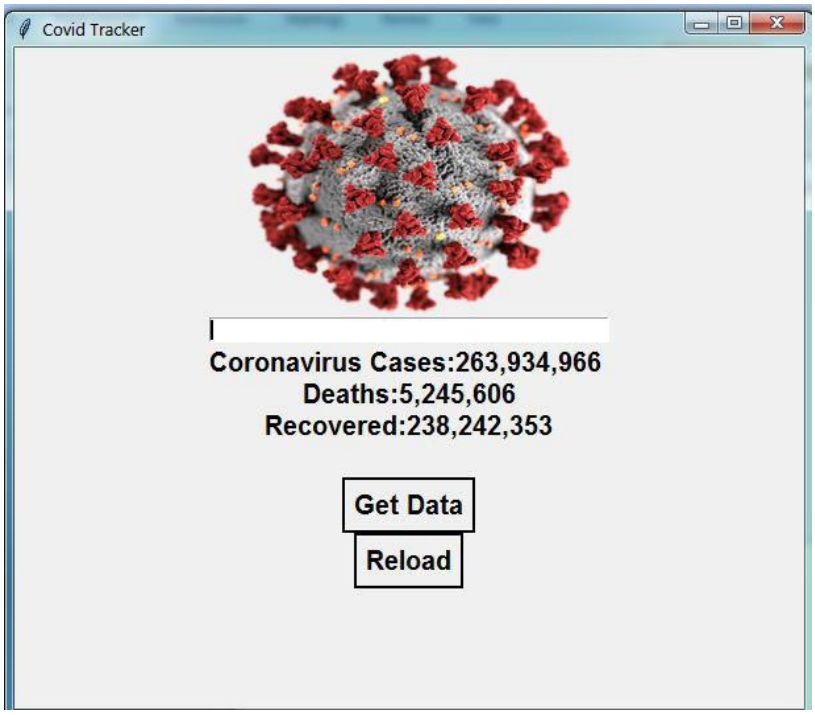
nch Chrome against localhost (hello sum) Ln 3, Col 32 Spaces: 4

```
58
59
60  img = Image.open('D:\Web scraping\logo.png')
61
62  resize_img = img.resize((250, 200), Image.ANTIALIAS)
63  new_img = ImageTk.PhotoImage(resize_img)
64  #banner = tk.PhotoImage(pic)
65  bannerlabel = tk.Label(root, image=new_img)
66  bannerlabel.pack()
67
68
69
70  textfield = tk.Entry(root, width = 50)
71  textfield.pack()
72
73  mainlabel = tk.Label(root, text=get_covid_data(),font=f)
74  mainlabel.pack()
75
76  gbtn = tk.Button(root, text="Get Data", font=f, relief='solid', command=get_country_data)
77  gbtn.pack()
78
79  rbtn = tk.Button(root, text="Reload", font=f, relief='solid', command=reload)
80  rbtn.pack()
81
82  root.mainloop()
```

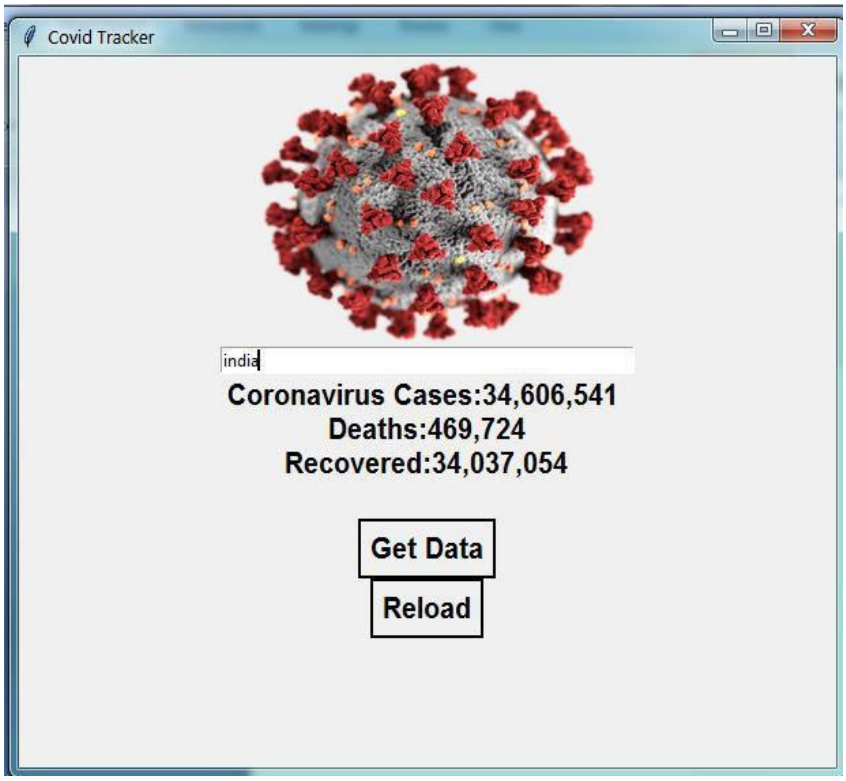
nch Chrome against localhost (hello sum) Ln 3, Col 32 Spaces:

Results:

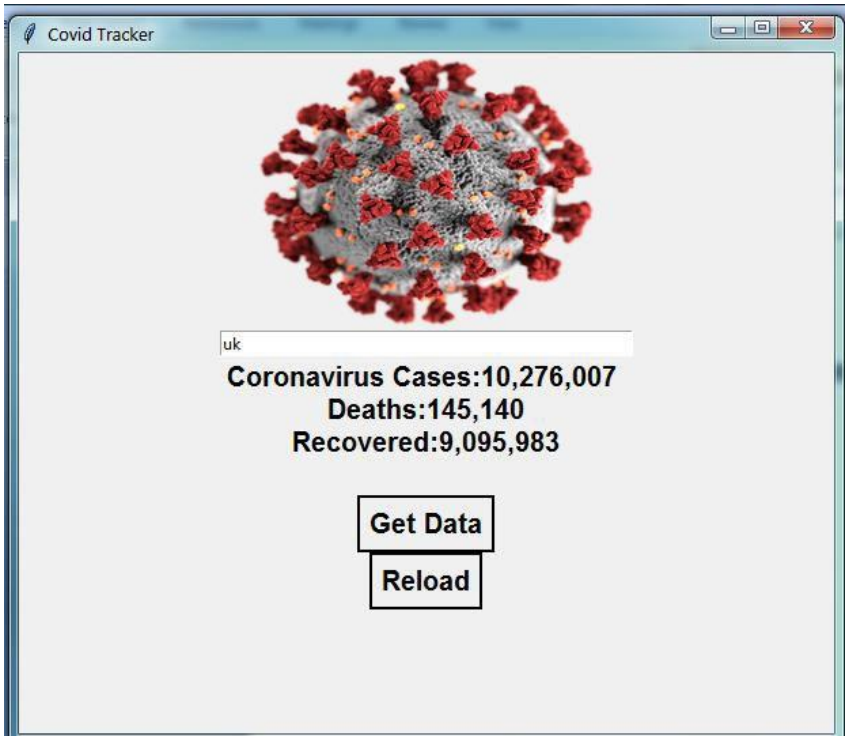
Example 1:



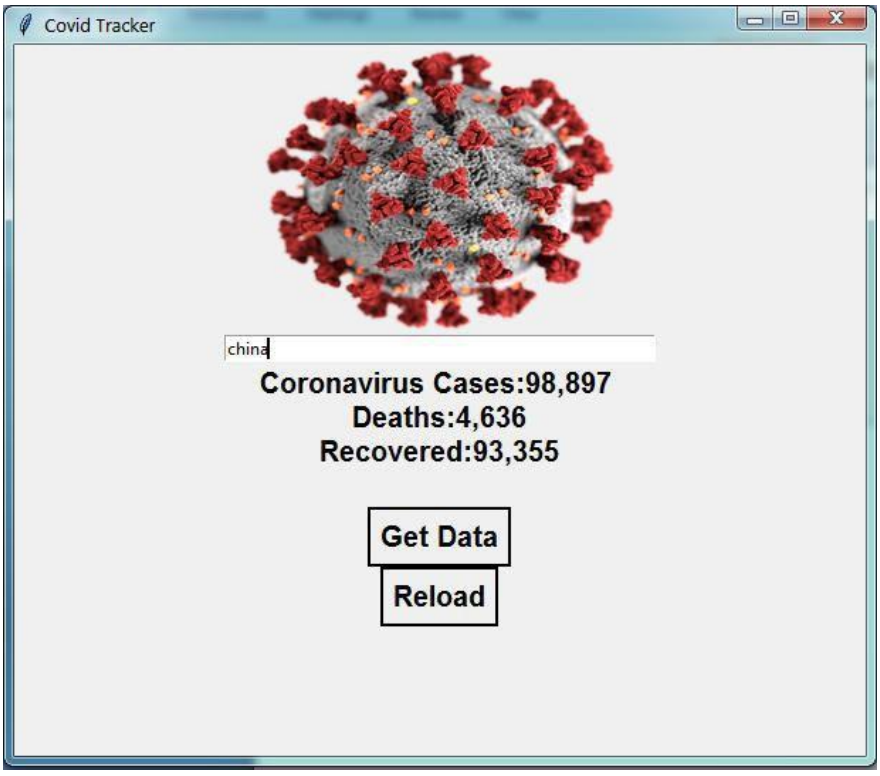
Example 2:



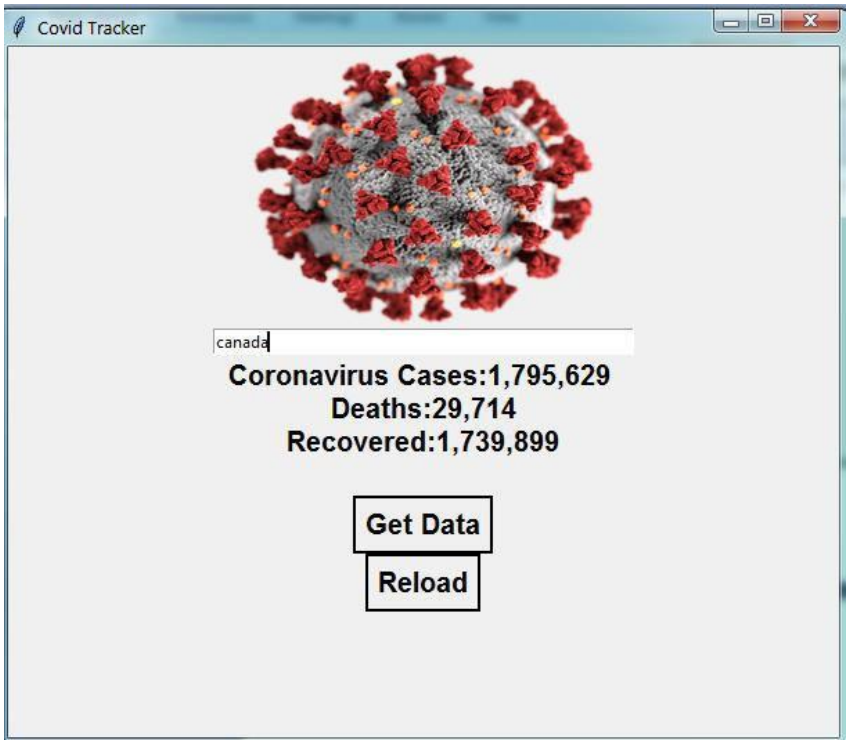
Example 3:



Example 4:



Example 5:


Covid Tracker
canada
Coronavirus Cases:1,795,629
Deaths:29,714
Recovered:1,739,899
Get Data
Reload

5. Conclusion and Future Scope

There are many methods or ways in which we can extract information from live hosted websites. Majorly when you do not have access or facility of ready web API available to get the logs or data. You can use web scraping to download the web data on to your local machine using python.If you want to dive deep into the concepts and applications of Python in various domains, upskill with Great Learning's programs in Machine Learning and Data Science.

here is a requirement for large datasets everywhere. Based on the features to be extracted, either new data must be collected, or existing data must be upgraded regularlyIn this Project we have presented a unique methodology to scrape data from popular web site like worldometer, aiming for its future application in sentiment and emotion analysis. There are many challenges that are to be faced in the world of websites during the process of web scraping. This project also gives a glance about them and includes generalized solutions that can be embedded into the main method itself. We hope that our methodology and information outlined in this project helps many users in creating new datasets.

\

This project provides an efficient way of showing the identified Covid-19 patient to the users in a Tkinter Dashboard. This developed project can be employed as a tool for creating further social awareness among the people. This application further tracks. checks whether it is present in the list of identified containment zones. It sends separate notification alerts to the user on entering. The developed android application further extracts the IMEI Number of the trespasser in the containment zones which can be useful to the local police to track and identify people who are frequently trespassing the containment zones. Thereby this application identifies the containment zones and highlights the need for taking further precautionary measures for combating Covid-19. The application has been tested in various locations and has been found to yield accurate results.

The future of web scraping is actually linked with the future of web! It is anybody's guess where web is headed. With 90% of the data piled up in the last two years, web is a giant mechanism absorbing data.

6. References

1. https://www.worldometers.info/coronavirus/
2. https://www.geeksforgeeks.org/implementing-web-scraping-python-beautiful-soup/
3. https://www.tutorialspoint.com/python_web_scraping/index.htm
4. https://www.youtube.com/watch?v=uufDGjTuq34&t=2357s