**PYTHON ASSIGNMENT -02**

**Name: BAVANI SRI M**

**Register No: 192311405**

**Date of**  **Submission: 26/08/2024**

**Title:**: **Real-Time COVID-19 Statistics Tracker**

**Scenario:**

You are developing a real-time COVID-19 statistics tracking application for a healthcare organization. The application should provide up-to-date information on COVID-19 cases, recoveries, and deaths for a specified region.

**Tasks:**

**1.** Model the data flow for fetching COVID-19 statistics from an external API and displaying it to the user.

**2**. Implement a Python application that integrates with a COVID-19 statistics API (e.g., disease.sh) to fetch real-time data.

**3.** Display the current number of cases, recoveries, and deaths for a specified region.

**4.** Allow users to input a region (country, state, or city) and display the corresponding COVID-19 statistics.

**Deliverables:**

**•** Data flow diagram illustrating the interaction between the application and the API.

• Pseudo code and implementation of the COVID-19 statistics tracking application.

• Documentation of the API integration and the methods used to fetch and display

• Explanation of any assumptions made and potential improvement.

**Solution:**

**Real-Time COVID-19 Statistics Tracker**

* **Data flow diagram:**

**Implementation:**

import requests

BASE\_URL = 'https://disease.sh/v3/covid-19'

def get\_global\_stats():

url = f'{BASE\_URL}/all'

try:

response = requests.get(url)

response.raise\_for\_status()

data = response.json()

print('Global COVID-19 Statistics:')

print('Total Cases: ' + str(data['cases']))

print('Total Deaths: ' + str(data['deaths']))

print('Total Recovered: ' + str(data['recovered']))

except requests.RequestException as e:

print('Failed to get global statistics: ' + str(e))

def get\_country\_stats(country):

url = f'{BASE\_URL}/countries/{country}'

try:

response = requests.get(url)

response.raise\_for\_status()

data = response.json()

print('\nCOVID-19 Statistics for ' + country.capitalize() + ':')

print('Date: ' + str(data['updated']))

print('Cases: ' + str(data['cases']))

print('Deaths: ' + str(data['deaths']))

print('Recovered: ' + str(data['recovered']))

except requests.RequestException as e:

print('Failed to get statistics for ' + country + ': ' + str(e))

if \_\_name\_\_ == '\_\_main\_\_':

get\_global\_stats()

country = 'us'

get\_country\_stats(country)

* **Display the Covid -19 Statistics:**

Global COVID-19 Statistics:

Total Cases: 704753890

Total Deaths: 7010681

Total Recovered: 675619811

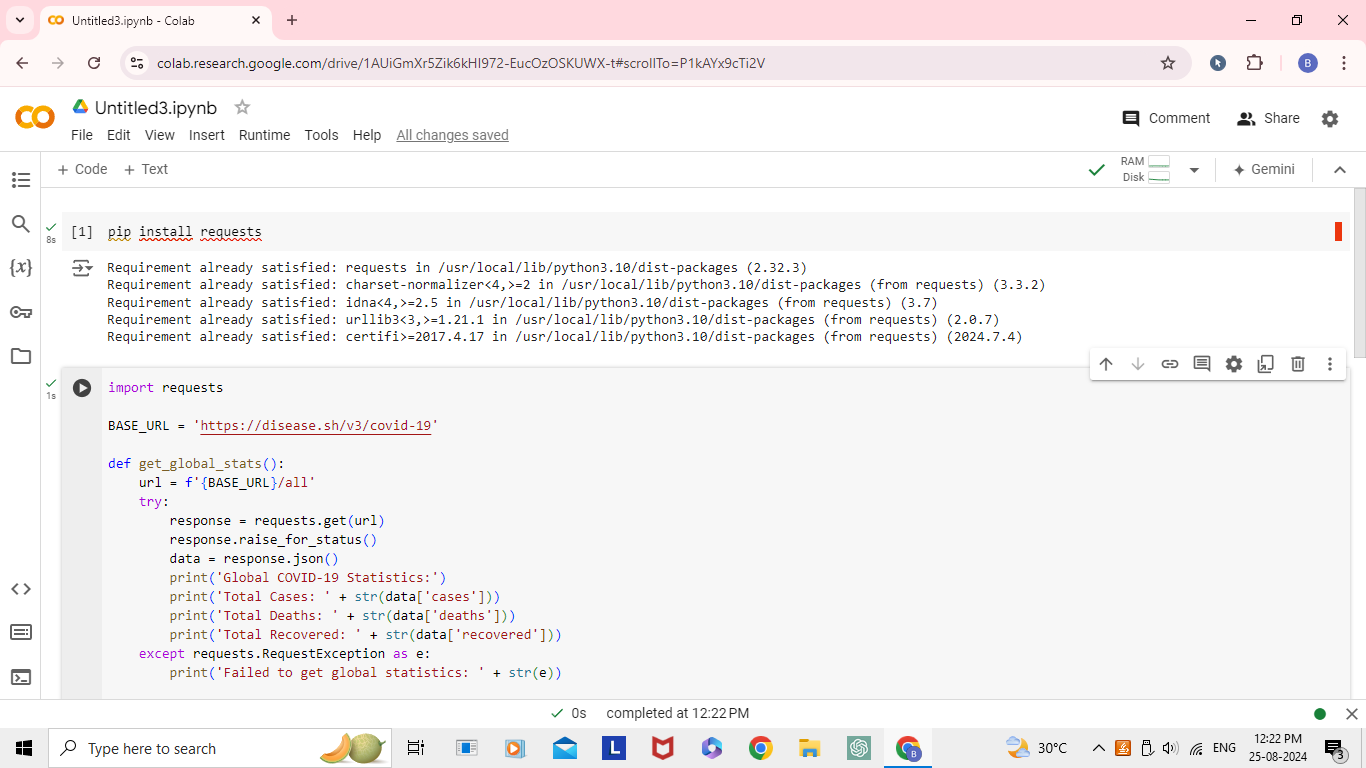
COVID-19 Statistics for Us:

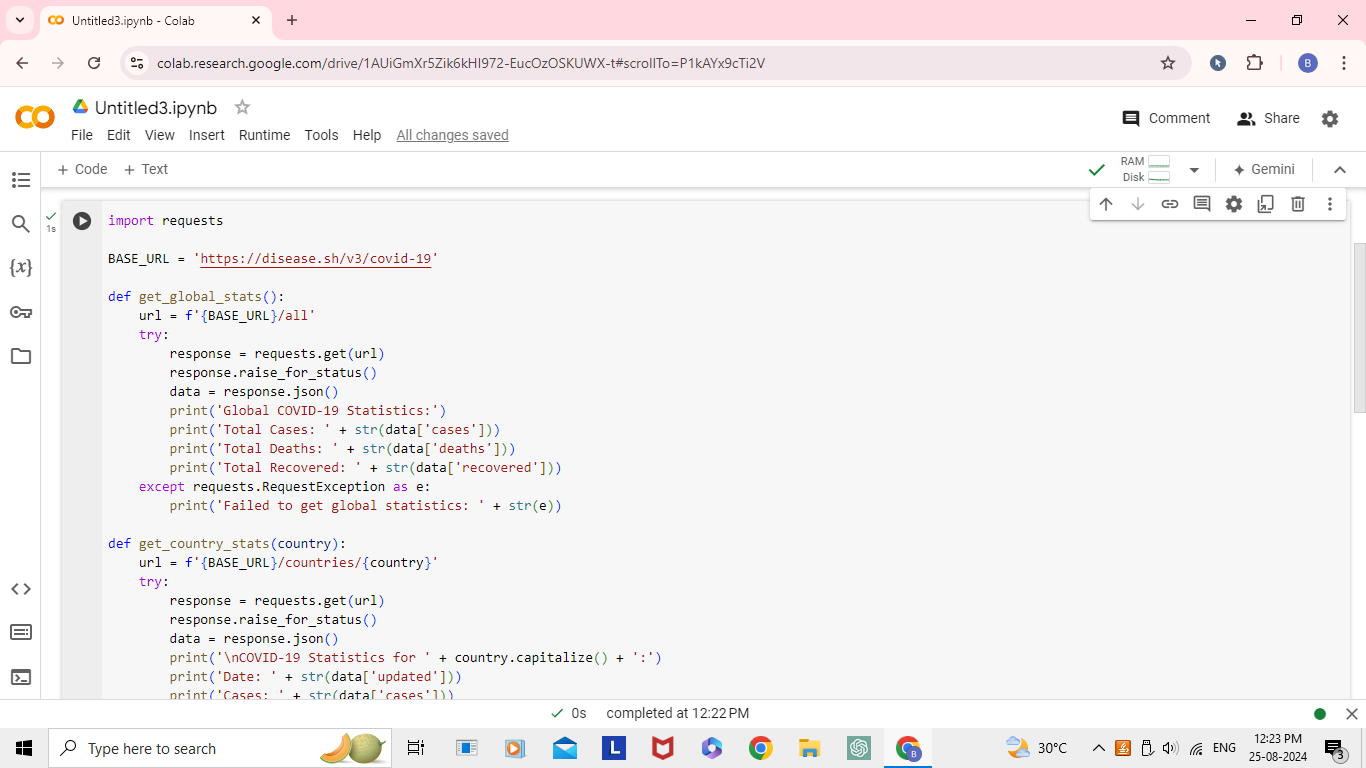
Date: 1724566330812

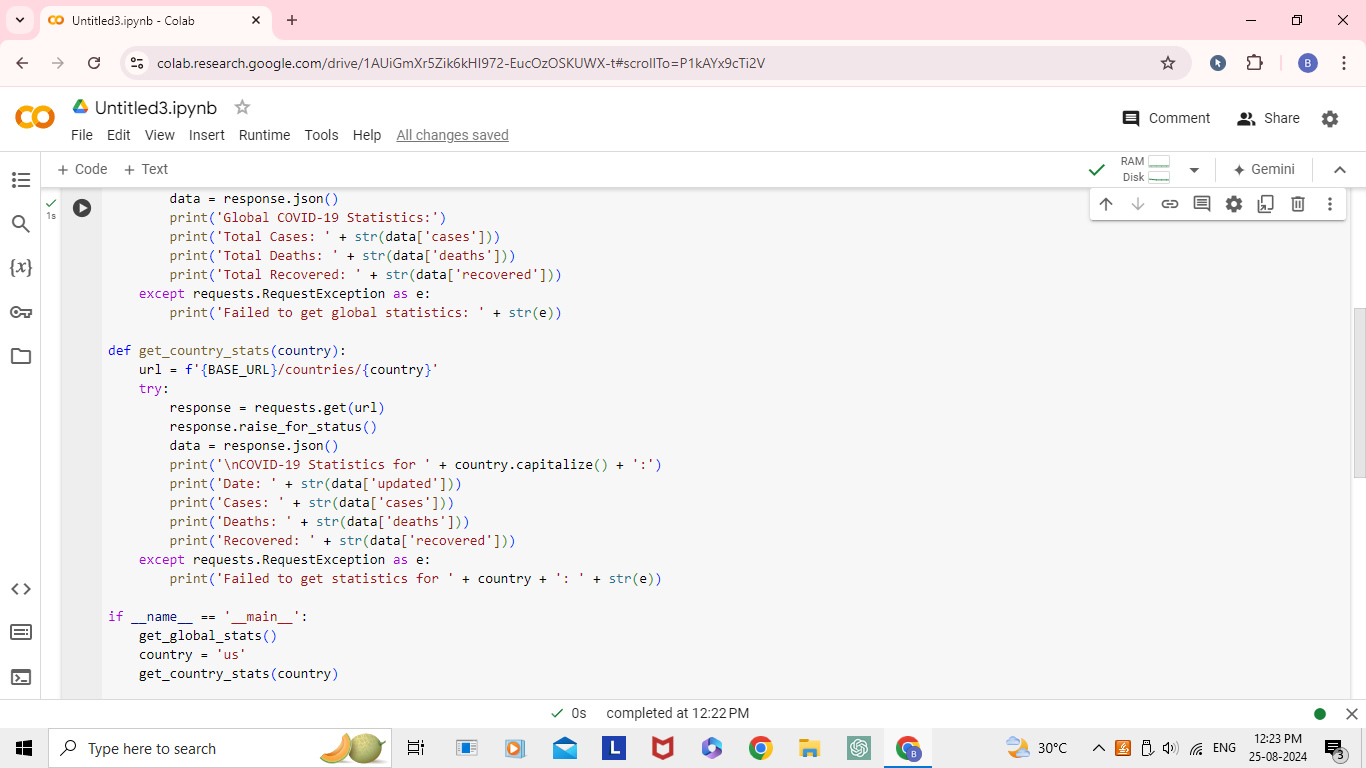
Cases: 111820082

Deaths: 1219487

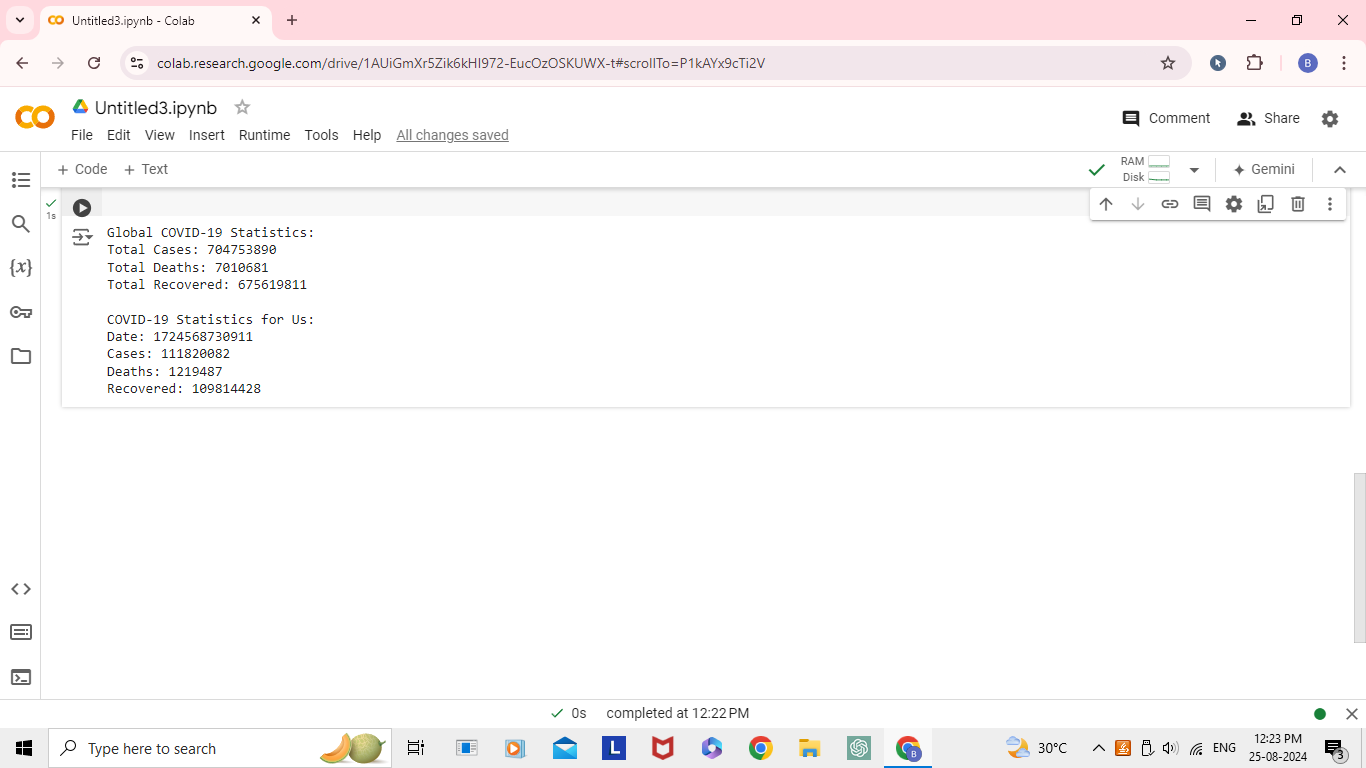
Recovered: 109814428

*  **User Input:**

****

****

* **Output:**

****

**Documentation:**

**Detailed explanation of the code:**

● The program sends HTTP requests to the disease.sh- provided COVID-19 API via the requests library. The get\_covid\_tates function delivers the current number of cases, recoveries, and deaths for a region (country, state, or city) that it receives as input.

● The user-friendly formatting and printing of the COVID-19 statistics is the responsibility of the display\_covid\_ stats function. It presents the data on cases, recoveries, and deaths with the proper formatting applied (e.g., adding commas to high numbers).

● The display\_covid\_ stats function is in charge of formatting and printing the COVID-19 statistics in an understandable manner. It displays the case, recovery, and death data with the appropriate formatting (commas added to large numbers, for example).

* **Assumptions:**

1. The program is predicated on the diseases API being operational and delivering precise real-time COVID-19 data.
2. The program anticipates that the user will enter a legitimate region—a nation, state, or city—that the API is able to identify.
3. Possible Enhancements:
4. Incorporate error management into the program to adapt to API malfunctions or erroneous input from the user.

5) Offer extra capabilities, like the capacity to show trending, historical COVID-19 data, or infographics.

6)To enhance the user experience, integrate the program with a user interface (such as a web application or a mobile app).   
7) Permit users to compare the COVID-19 statistics side by side by selecting several regions.

8) Offer the option to configure warnings or alerts for noteworthy alterations in COVID-19 data.

**Limitations:**

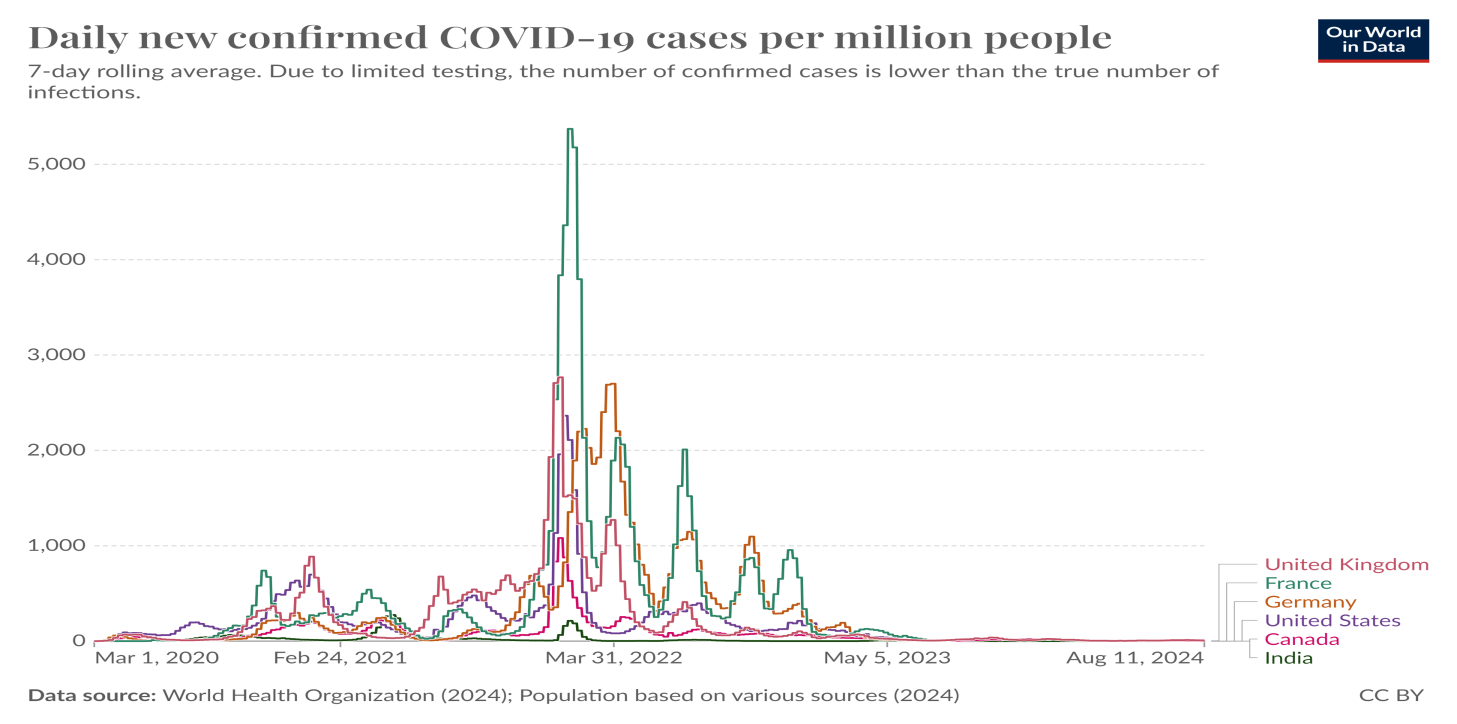
1. Because of reporting delays, the data might not always be current.
2. At the moment, the application only supports nations; inquiries at the state and city levels might need further endpoints.

1. Lack of Configuration for Base URL

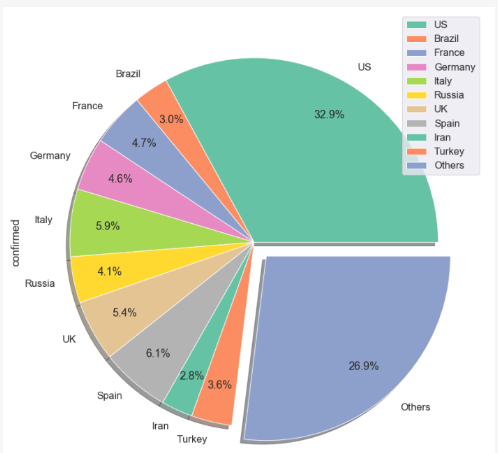
 **Issue**: The base URL for the API is hardcoded. If the API endpoint changes or if a different API is used, the script will need modification.

* **Solution**: Consider storing the base URL in a configuration file or environment variable for easier updates.
* **Graph:**

[**Daily new confirmed COVID-19 cases per million people**](https://ourworldindata.org/explorers/covid?zoomToSelection=true&time=2020-03-01..latest&facet=none&pickerSort=desc&pickerMetric=new_cases_smoothed_per_million&Metric=Confirmed+cases&Interval=7-day+rolling+average&Relative+to+population=true&country=IND~USA~GBR~CAN~DEU~FRA)**:**

****

* Distribution of cumulative confirmed COVID-19 cases per country as of 4 May 2019

****