**CS5540 - PRINCIPLES OF BIG DATA MANAGEMENT**

**Twitter Data Analysis**

**Project Phase-2 Report**

Team Members:

|  |  |  |
| --- | --- | --- |
| **Name** | **ID** | **Email ID** |
| Shiva Kumar Murarishetti | 16296890 | [smpf4@mail.umkc.edu](mailto:smpf4@mail.umkc.edu) |
| Lavanya Gadde | 16298160 | [lavanya.gadde@mail.umkc.edu](mailto:lavanya.gadde@mail.umkc.edu) |
| Sravani Garikapati | 16296889 | [sg324@mail.umkc.edu](mailto:sg324@mail.umkc.edu) |

GitHub Link for Code: <https://github.com/Murarishetti-Shiva-Kumar/PBDM.git>

Tweets Data File: <https://drive.google.com/file/d/116XKRNEKs1aBrX-H4LJh3Tyg0QScETcP/view?usp=sharing>

**Prerequisites to be installed & configured**

1. Install java (<https://www.oracle.com/java/technologies/javase-jdk14-downloads.html>)
2. Set Java\_Home
3. Install Python
4. Install Spark(<https://spark.apache.org/downloads.html>)
5. If Hadoop is not installed in windows, download winutils.exe file
6. Set Hadoop\_Home & Spark\_Home in Environmental variables

**Design**

Analysing data using Apache Spark

Loading of JSON data into Apache Spark

Collection of Tweets Data

Processing data using Spark SQL-> Generating Visualizations

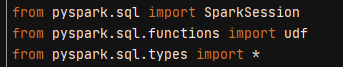
**Implementation Steps**

1. Spark can be triggered either with scala orelse with PySpark
2. We used PySpark as Python API for Spark
3. Flask is used to create web Framework
4. Spark Session is used as entry point to interact with spark, to use SparkSession import package spark

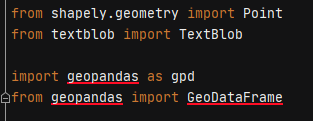




1. To Create RDD’s and Data Frames, we have imported packages



1. To generate Visualization graphs, we have used packages matplotlib, geopandas, textblob, shapely





1. To delete files generated, import package os



1. For date and time implementation used package datetime



1. To render graph onto UI page, we converted the generated image into BytesIO(internally generates 1, 0’s) and invoked them on UI. This is done using BytesIO package



1. We have used Flask Web Framework to create UI application, so imported packages related to flask



1. Creating Spark Session using SparkSession method to interact with Apache Spark



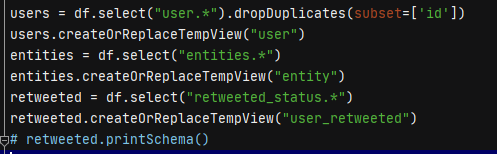
1. Loading the Tweet Data into a Data Frame



1. To Use the Data frame in the Spark SQL , lets create a view for it



1. As this has many child objects, to invoke them in easy way. We have created separate Views for Users, Entities and Retweets
2. To Clean the data to some extent we have removed duplicates by using dropDuplicates()

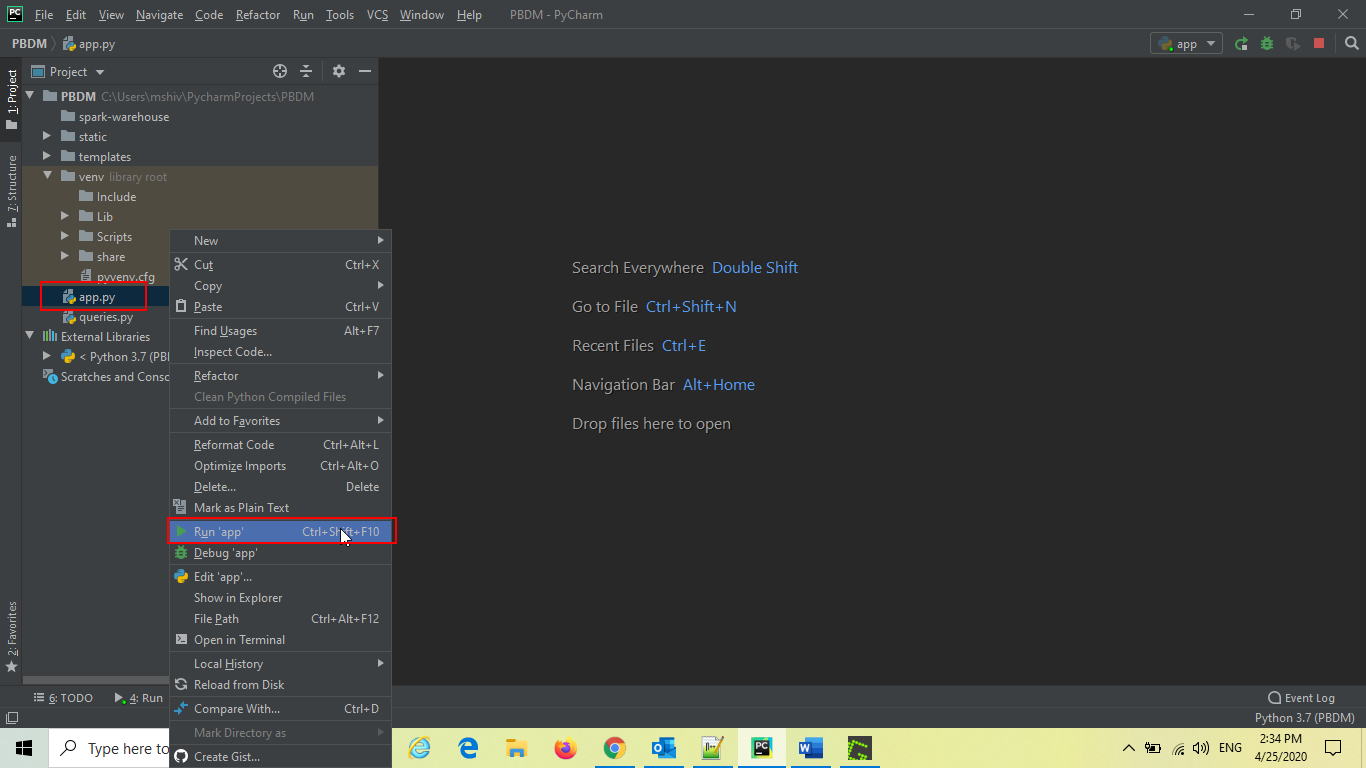


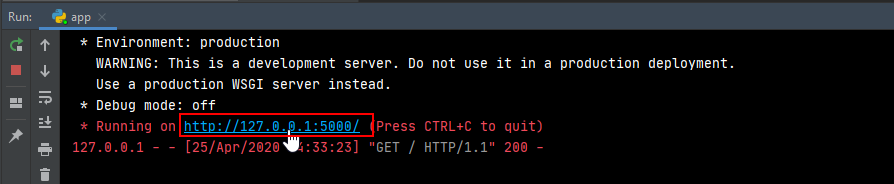
1. To check the schema structure of the view, use printschema()



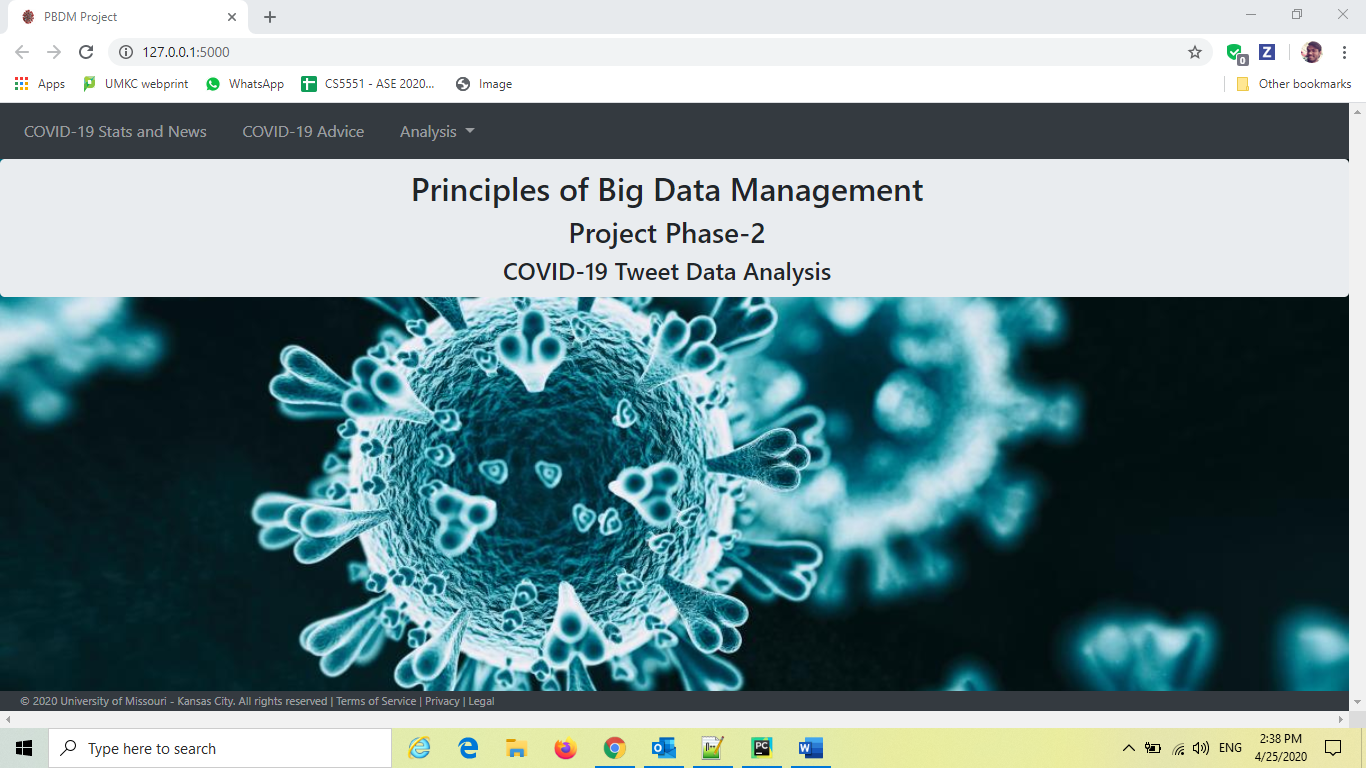
**Steps to start application**

1. Clone the code from the git
2. Open any of the Python IDE(PyCharm)/cmd terminal
3. Download the Twitter data from the drive link orelse run the importing\_tweets.py. It downloads tweetsdata.txt(Tweets Data File)
4. Set the Path of tweet data file in the queries.py program
5. Install all the dependencies mentioned in the import statements
6. Run app.py file

****

****

Once the application starts, it generates the URL. Just click on the URL. Homepage of the application gets opened.

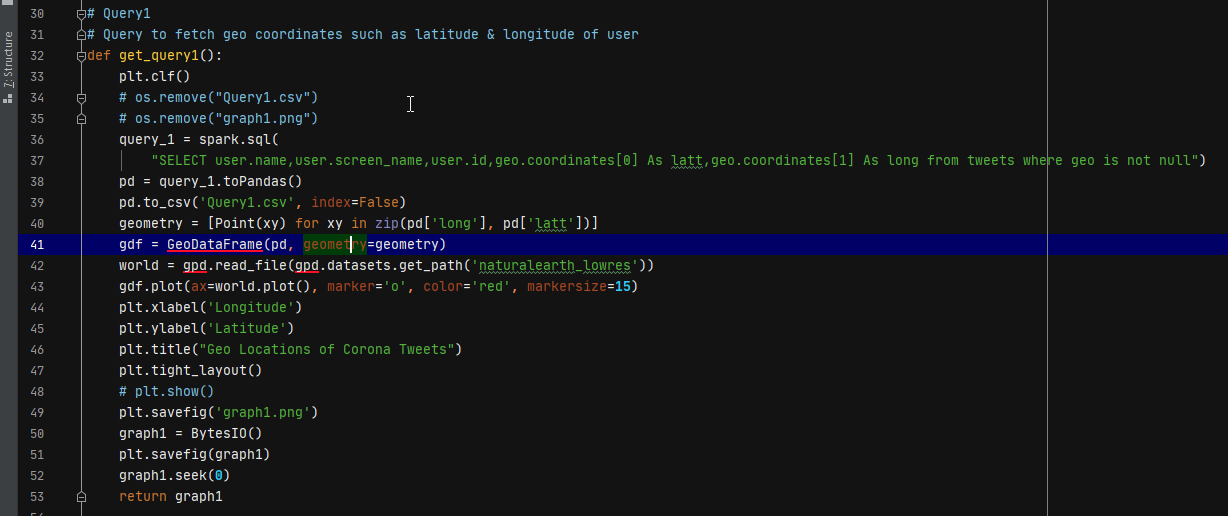




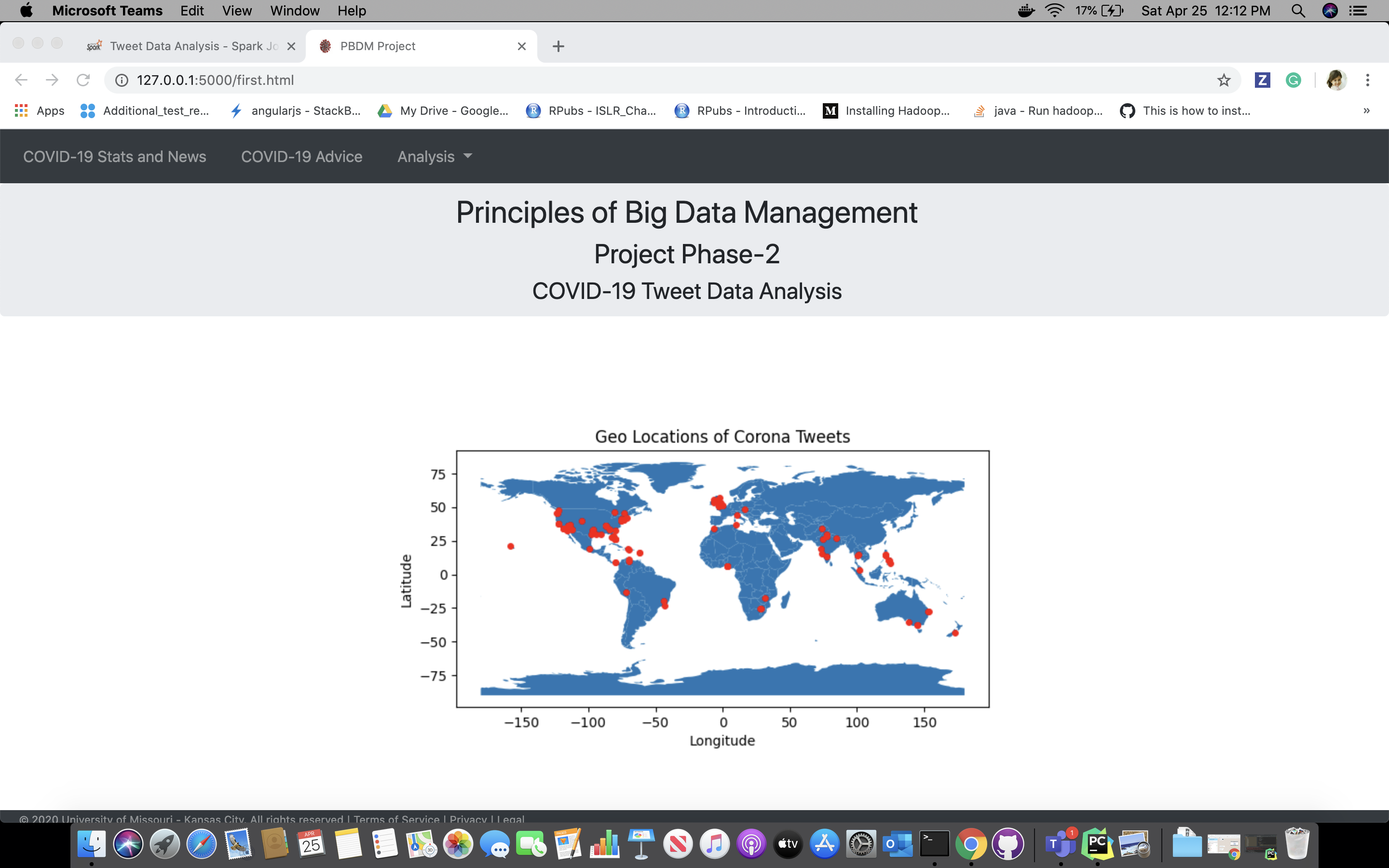
**# Query1**

**# Query to fetch geo coordinates such as latitude & longitude of user**

**Geo Locations of Corona Tweets**



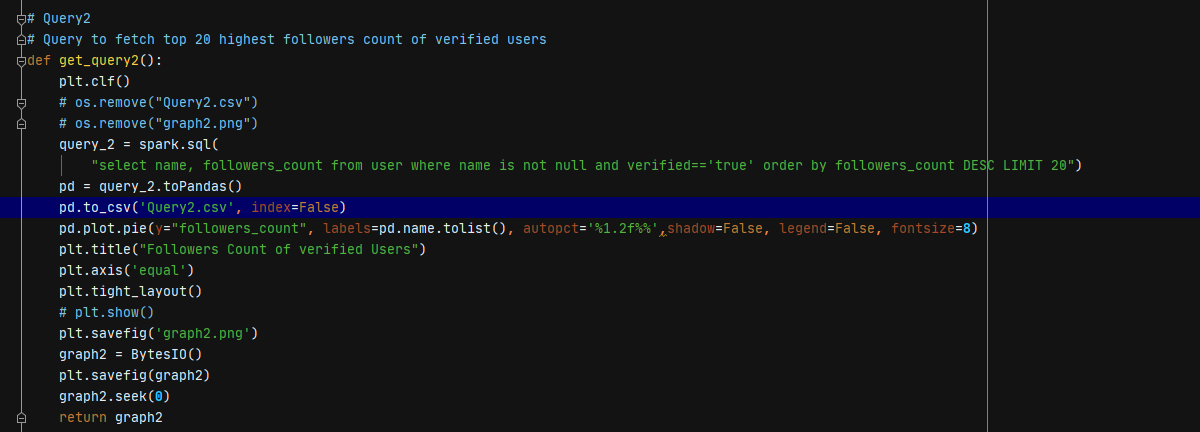
os.remove is used to delete the previously generated files. GeoDataFrame is used to plot the map with Latitude & Longitude



**# Query2**

**# Query to fetch top 20 highest followers count of verified users**

**Followers Count of verified Users**



Matplotlib is used to generate the pie graph, autopct is used to specify values on graph, fontsize is specified for the font size of the label

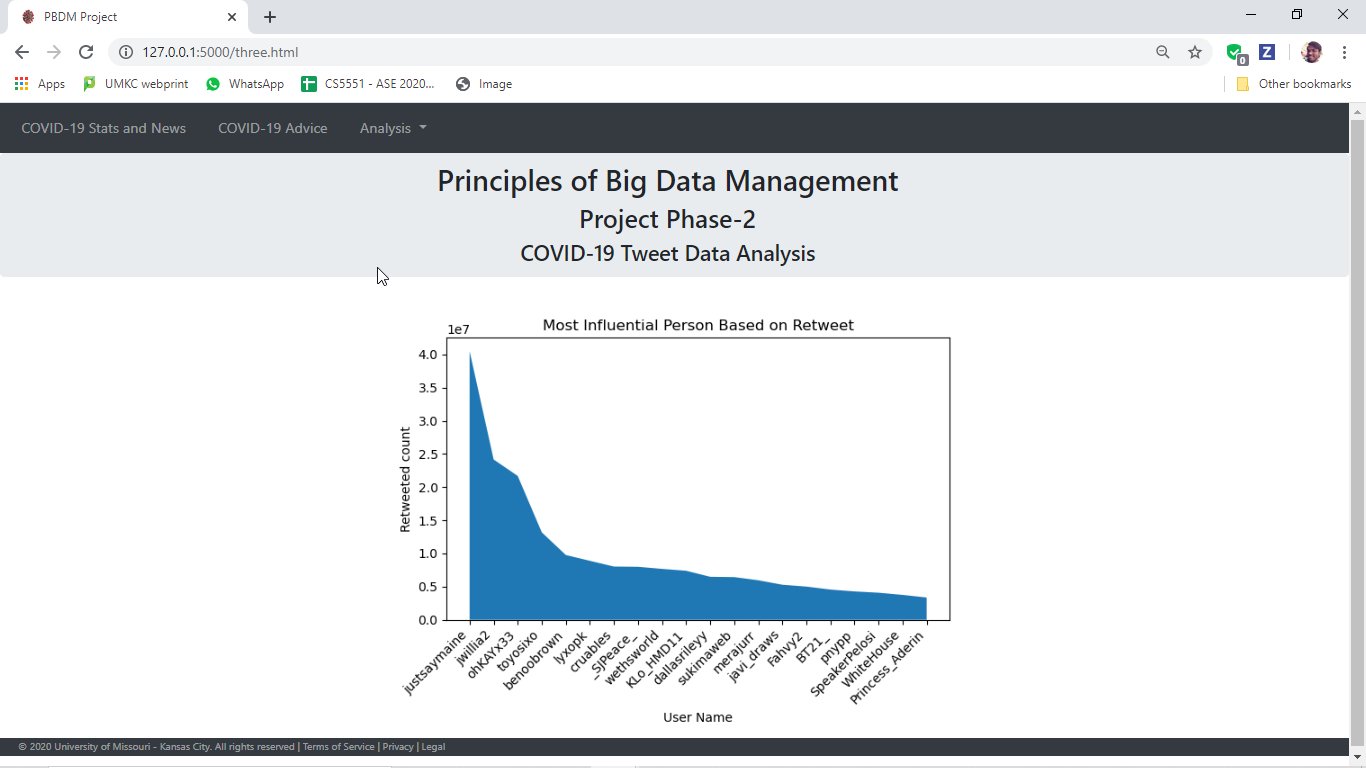


**# Query3**

**# Query to fetch 20 most influential person based on retweeted count**

**Most Influential Person Based on Retweet**

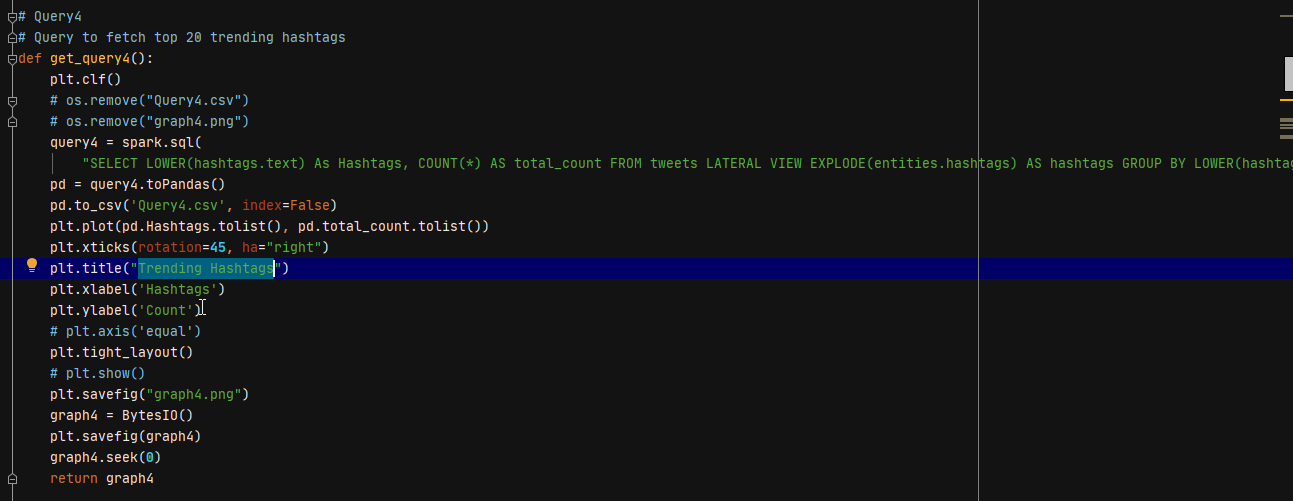


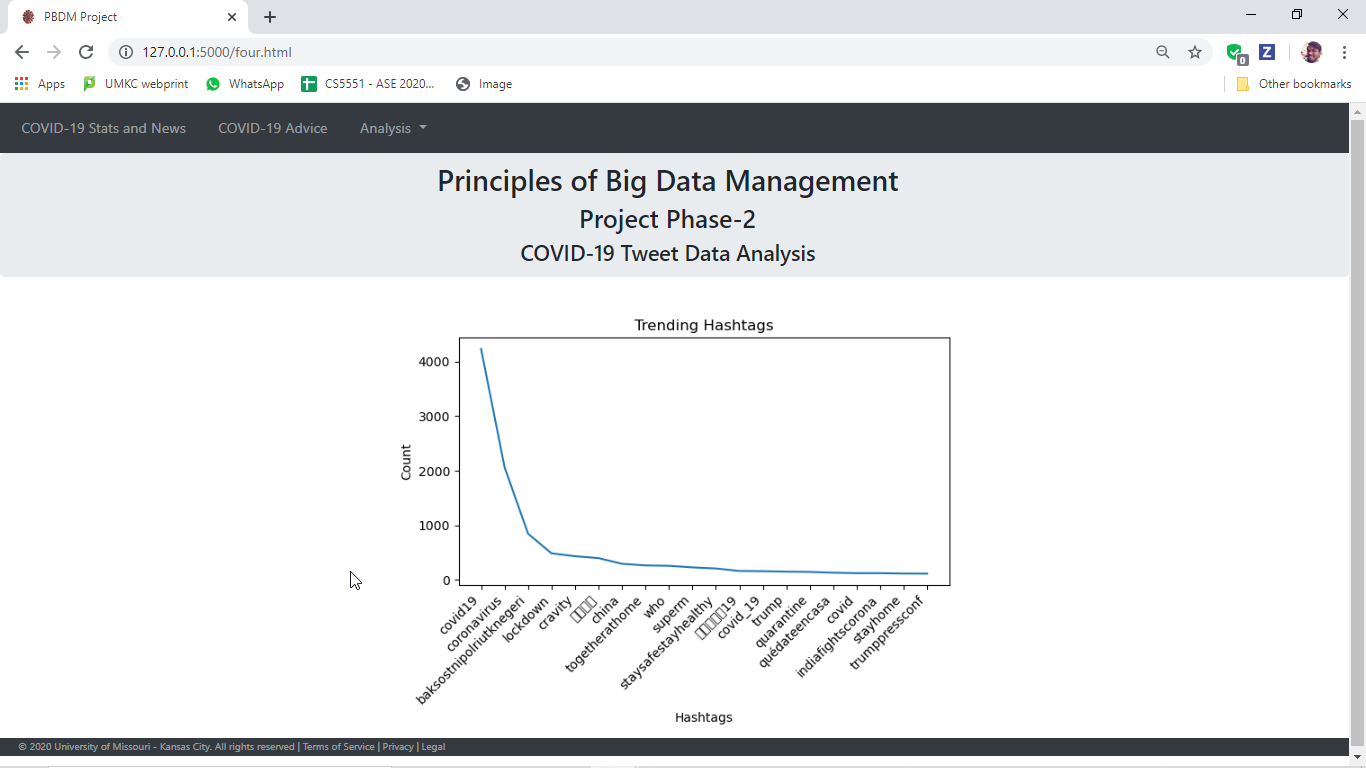


**# Query4**

**# Query to fetch top 20 trending hashtags**

**Trending Hashtags**

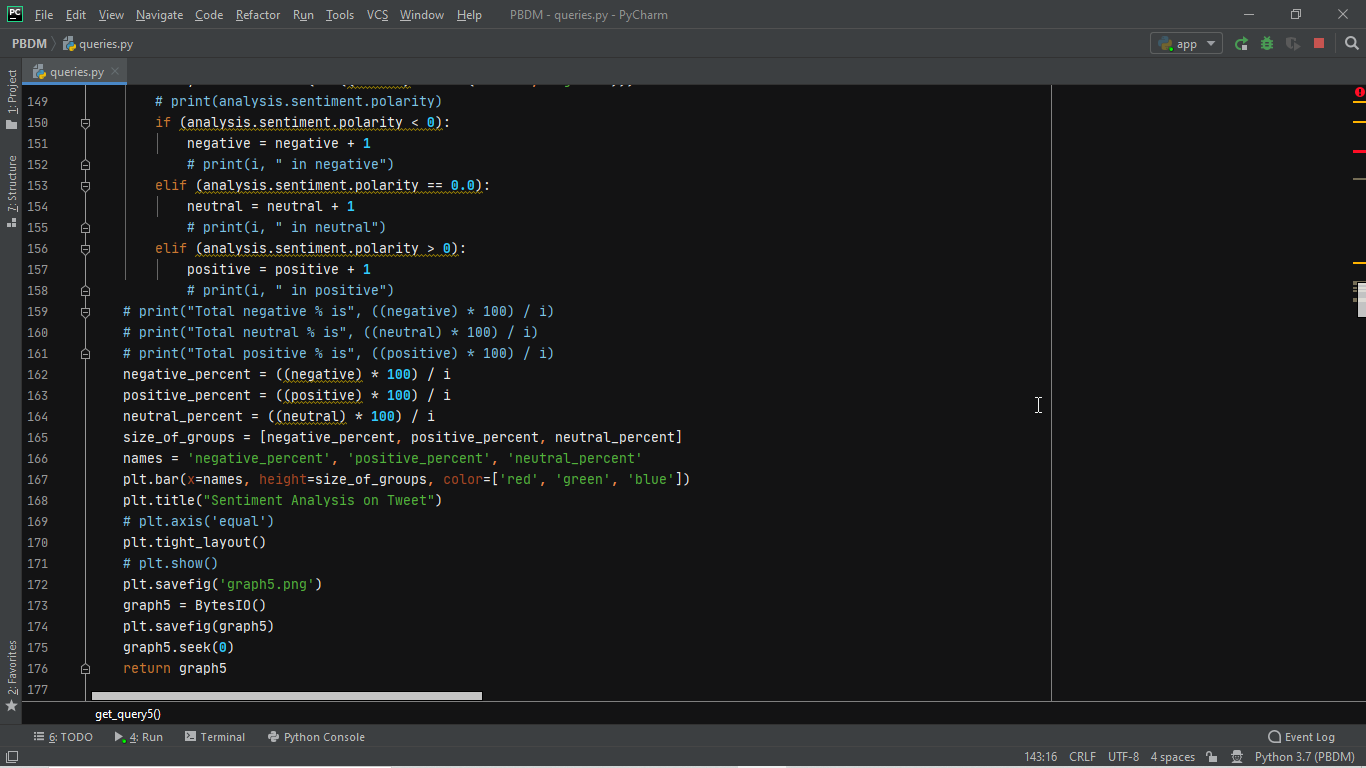
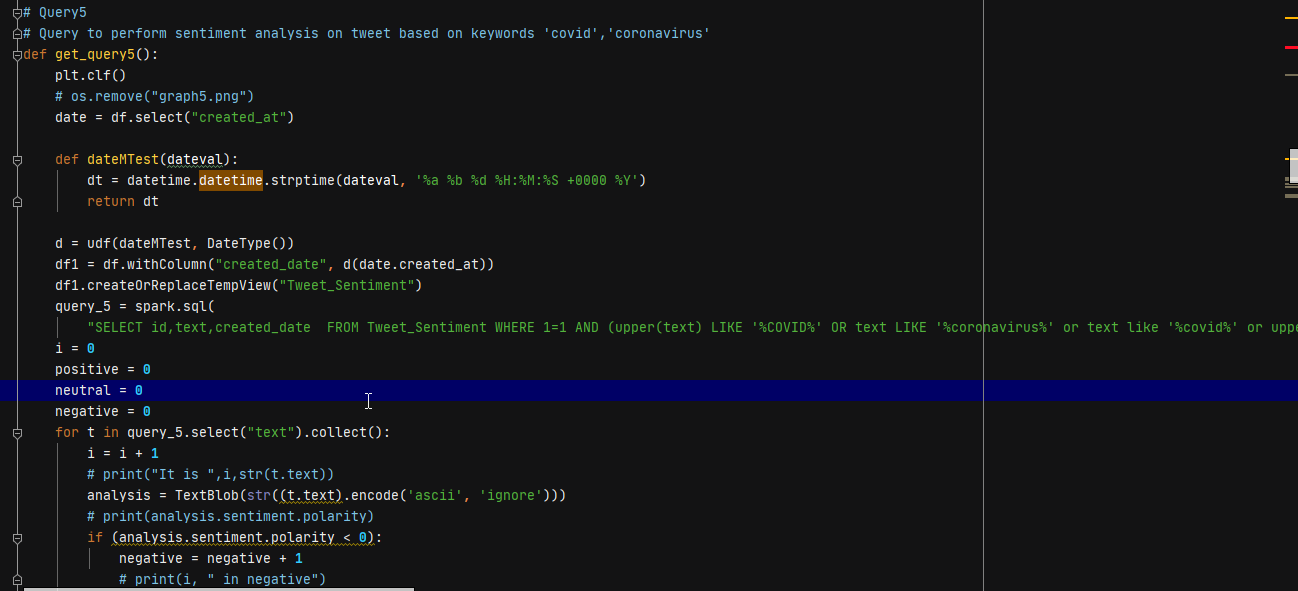




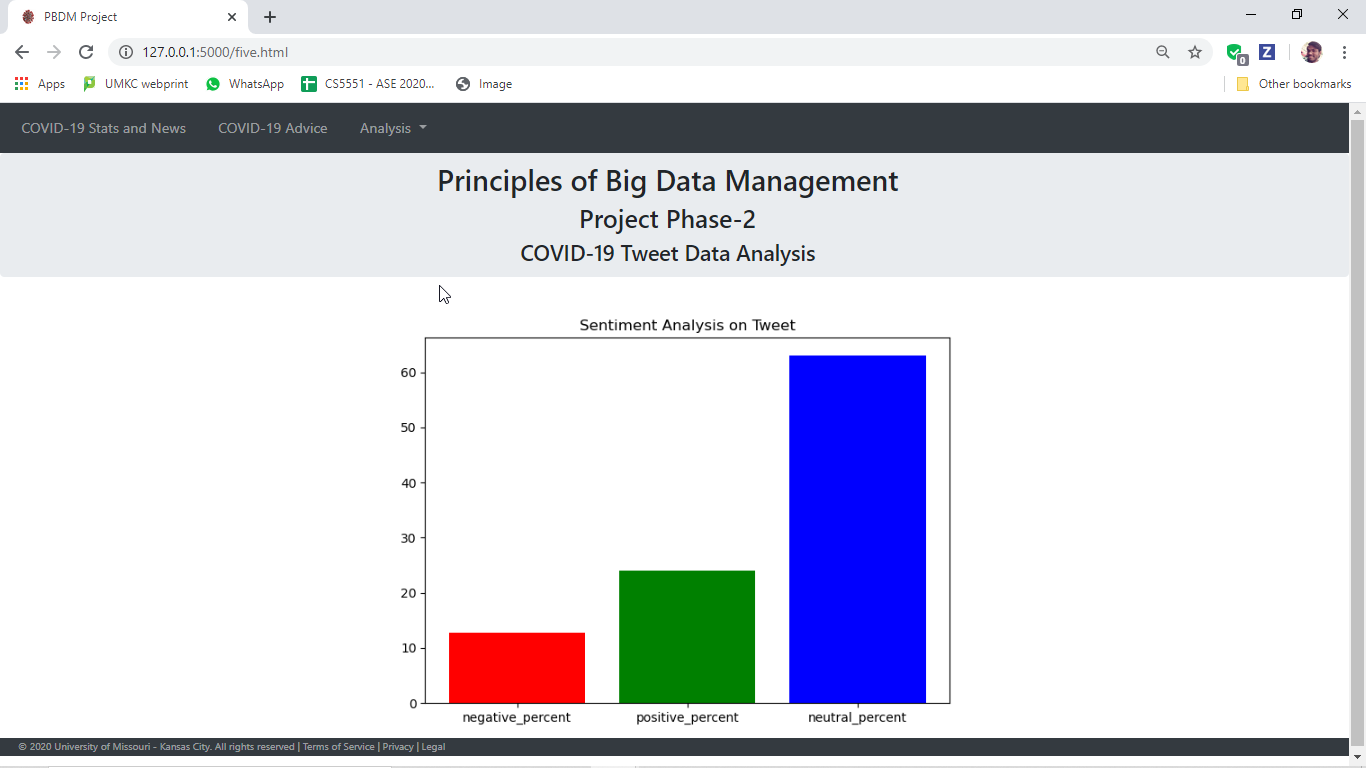
**# Query5**

**# Query to perform sentiment analysis on tweet based on keywords 'covid','coronavirus'**

**Sentiment Analysis on Tweet**



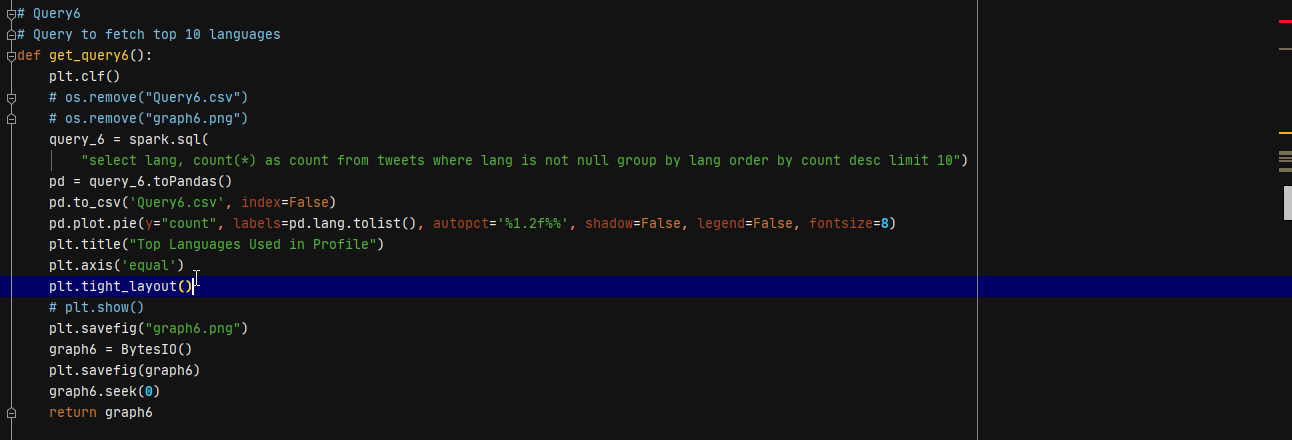
We have used some TextBlob, analysis Package. Using sentiment pre-defined library to perform sentiment analysis on the text. Generating bar graph based on the type whether positive, negative or neutral, with different color for each.

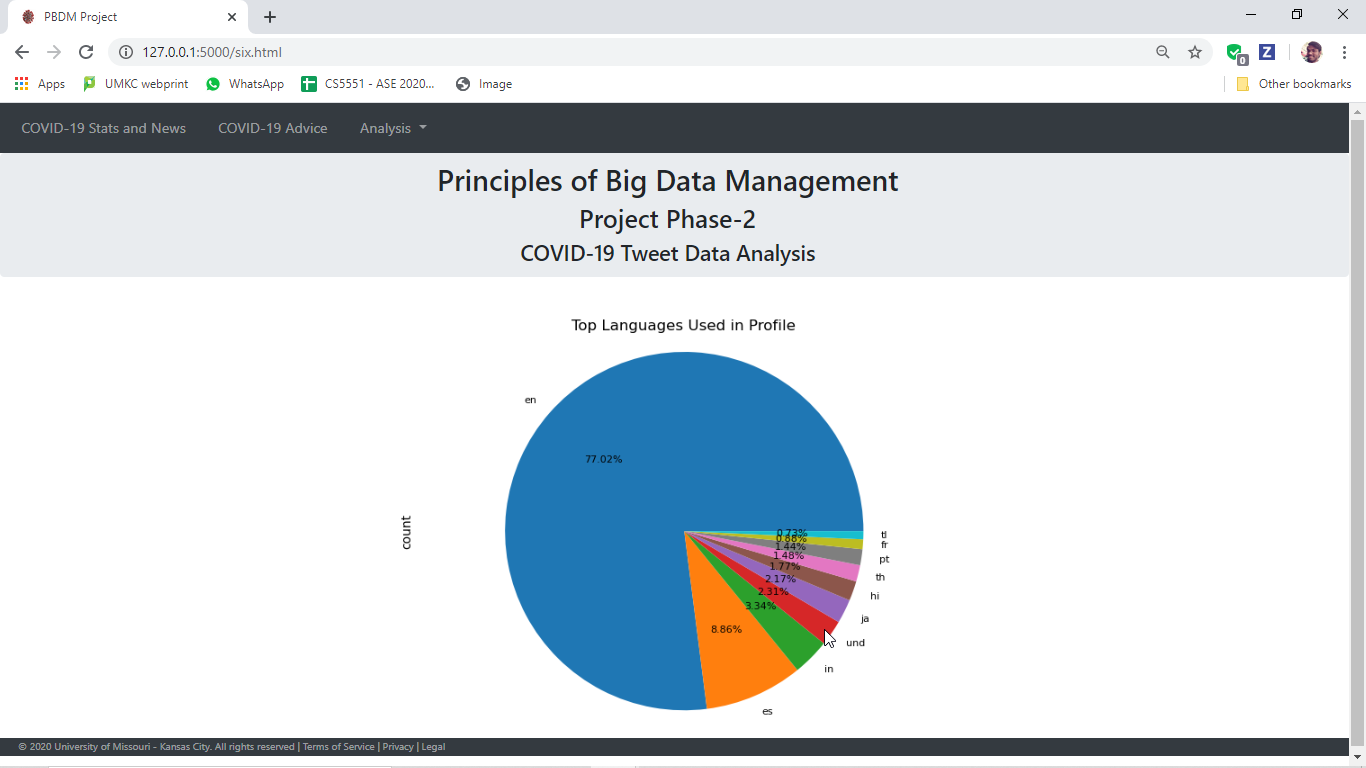


**# Query6**

**# Query to fetch top 10 languages**

**Top Languages Used in Profile**

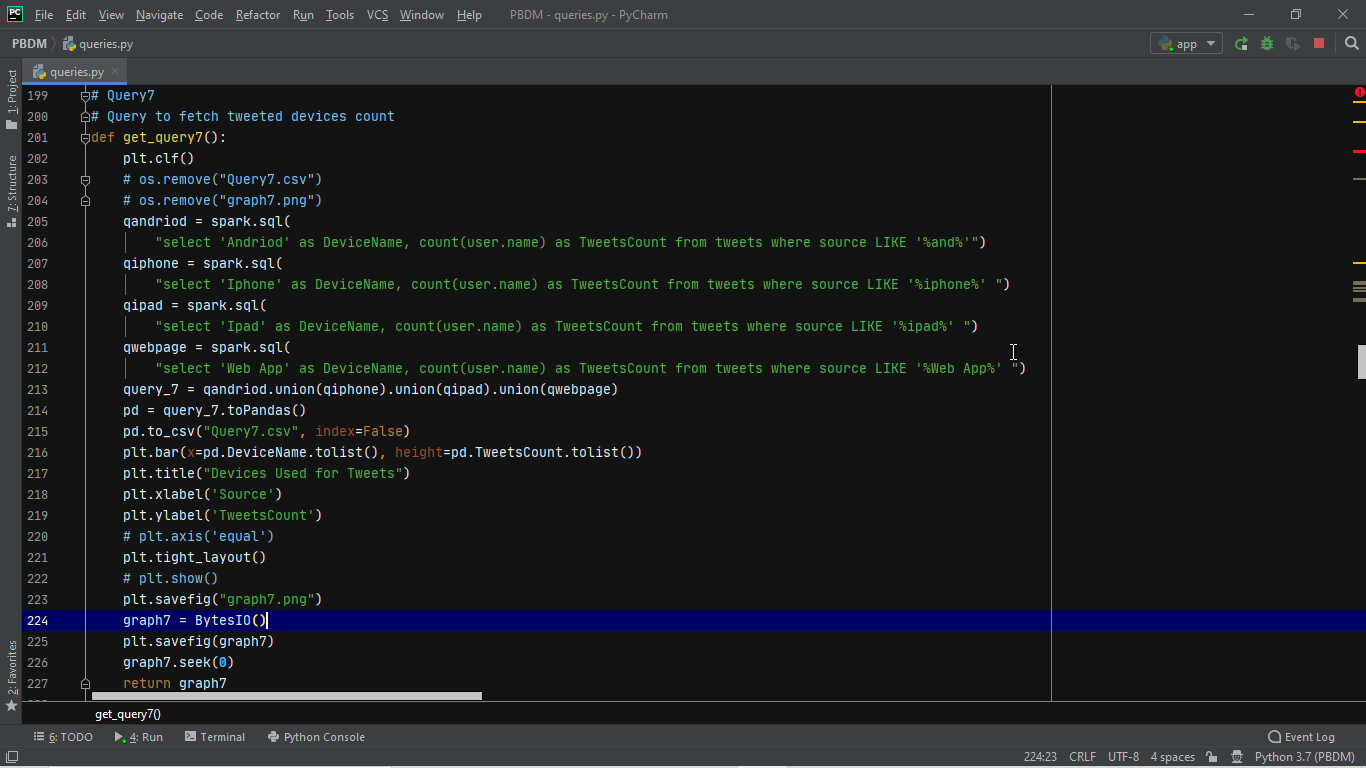


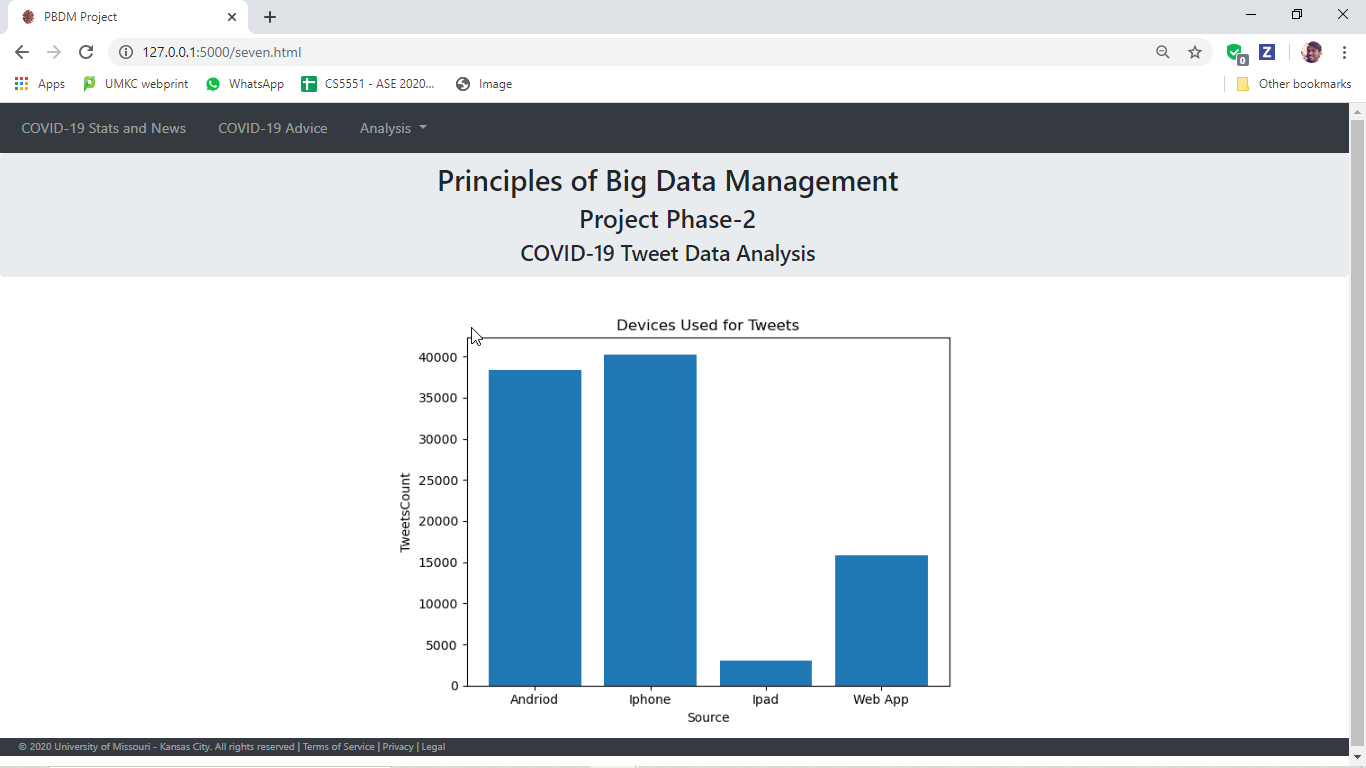


**# Query7**

**# Query to fetch tweeted devices count**

**Devices Used for Tweets**

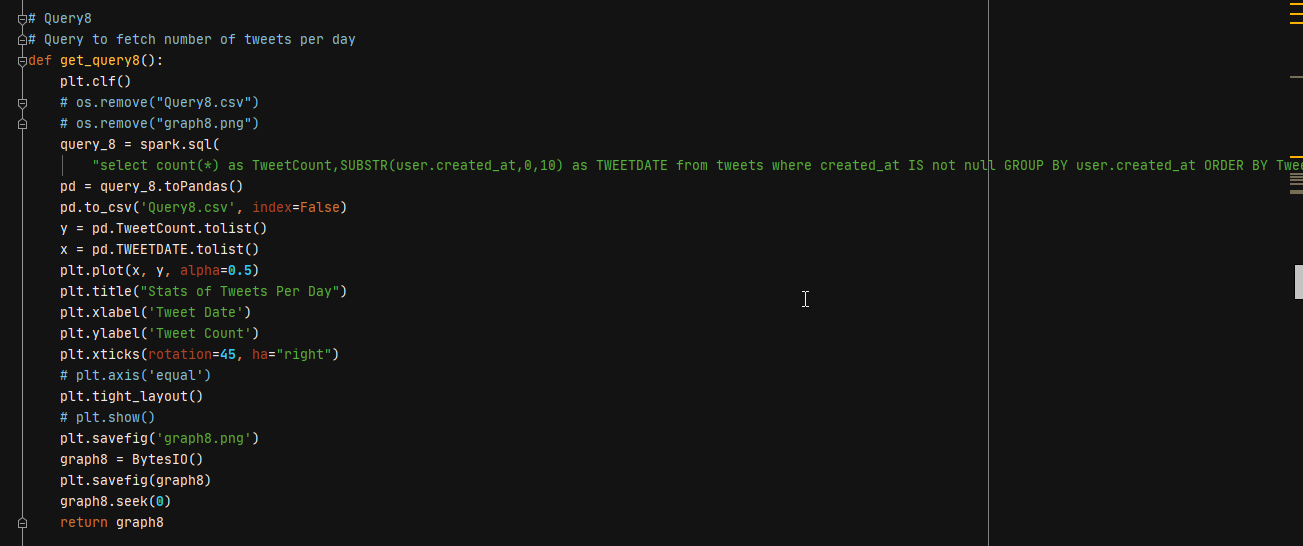


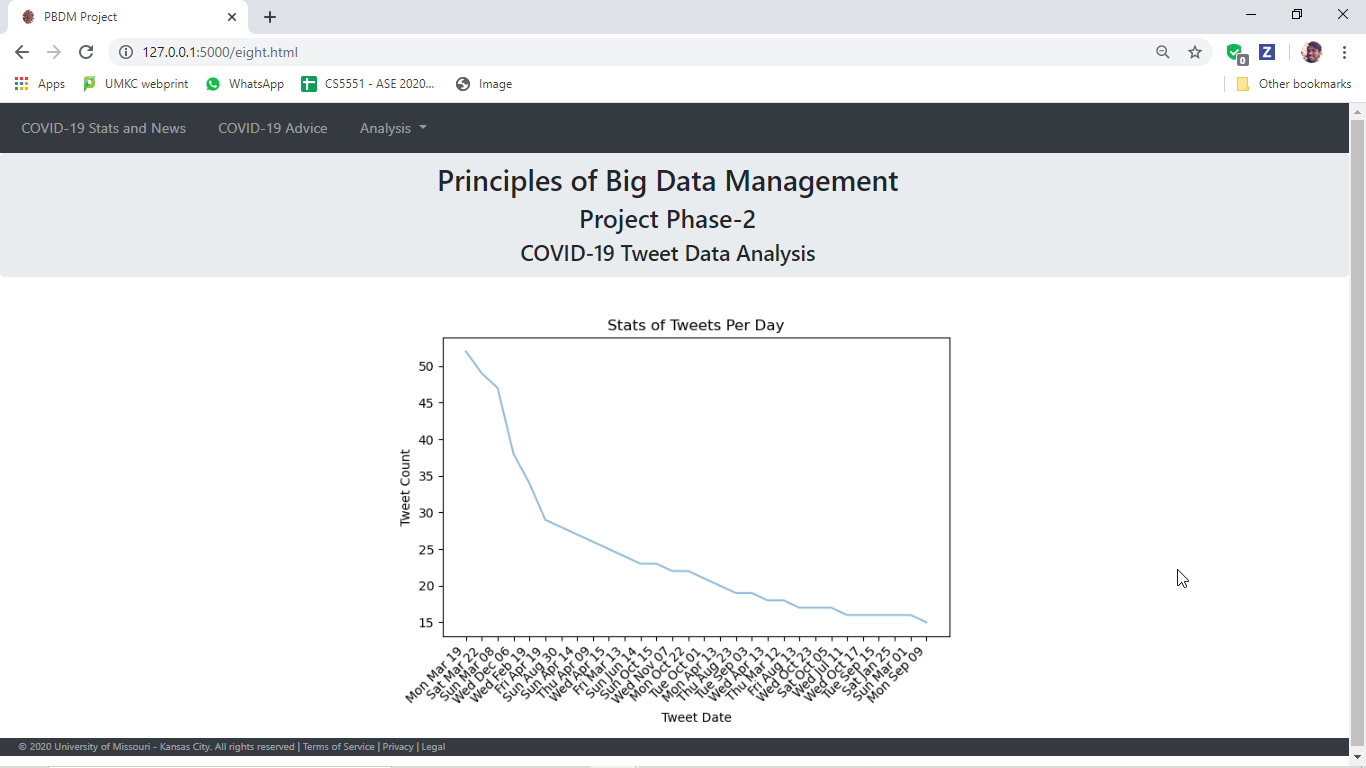


**# Query8**

**# Query to fetch number of tweets per day**

**Stats of Tweets Per Day**

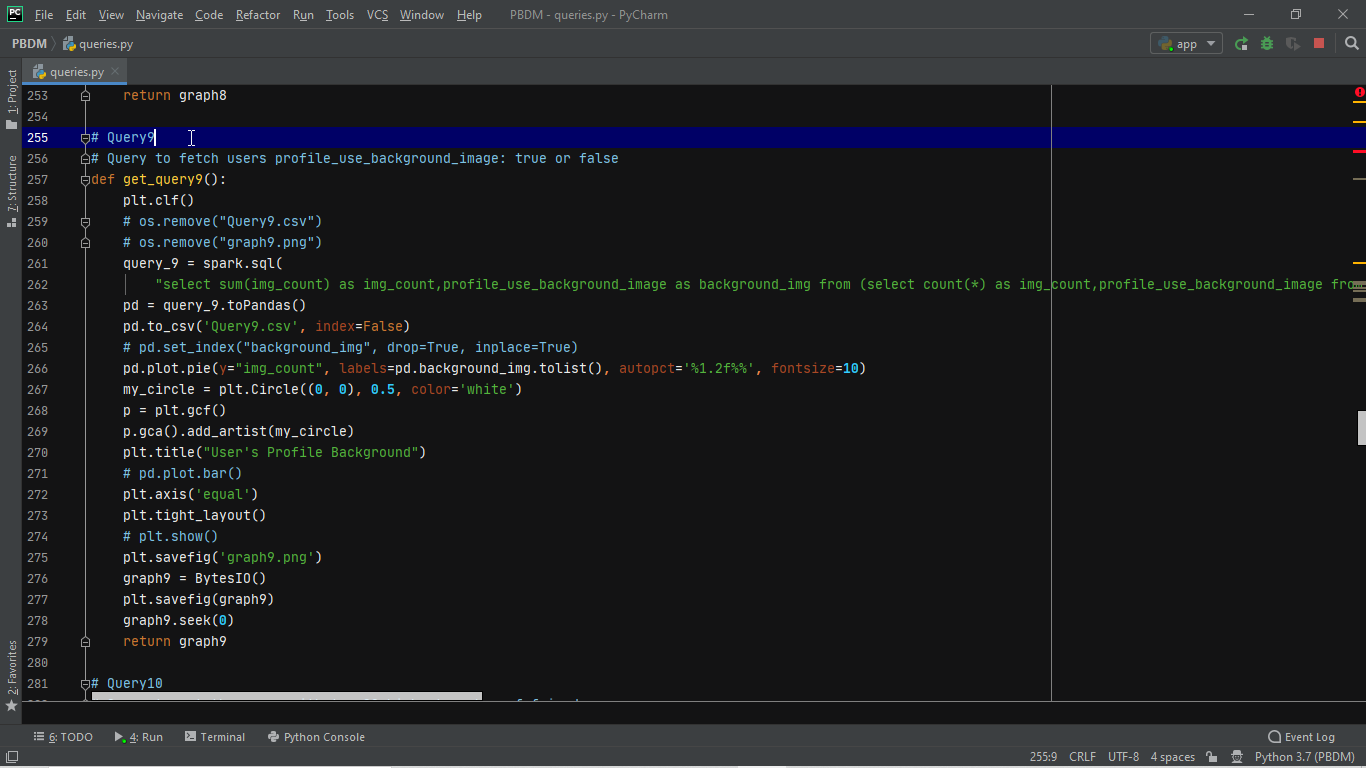


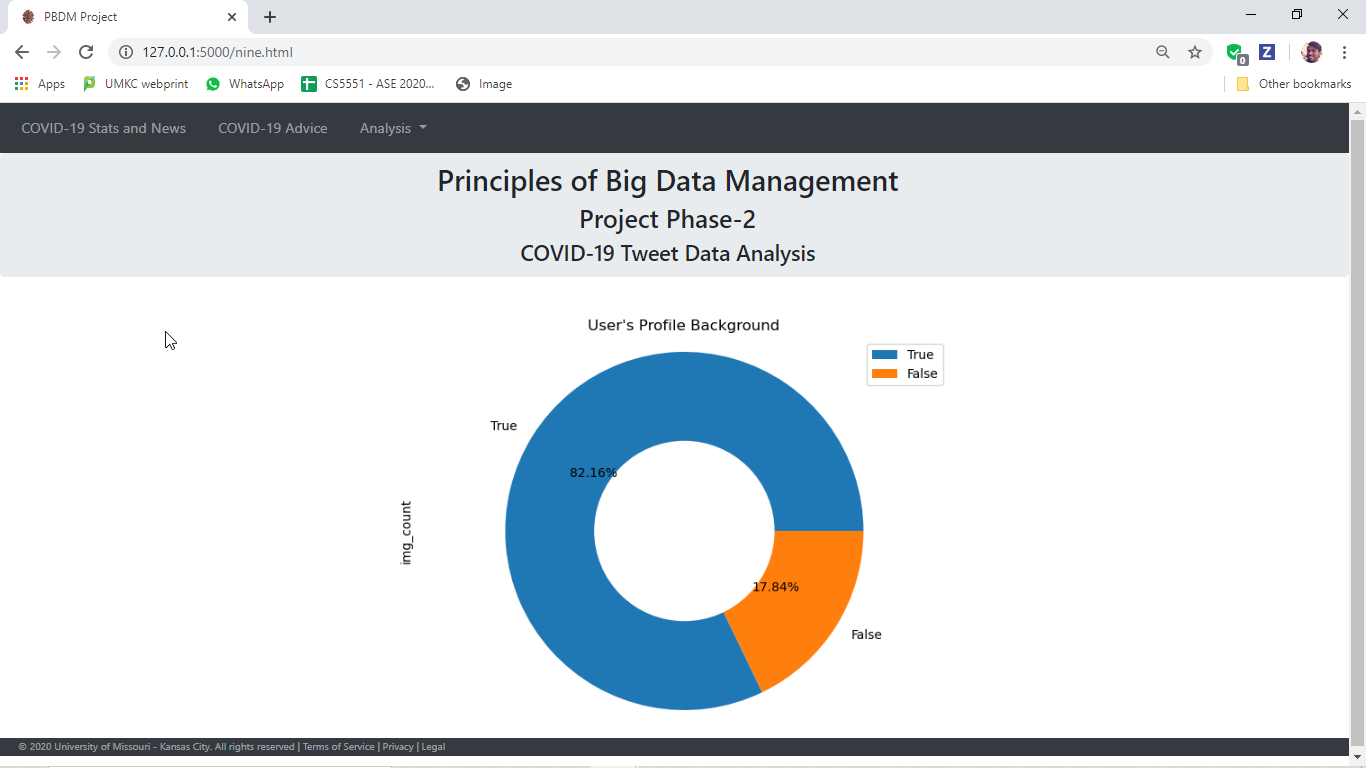


**# Query9**

**# Query to fetch users profile\_use\_background\_image: true or false**

**User's Profile Background**

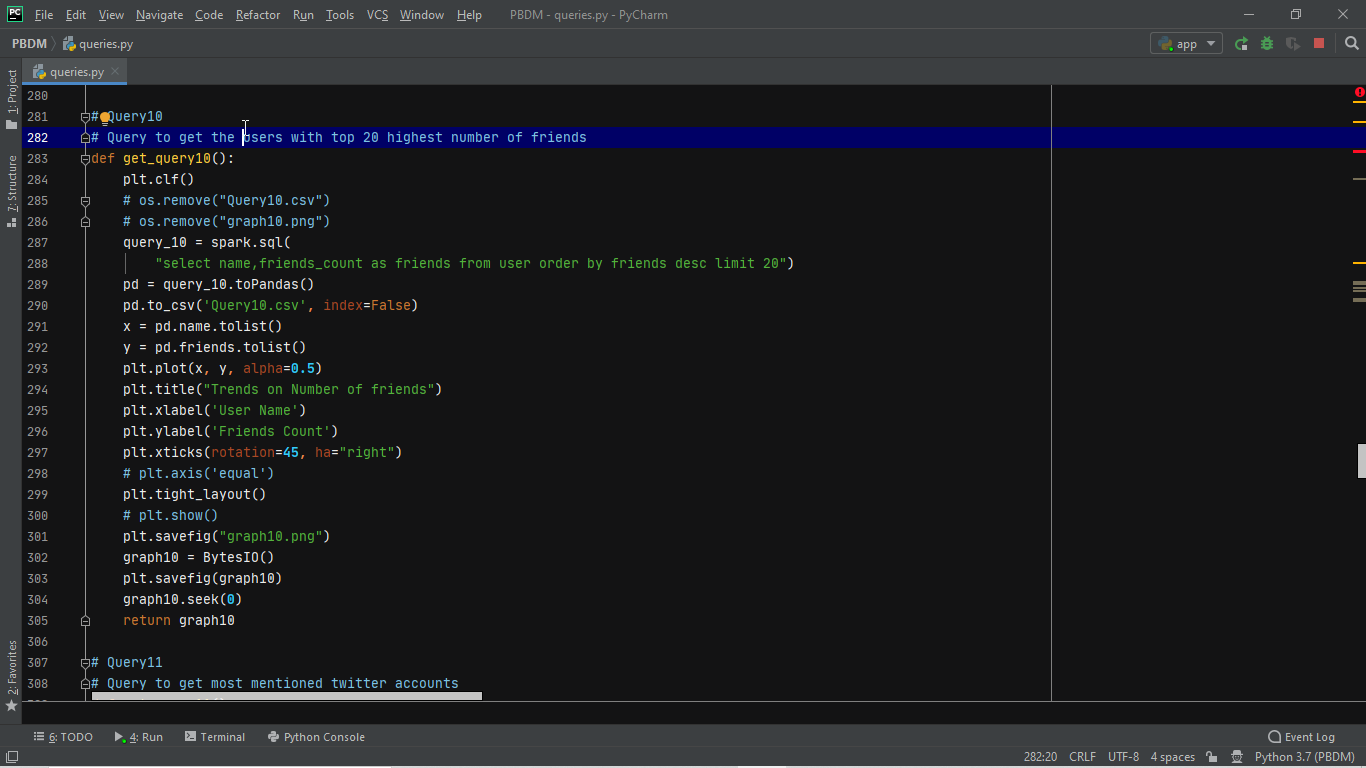


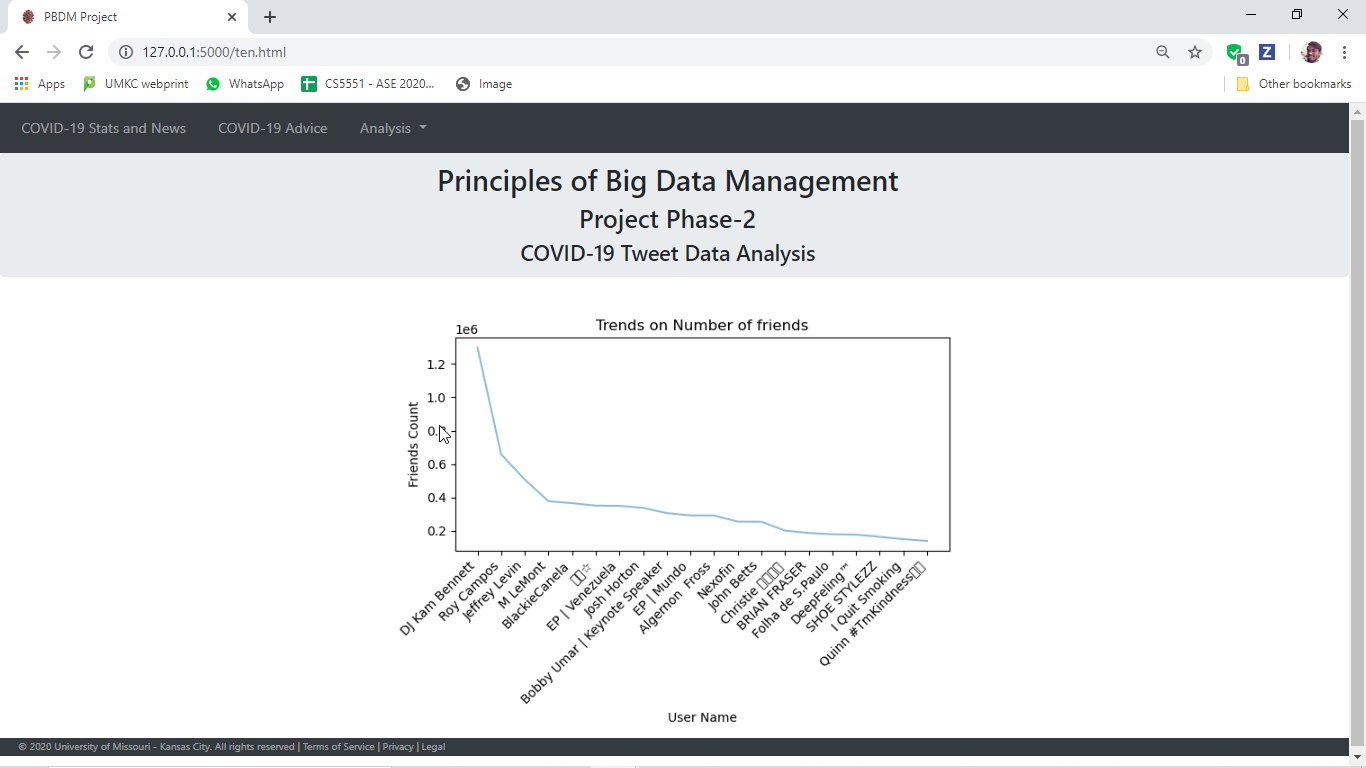


**# Query10**

**# Query to get the users with top 20 highest number of friends**

**Trends on Number of friends**

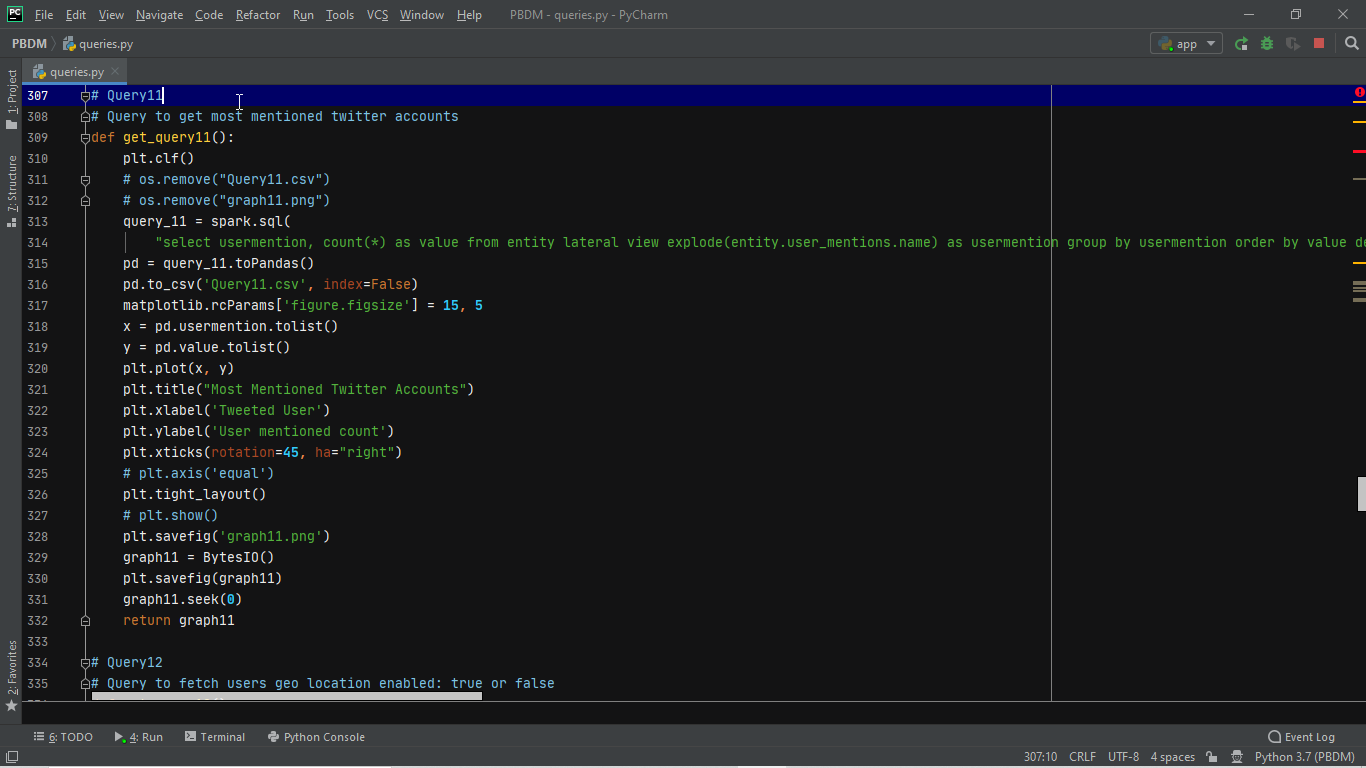


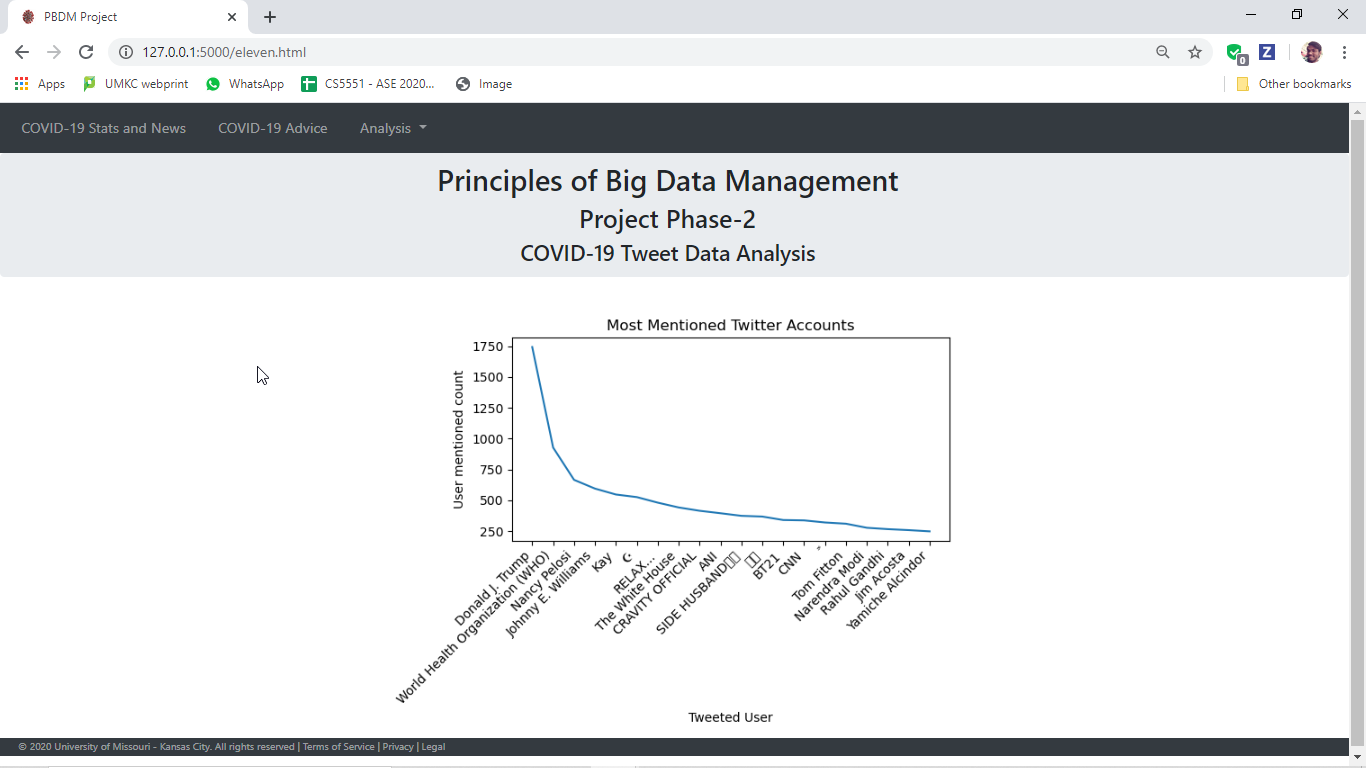


**# Query11**

**# Query to get most mentioned twitter accounts**

**Most Mentioned Twitter Accounts**

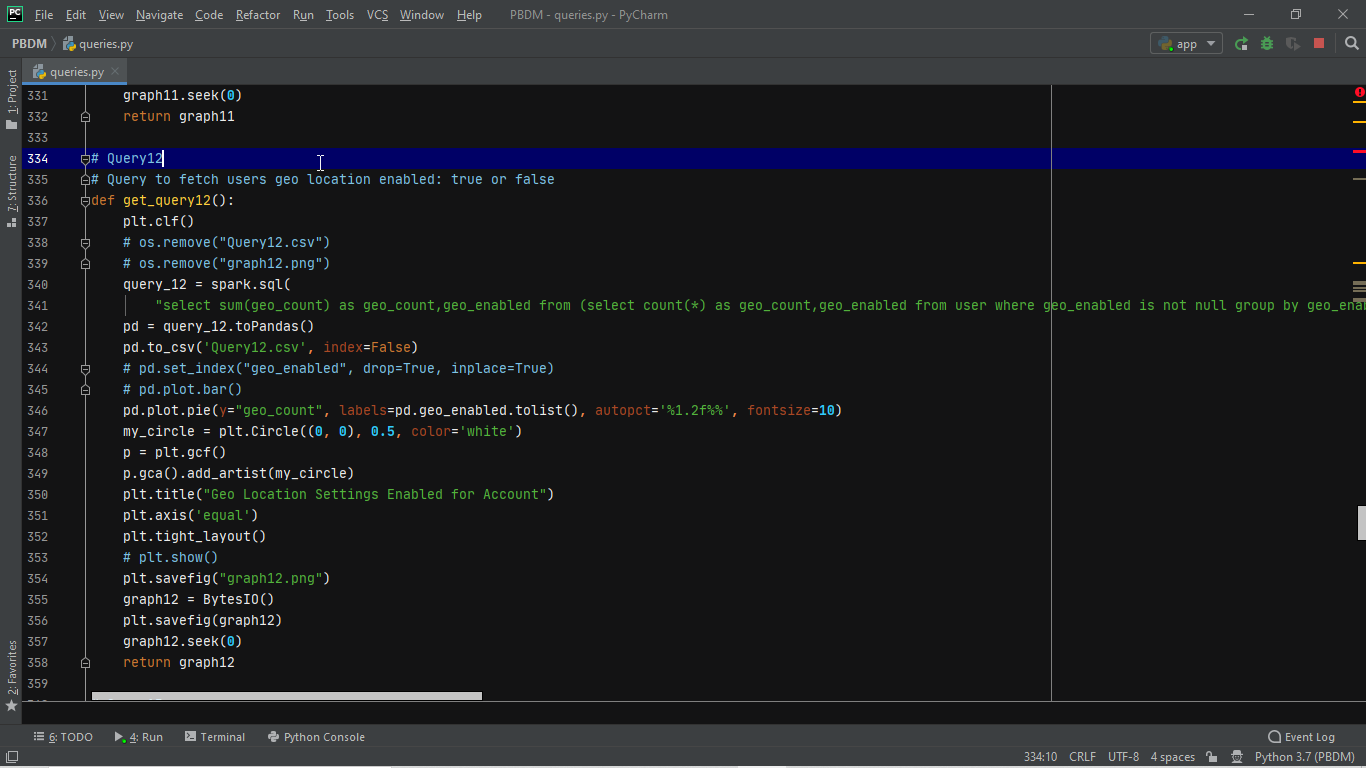


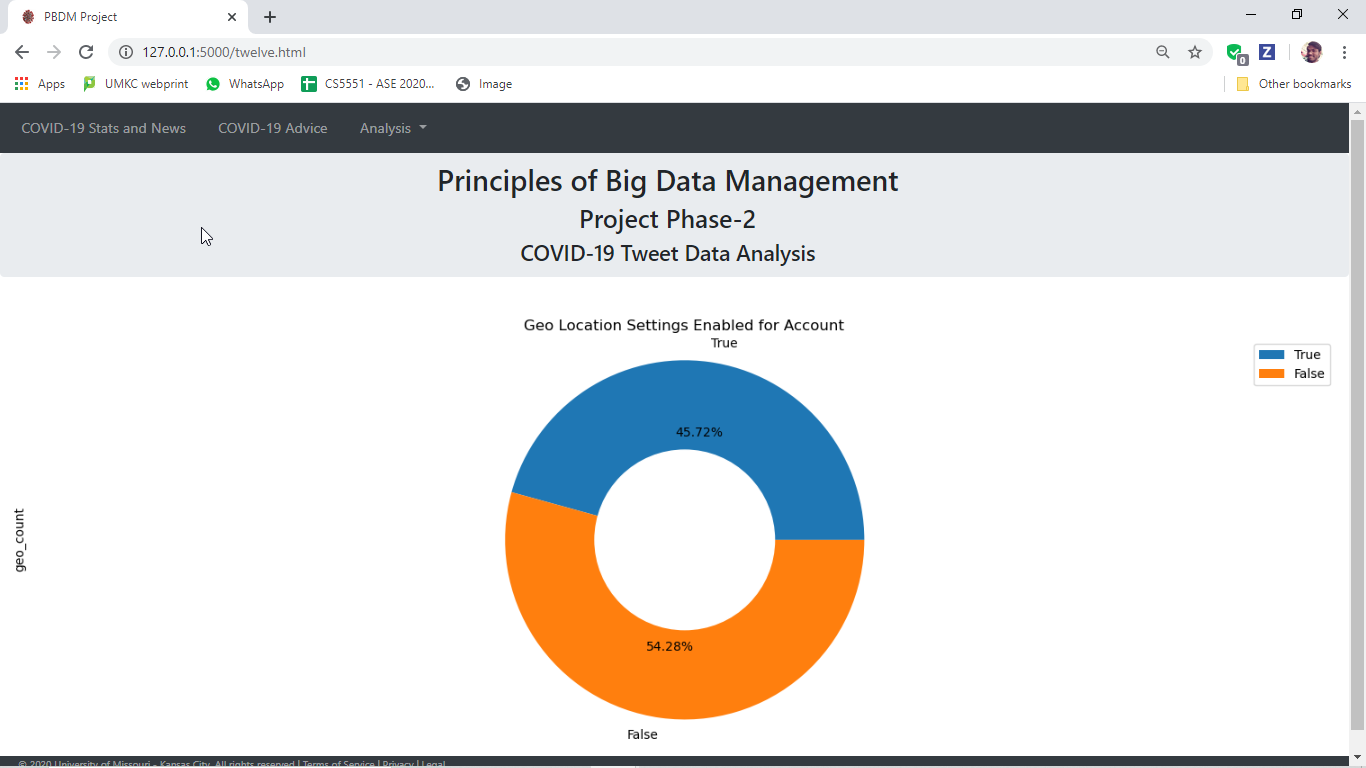


**# Query12**

**# Query to fetch users geo location enabled: true or false**

**Geo Location Settings Enabled for Account**

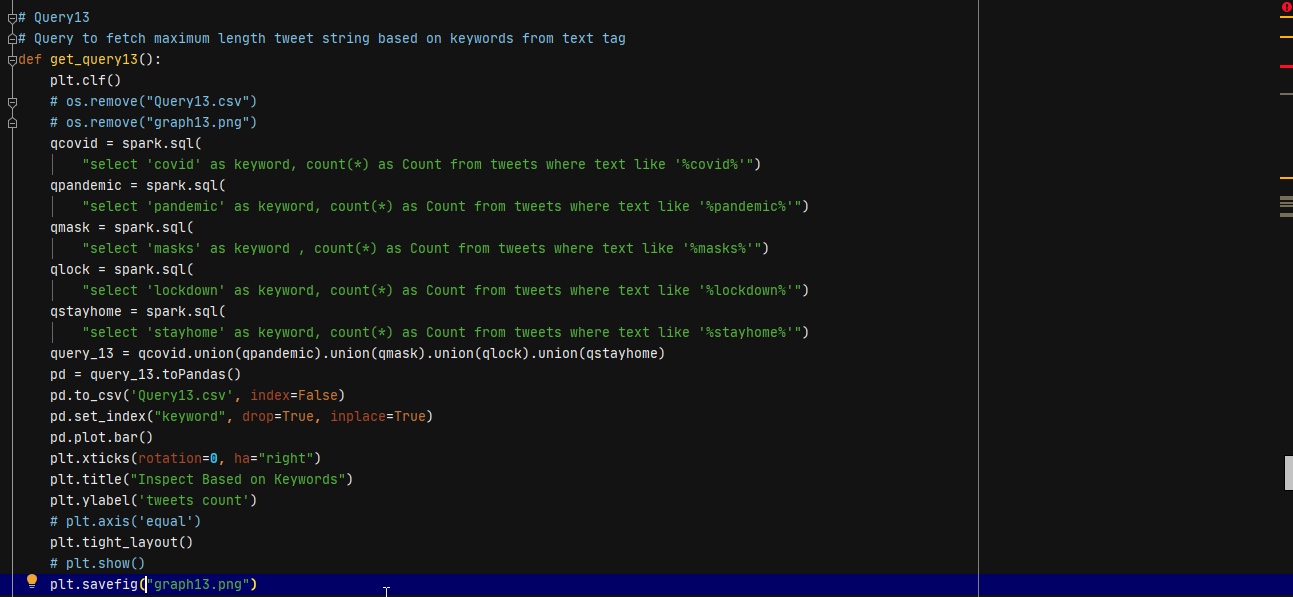


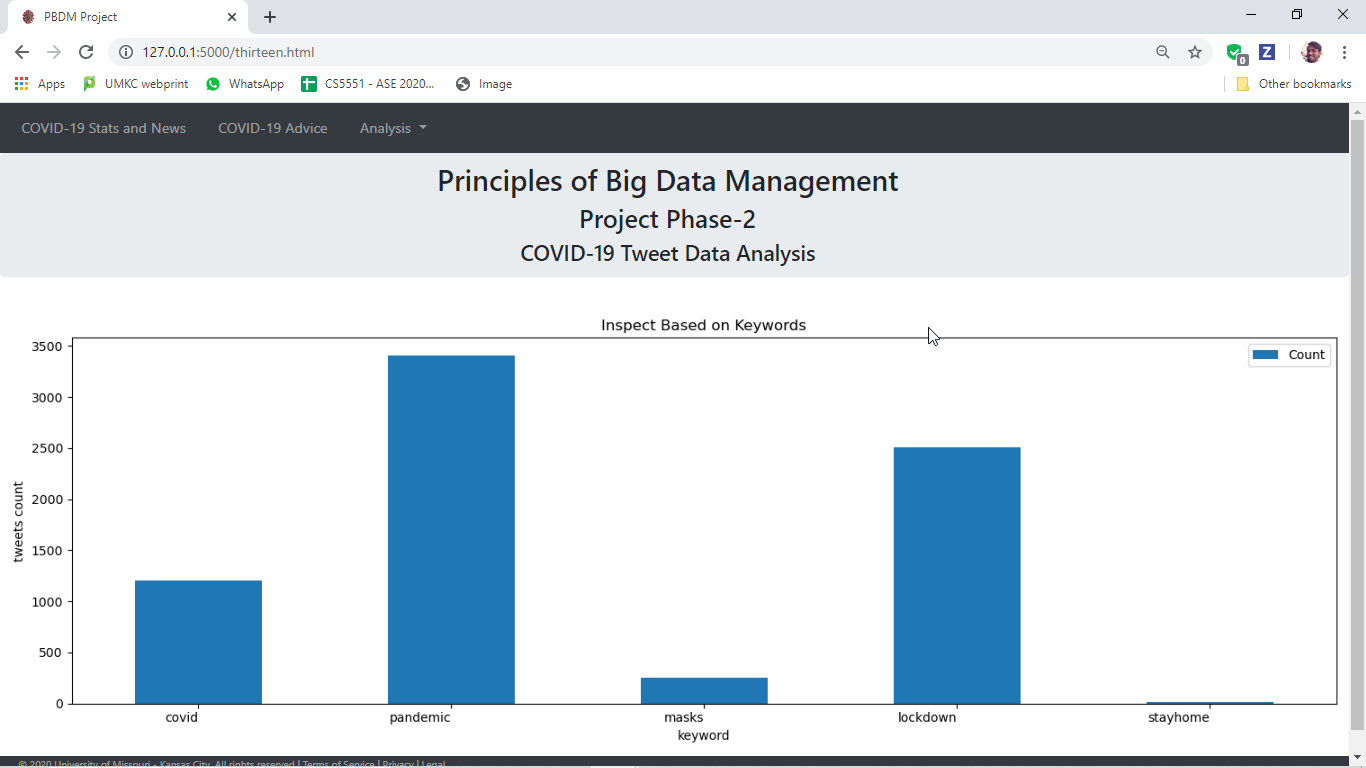


**# Query13**

**# Query to fetch maximum length tweet string based on keywords from text tag**

**Inspect Based on Keywords**

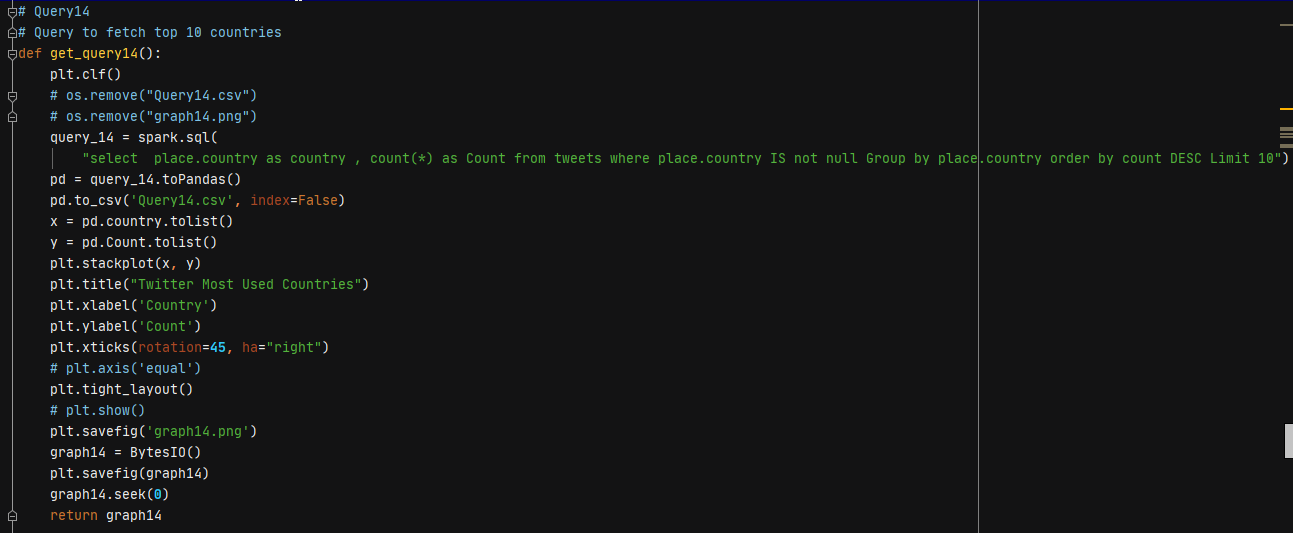
 

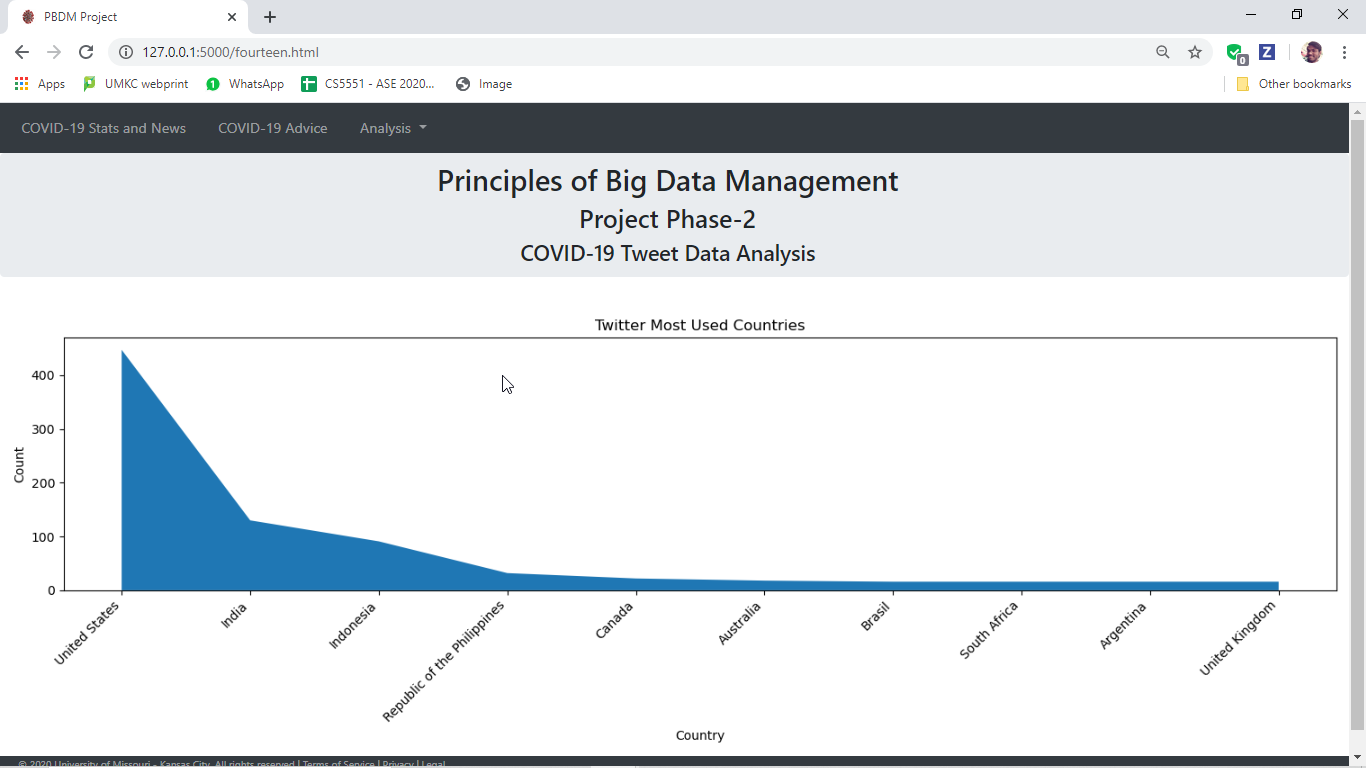


**# Query14**

**# Query to fetch top 10 countries**

**Twitter Most Used Countries**

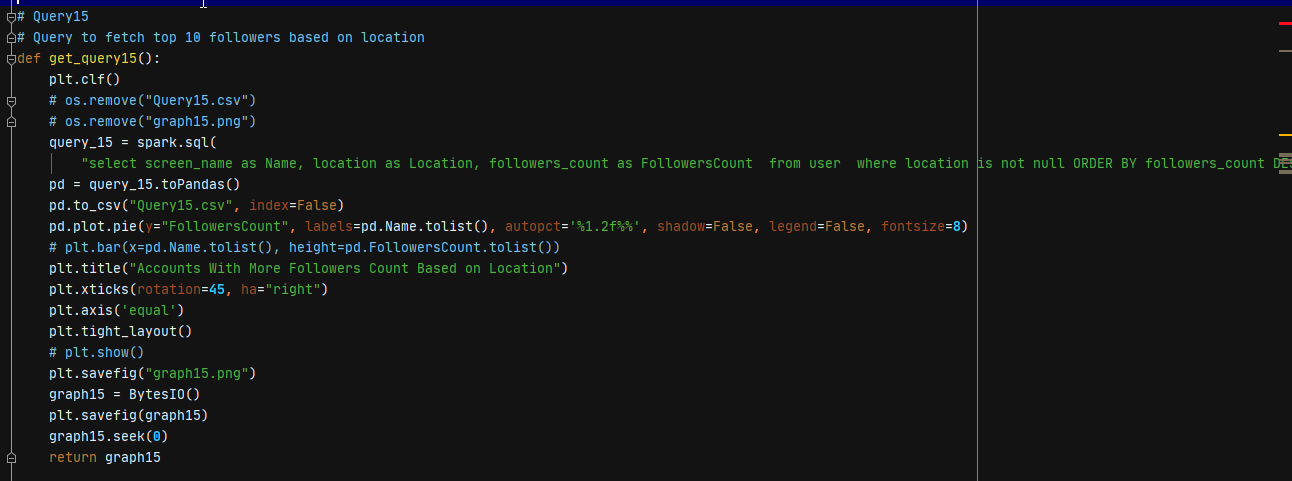


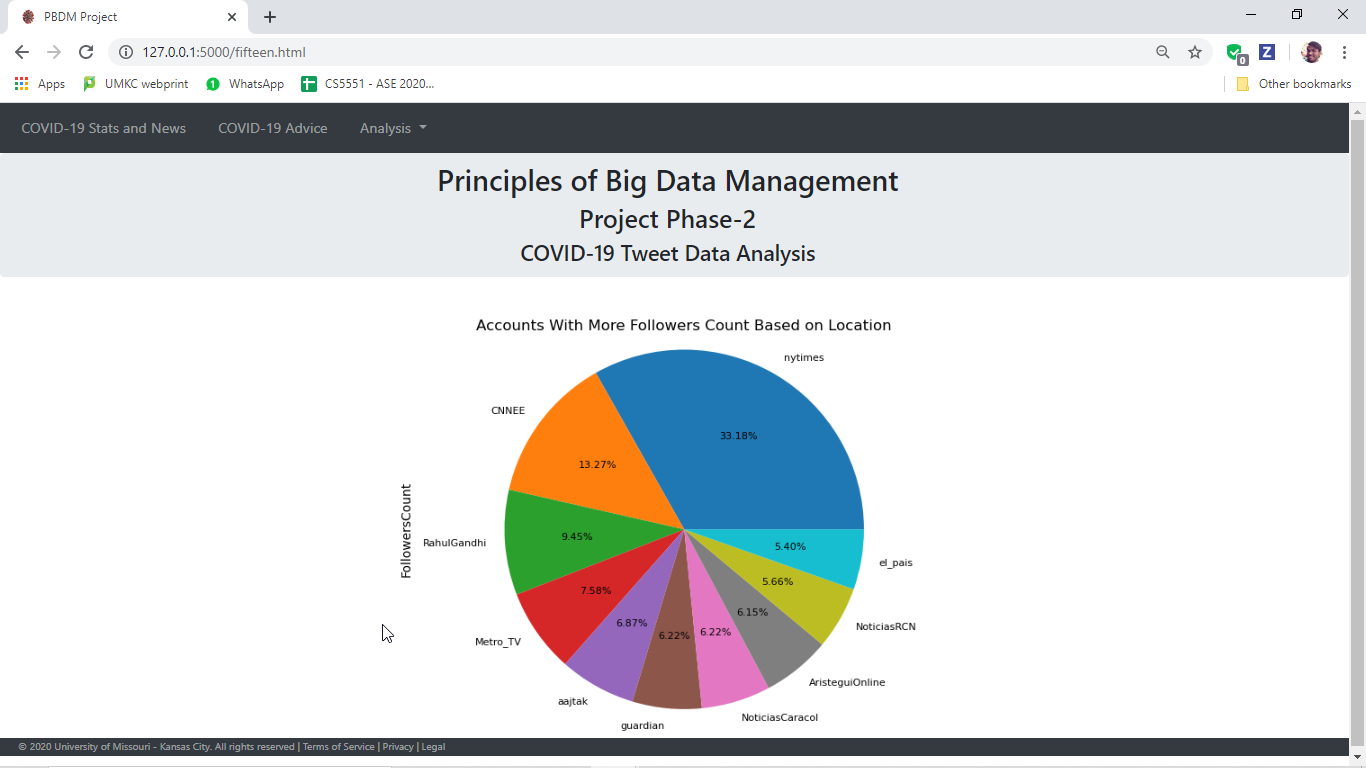


**# Query15**

**# Query to fetch top 10 followers based on location**

**Accounts With More Followers Count Based on Location**

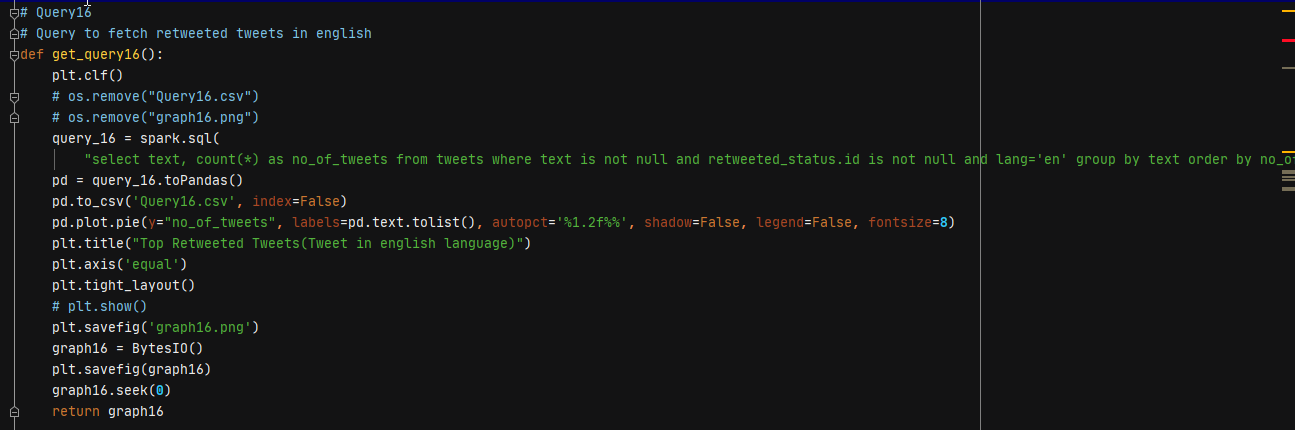


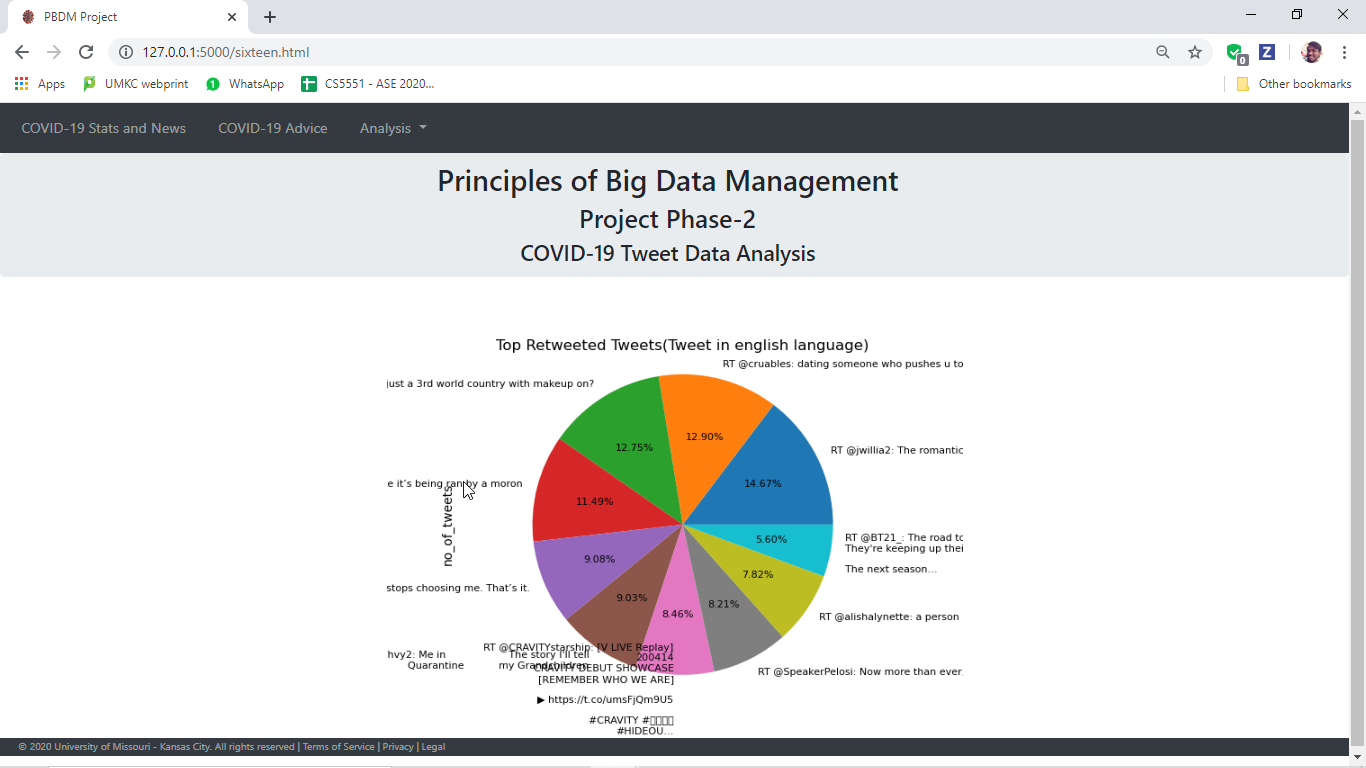


**# Query16**

**# Query to fetch retweeted tweets in English**

**Top Retweeted Tweets(Tweet in english language)**



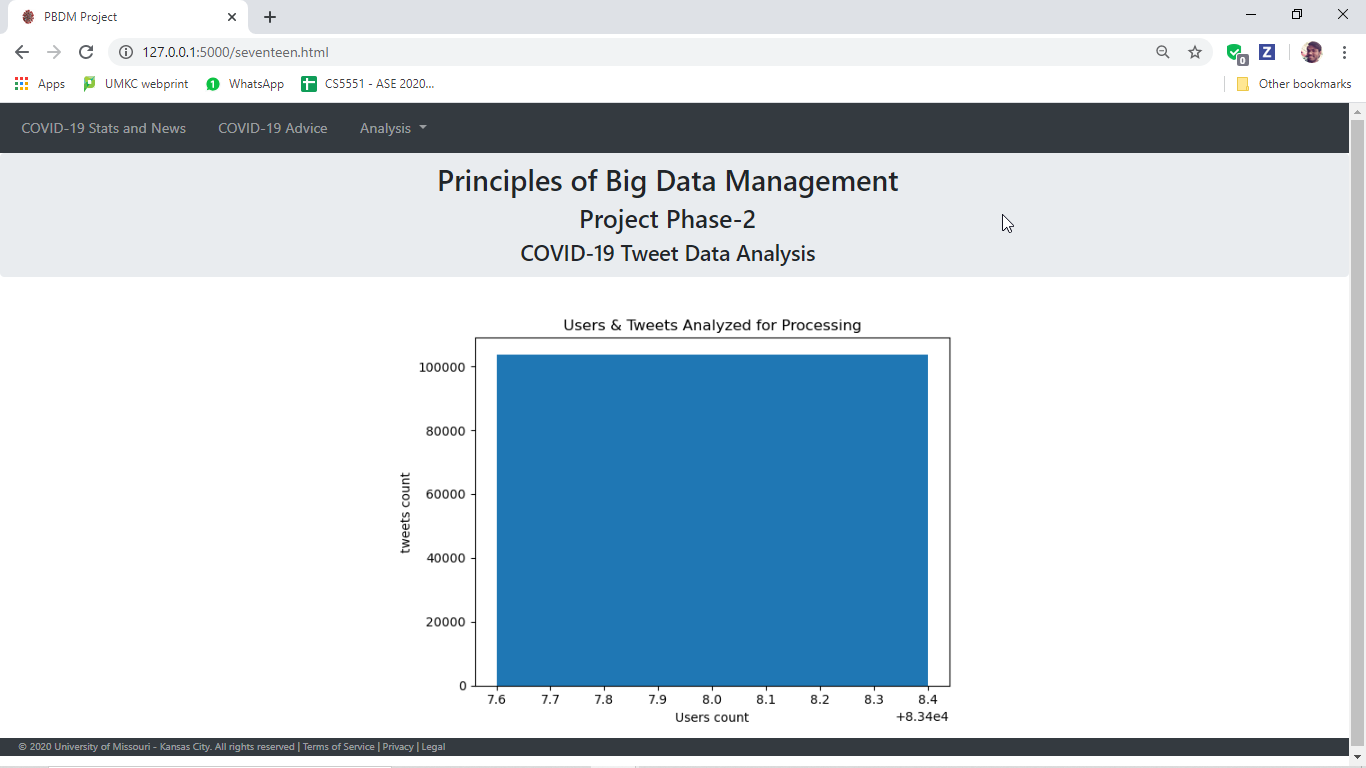


**# Query17**

**# Query to fetch number of tweets and users collected**

**Users & Tweets Analyzed for Processing**





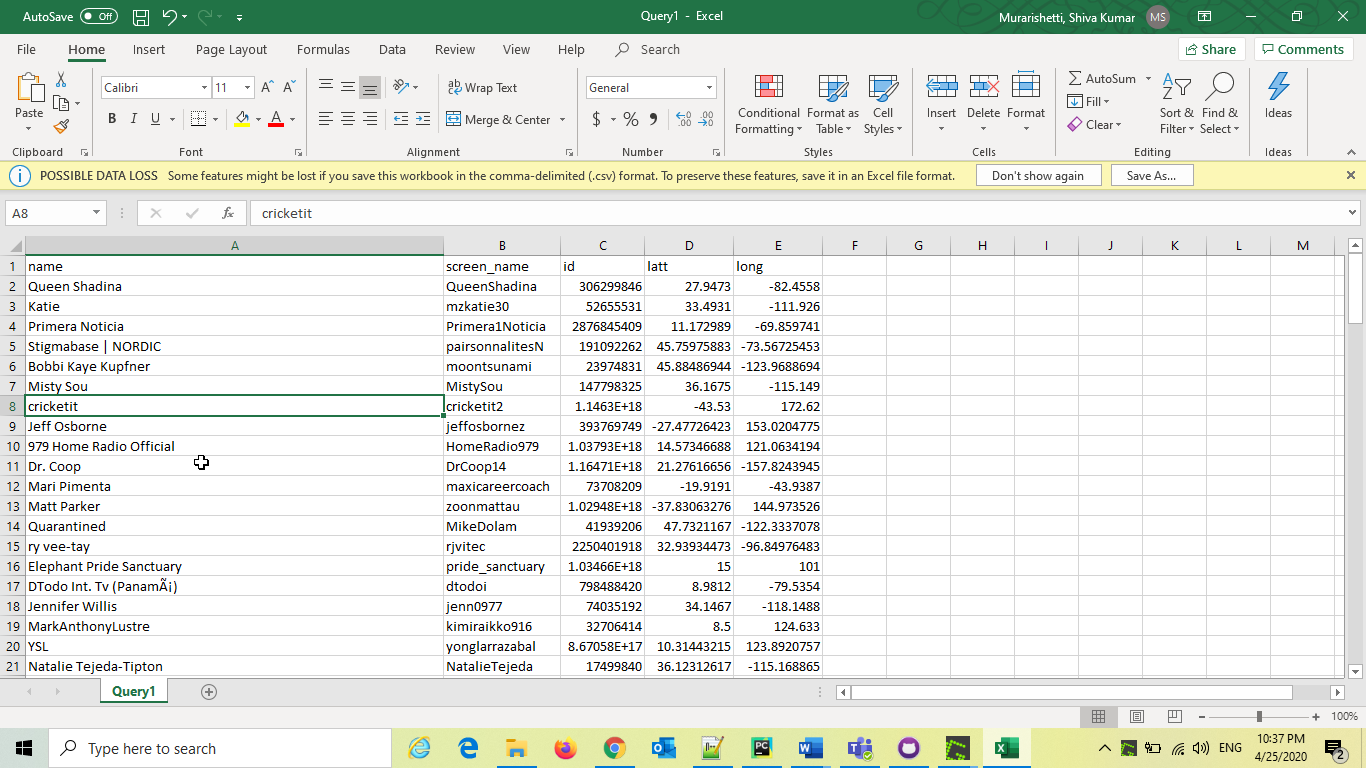
**Testing:**

We have tested queries by running, with small amount of data initially and writing the query result into a .csv file. When we see that the data generated in .csv file is appropriate, we implemented graphs on them and then implemented on larger tweet data to run the project.

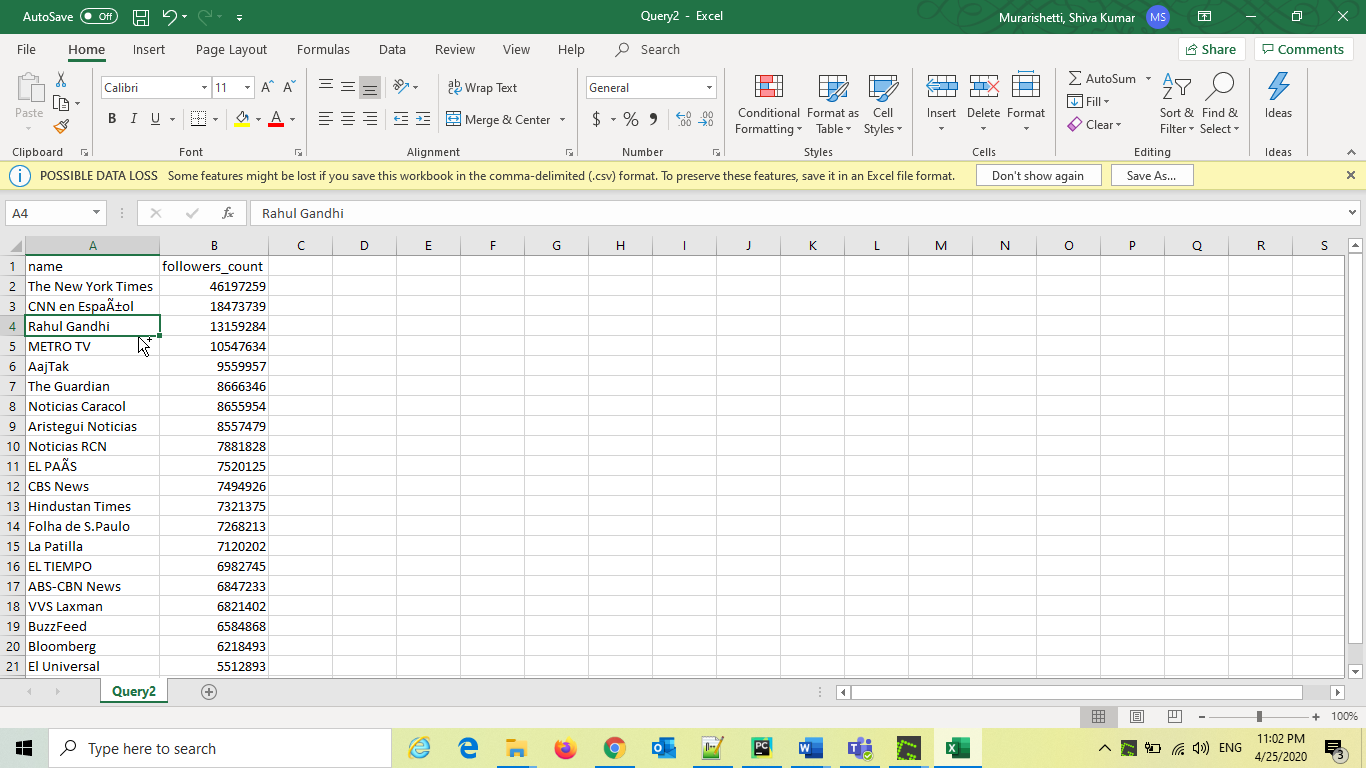
Test Tweet Data file: <https://drive.google.com/file/d/1eMSMShS8h8EpGgXyc9exktIuAJoTM7tk/view?usp=sharing>

Tweet Data Used after testing file: <https://drive.google.com/file/d/116XKRNEKs1aBrX-H4LJh3Tyg0QScETcP/view?usp=sharing>

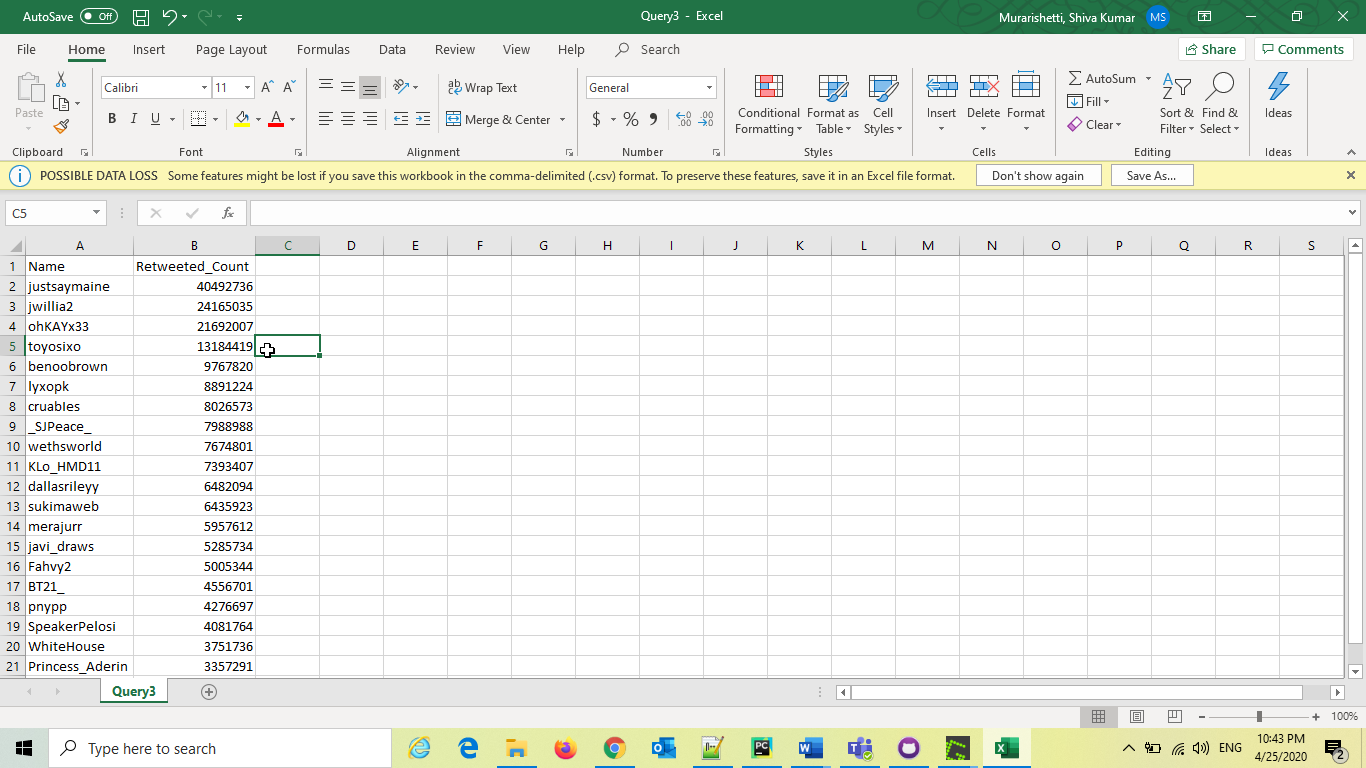
**Query 1 Test**



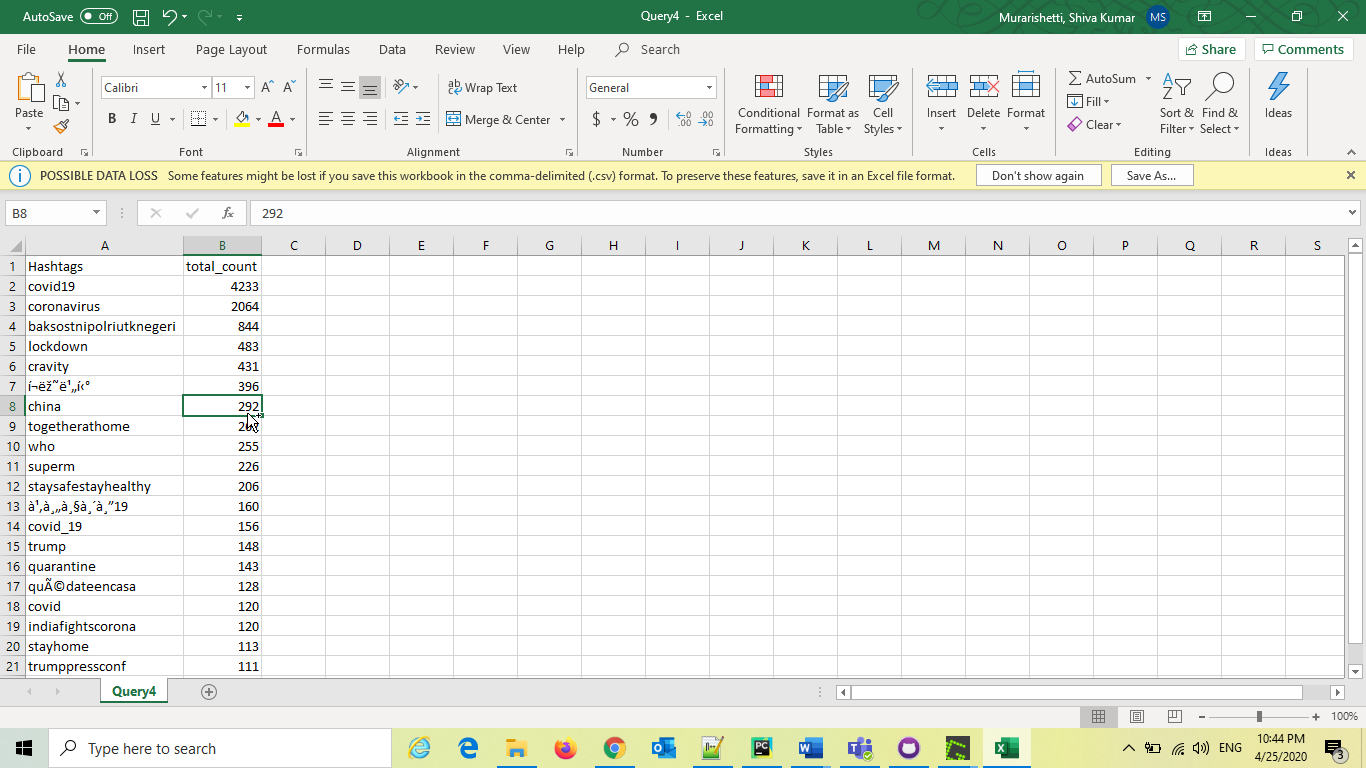
**Query 2 Test**



**Query 3 Test**

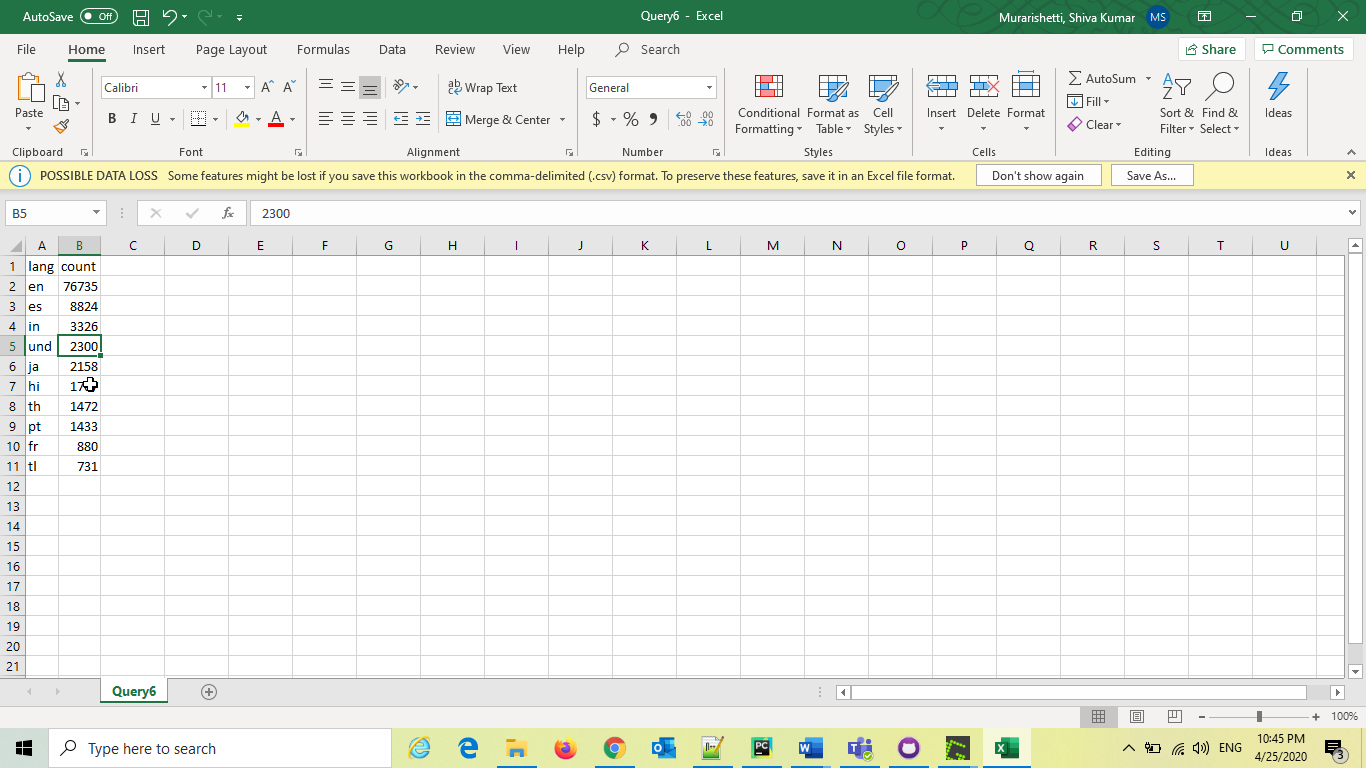


**Query 4 Test**

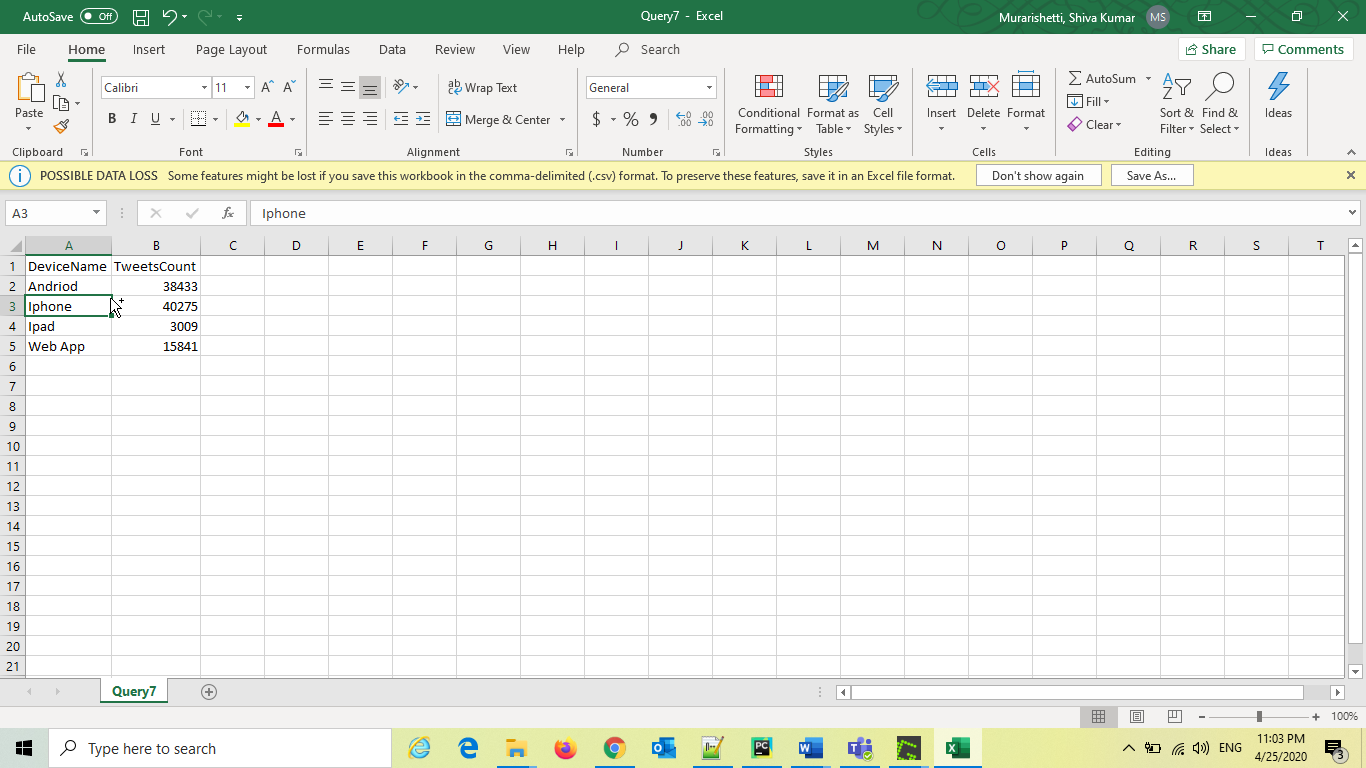


**Query 5 Test:** As we have used pre-defined libraries for sentimental analysis, we have not tested it by loading into .csv file. We directly generated graph for it.

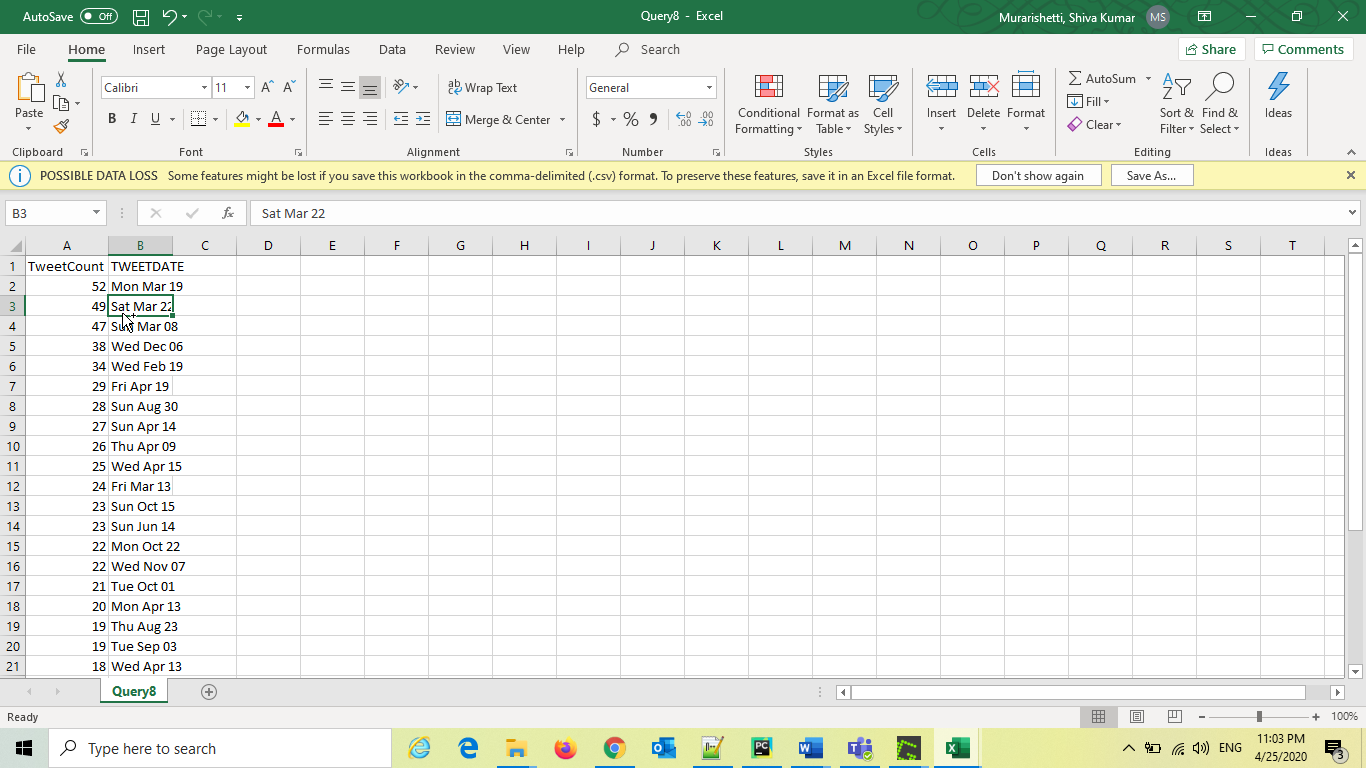
**Query 6 Test**



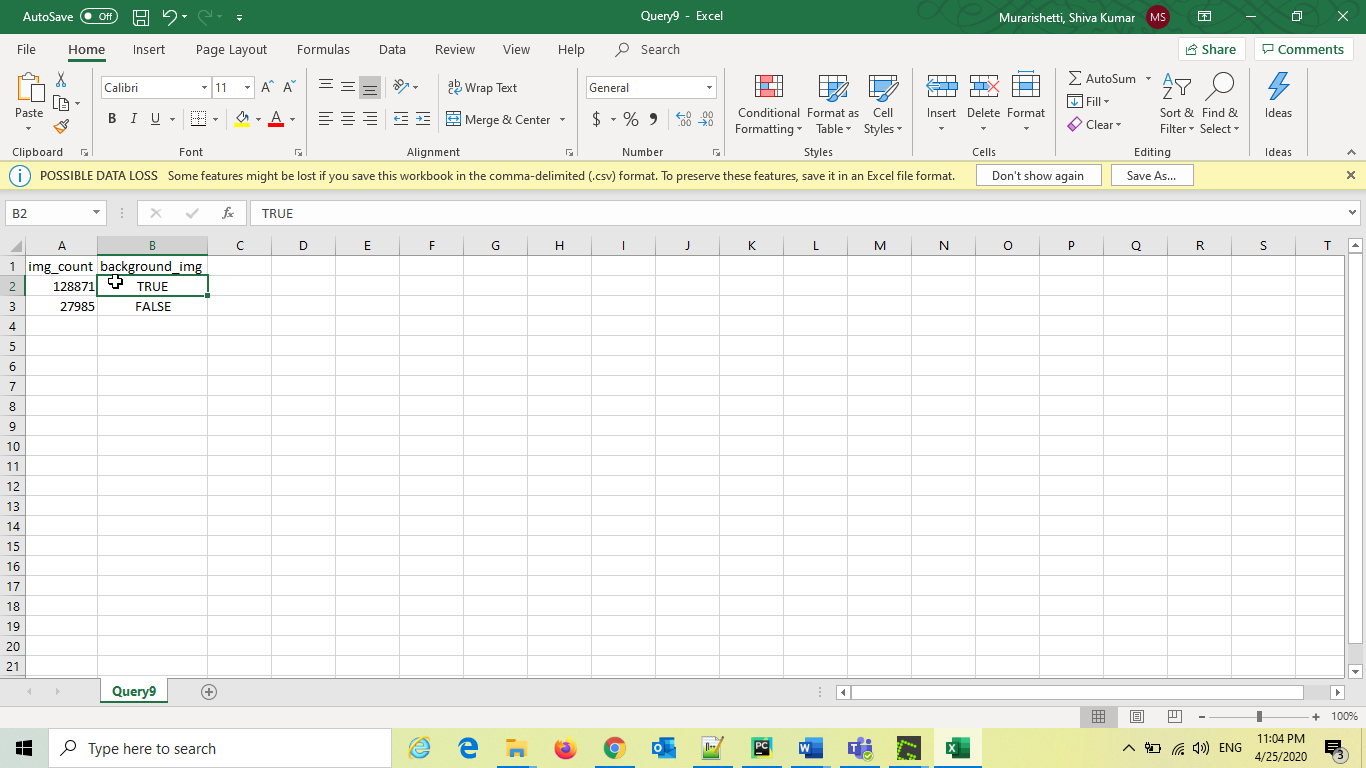
**Query 7 Test**



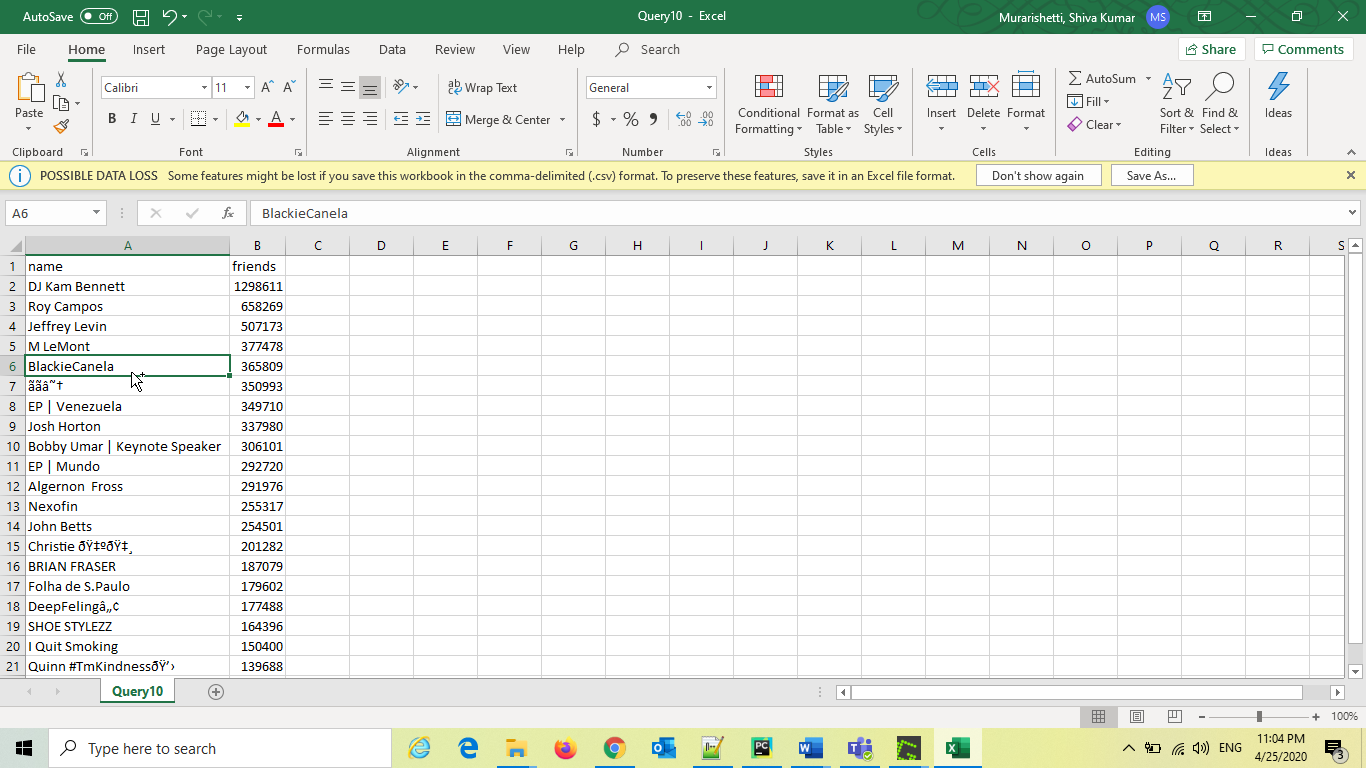
**Query 8 Test**



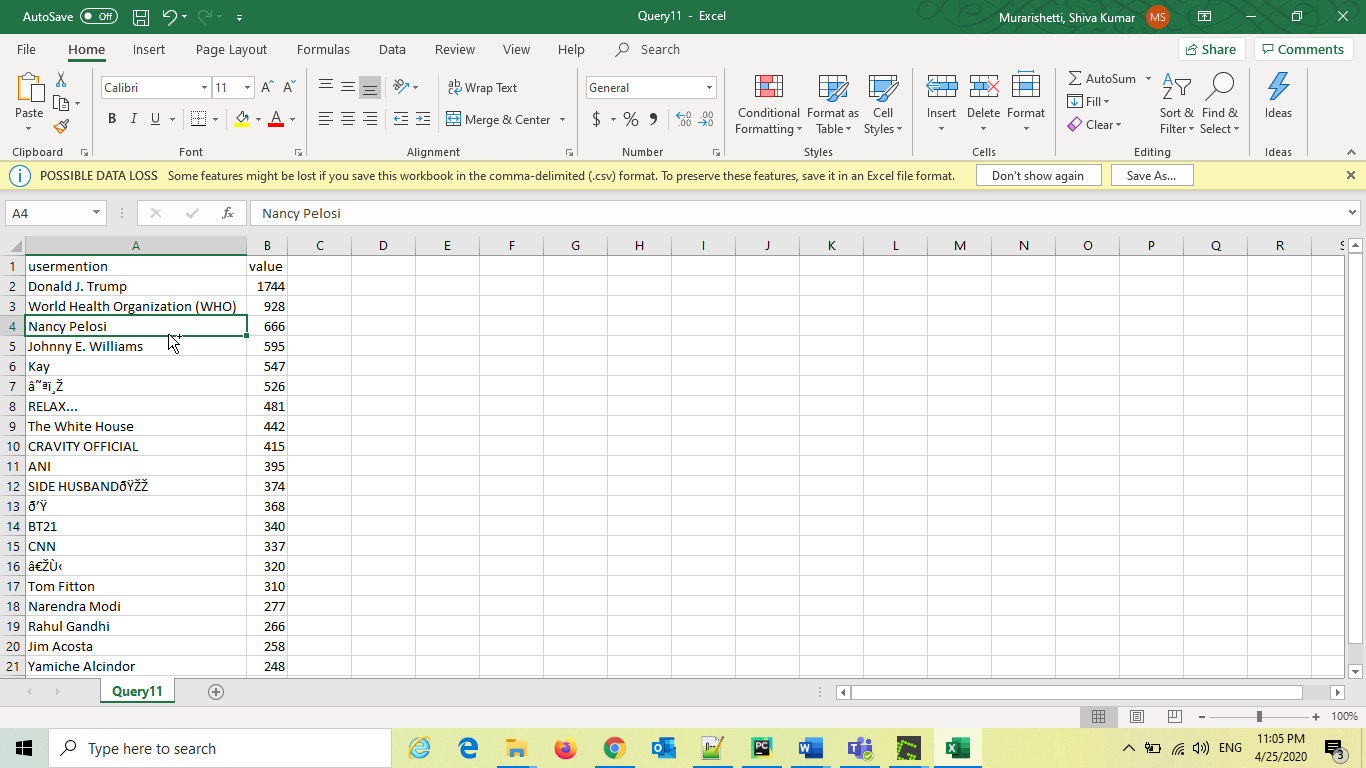
**Query 9 Test**



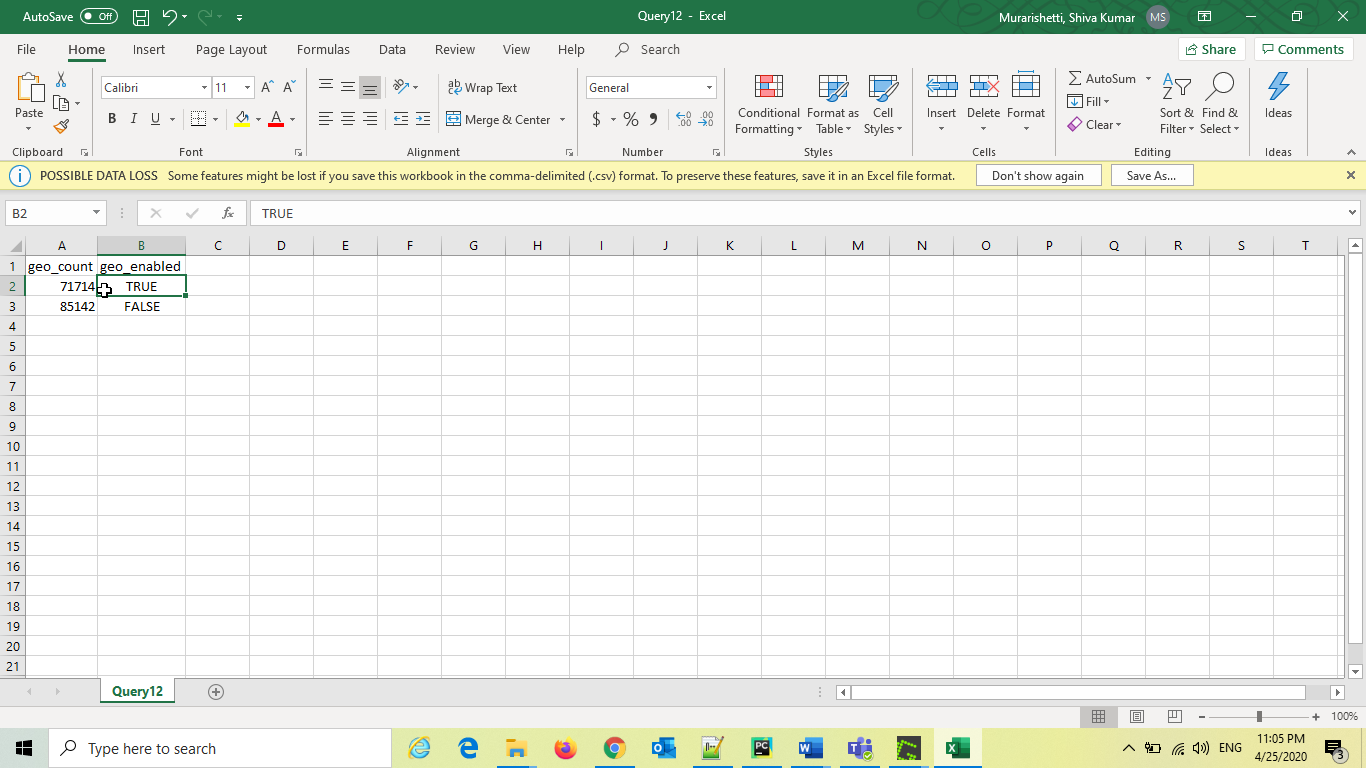
**Query 10 Test**



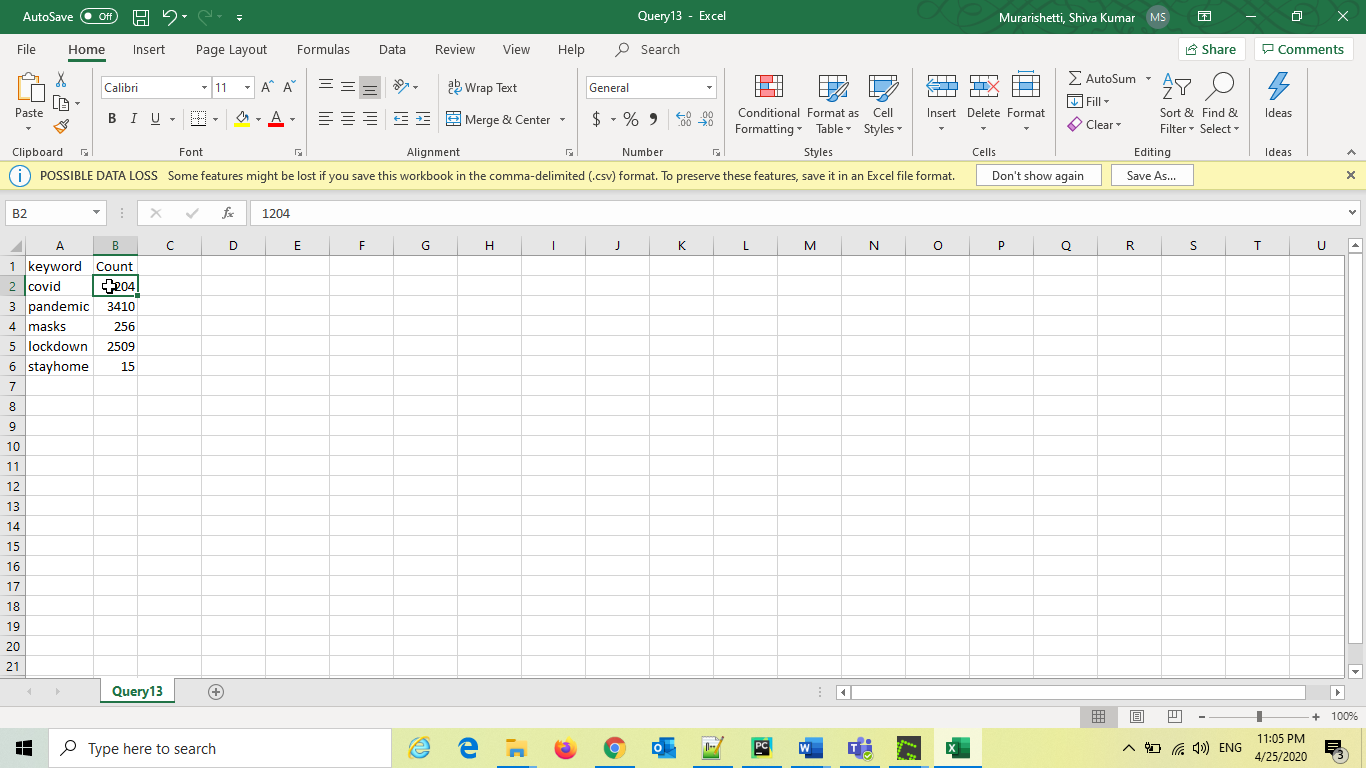
**Query 11 Test**



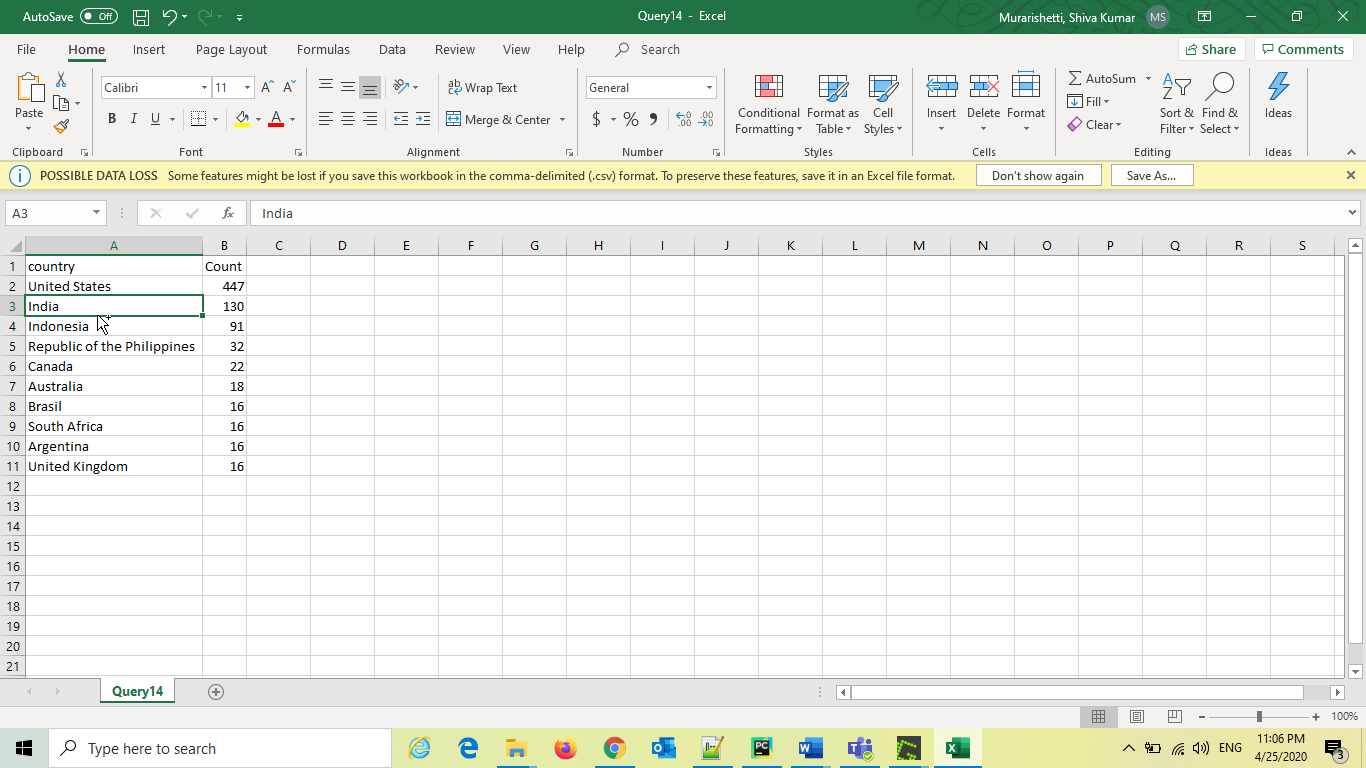
**Query 12 Test**



**Query 13 Test**



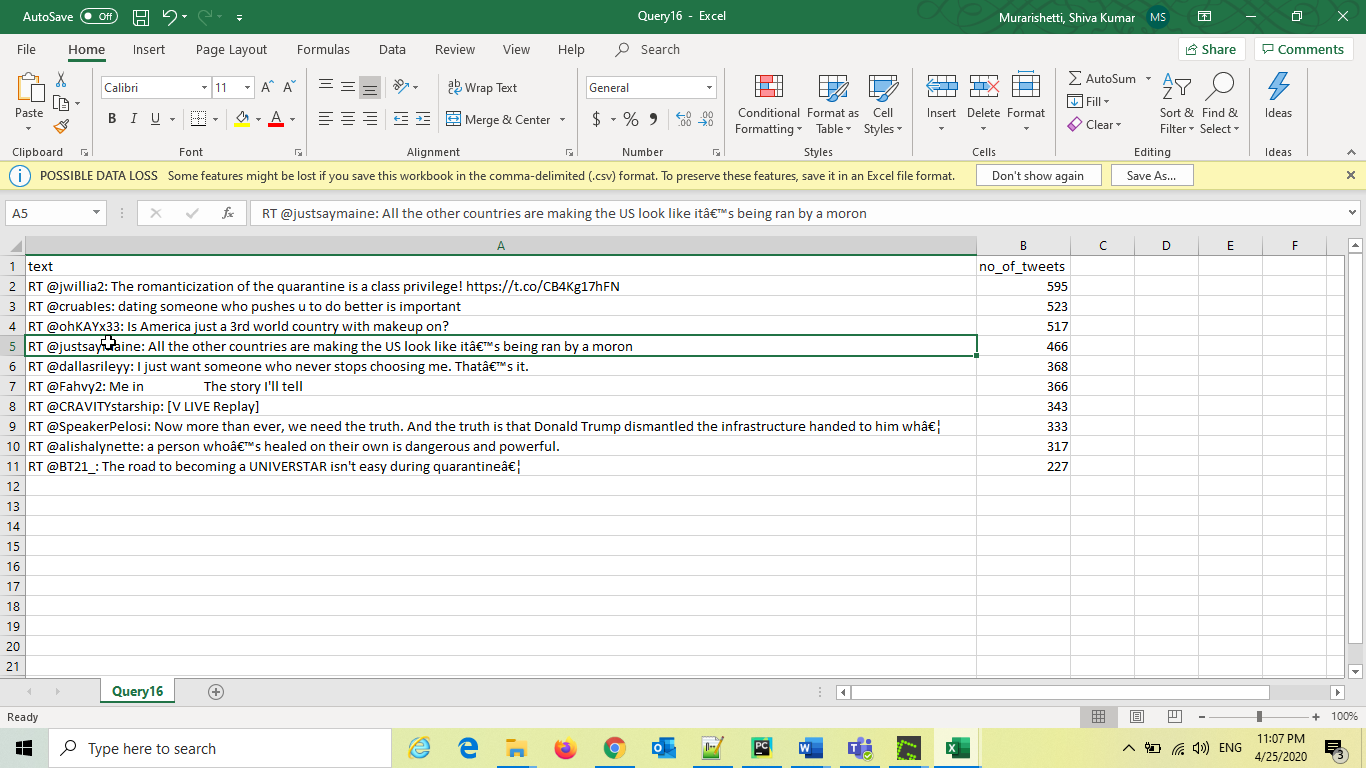
**Query 14 Test**



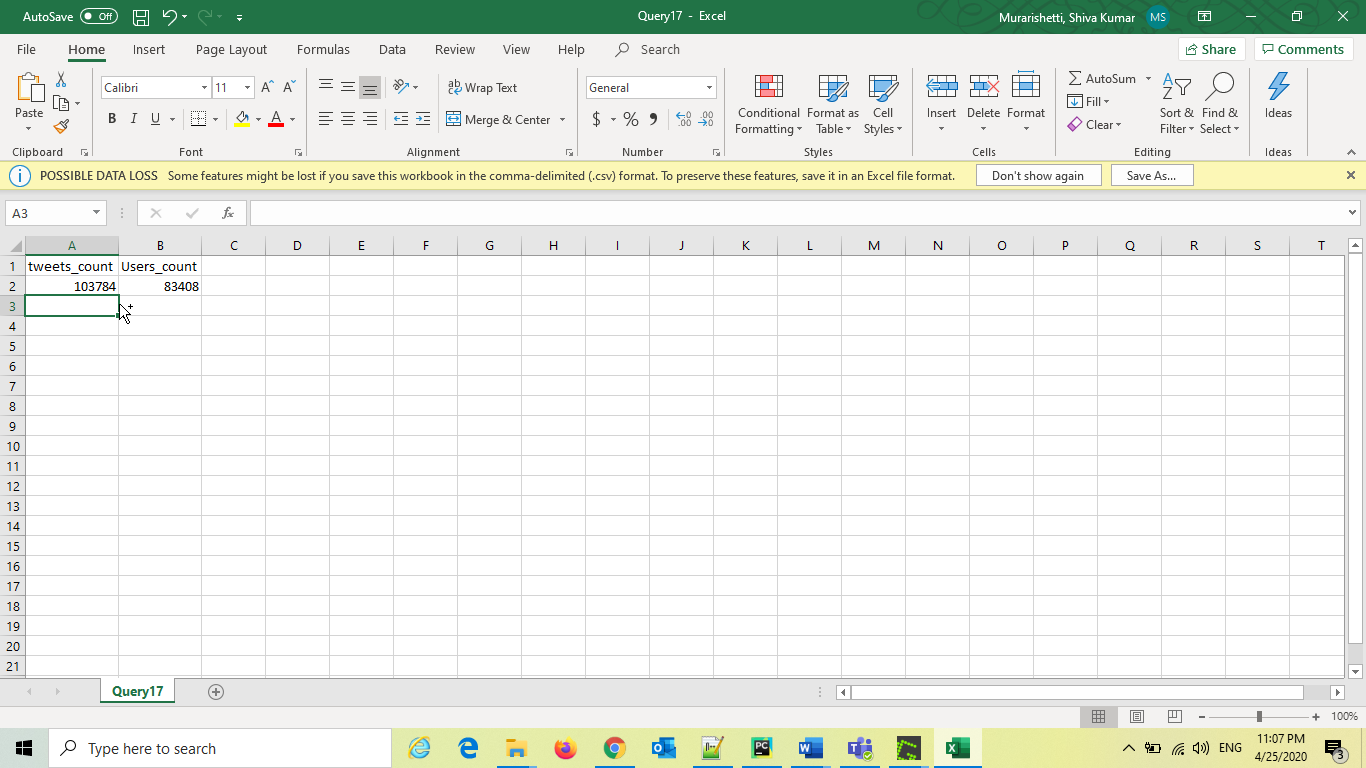
**Query 15 Test**



**Query 16 Test**

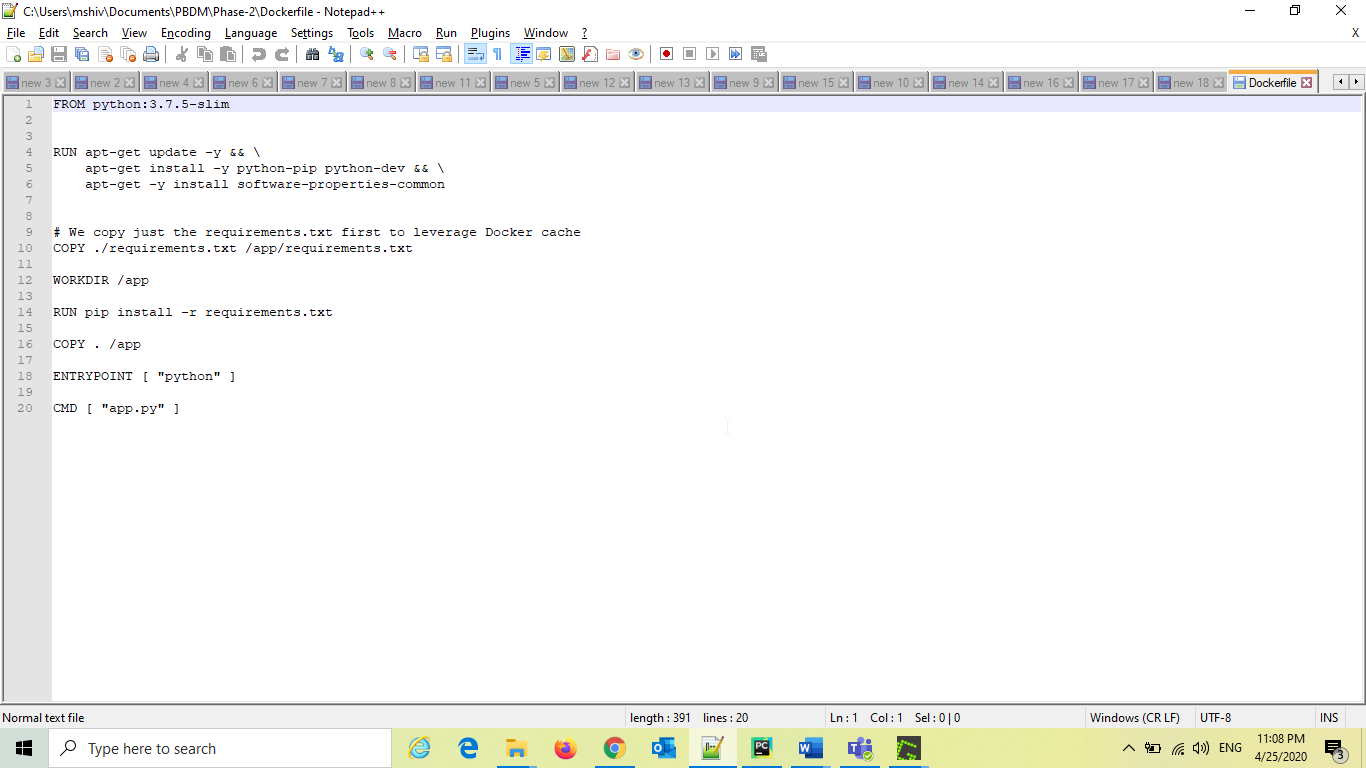


**Query 17 Test**

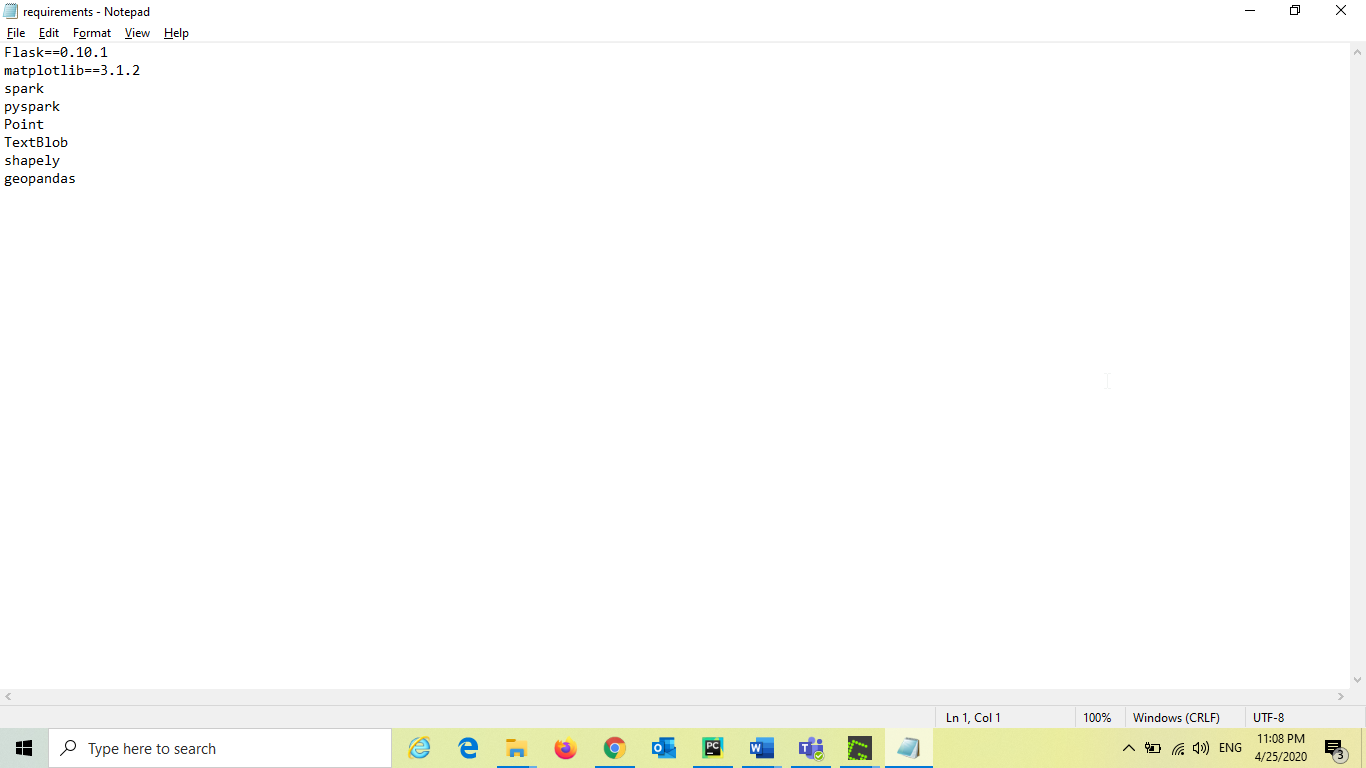


**Docker Implementation**

Dockerfile

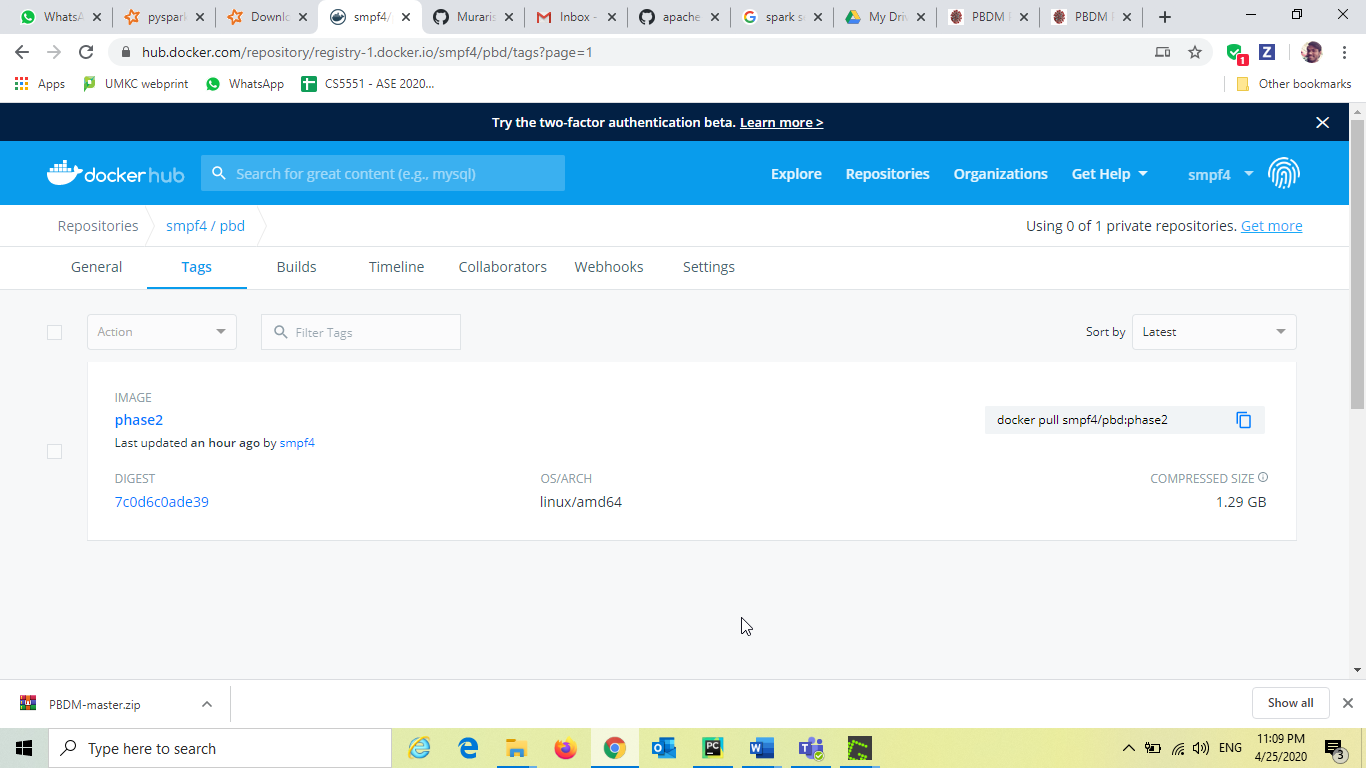


Requirements.txt file



Docker Image link: <https://hub.docker.com/repository/registry-1.docker.io/smpf4/pbd/tags?page=1>

Docker Pull URL: **docker pull smpf4/pbd:phase2**



**References:**

<https://www.jetbrains.com/help/pycharm/creating-web-application-with-flask.html>

<https://spark.apache.org/docs/2.3.0/api/java/org/apache/spark/sql/SparkSession.html>

<https://matplotlib.org/3.2.1/tutorials/introductory/pyplot.html>

<https://geopandas.org/mapping.html>