**Pixel Tweets Analyzer**

Version No.: 1.0

Date: 04/26/2018

Project Name: Pixel Tweets Analyzer

Presented by

Ashok Ramasami – A20441032

Gowrisankar Arumugam – A20400590

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version No. | Date | Prepared by / Modified by | Significant Changes |
| 1.0 | 04/26/2018 | Ashok  Gowrisankar | Initial draft |
| 1.1 | 04/28/2018 | Ashok  Gowrisankar | Added the topics till Tech Giants category |
| 1.2 | 04/29/2018 | Ashok  Gowrisankar | Added topics till My Handle |

Table of Contents

[1. Introduction 5](#_Toc512798509)

[1.1 Cynosure of Pixel Tweets Analyzer 5](#_Toc512798510)

[1.2 Expected outcomes 5](#_Toc512798511)

[2. Technical Details 6](#_Toc512798512)

[2.1 Application Architecture 6](#_Toc512798513)

[2.2 Technology Used 6](#_Toc512798514)

[2.2.1 Front End: 6](#_Toc512798515)

[2.2.2 Back End: 6](#_Toc512798516)

[2.2.3 Python Packages used: 7](#_Toc512798517)

[2.2.4 Domain Knowledge Requirements 7](#_Toc512798518)

[2.2.5 Tools Used: 7](#_Toc512798519)

[3. Data 7](#_Toc512798520)

[3.1 Data cleaning 8](#_Toc512798521)

[3.2 Database 8](#_Toc512798522)

[4. Login 8](#_Toc512798523)

[4.1 Login Screen: 8](#_Toc512798524)

[4.1.1 Signup 8](#_Toc512798525)

[4.1.2 Login 11](#_Toc512798526)

[5. Home 12](#_Toc512798527)

[5.1 Home Screen: 12](#_Toc512798528)

[5.1.1 Py files: 12](#_Toc512798529)

[5.1.2 Database files: 13](#_Toc512798530)

[6. Top Trending 13](#_Toc512798531)

[6.1 Screen: 13](#_Toc512798532)

[6.1.1 Label Description: 16](#_Toc512798533)

[6.1.2 Python Reference: 16](#_Toc512798534)

[6.1.3 Database Tables: 16](#_Toc512798535)

[6.1.4 Exception handling 16](#_Toc512798536)

[6.2 Sentiment Analysis – Pie Chart: 17](#_Toc512798537)

[6.2.1 Python reference 17](#_Toc512798538)

[6.3 Sentiment Reach – Scatter Plot: 18](#_Toc512798539)

[6.3.1 Python reference 18](#_Toc512798540)

[6.4 Source of Tweets – Tree Map: 19](#_Toc512798541)

[6.4.1 Python reference 19](#_Toc512798542)

[6.5 Frequency of Words – Word Cloud: 20](#_Toc512798543)

[6.5.1 Python reference 20](#_Toc512798544)

[6.6 Exception Handling: 21](#_Toc512798545)

[7. Search Tweets 21](#_Toc512798546)

[7.1 Screen: 21](#_Toc512798547)

[7.1.1 Label Description: 24](#_Toc512798548)

[7.1.2 Python Reference: 24](#_Toc512798549)

[7.1.3 Database Tables: 25](#_Toc512798550)

[7.1.4 Exception handling 25](#_Toc512798551)

[7.2 Sentiment Analysis – Pie Chart: 25](#_Toc512798552)

[7.2.1 Python reference 26](#_Toc512798553)

[7.3 Sentiment Reach – Scatter Plot: 26](#_Toc512798554)

[7.3.1 Python reference 27](#_Toc512798555)

[7.4 Source of Tweets – Tree Map: 27](#_Toc512798556)

[7.4.1 Python reference 28](#_Toc512798557)

[7.5 Frequency of Words – Word Cloud: 28](#_Toc512798558)

[7.5.1 Python reference 29](#_Toc512798559)

[8. Tech Gaints 29](#_Toc512798560)

[8.1 Tech Industry Firms 30](#_Toc512798561)

[8.1.1 Details Screen 31](#_Toc512798562)

[8.1.2 Label Description: 32](#_Toc512798563)

[8.1.3 Python Reference: 32](#_Toc512798564)

[8.1.4 Database Tables: 32](#_Toc512798565)

[8.1.5 Comparison plot 33](#_Toc512798566)

[8.1.6 Tweet trend 33](#_Toc512798567)

[8.1.7 Latest Tweet Sentiment 34](#_Toc512798568)

[9. History Screen 34](#_Toc512798569)

[9.1.1 Label Description: 35](#_Toc512798570)

[9.1.2 Python Reference: 35](#_Toc512798571)

[9.1.3 Database Tables: 35](#_Toc512798572)

[10. My twitter Handle 36](#_Toc512798573)

[10.1.1 My Handle Screen 37](#_Toc512798574)

[10.1.2 Python Reference: 38](#_Toc512798575)

[10.1.3 Database Tables: 38](#_Toc512798576)

[10.1.4 Exception handling 38](#_Toc512798577)

[10.1.5 Label Description: 39](#_Toc512798578)

[10.1.6 Tweet post 39](#_Toc512798579)

[10.1.7 Tweet trend 40](#_Toc512798580)

[10.1.8 Latest Tweet Sentiment 41](#_Toc512798581)

# Introduction

**Pixel** is an application which analyses the tweets to get insights about the tweets posted by the users on specific topic. The analysis can be used to identify the crowd sentiment, the source of the tweet, the reach of a sentiment with respect to a specific topic and a word cloud map which shows the most frequently used words. All the above charts can be displayed for both streaming and static data. Apart from the tweet analysis there is an option to check for 5 top tech companies which shows a home page similar to the twitter page, which has options for comparison, trends of the latest 25 tweets and sentiment analysis of the latest tweet posted by the specific user. At last you have an option to manage your twitter home page, do analysis of the replies for the latest tweet and an option to post a tweet.

## Cynosure of Pixel Tweets Analyzer

* Login and registration for users to login to the application.
* MD5 Password authentication is provided for making the password secure.
* Handled maximum exceptional handling.
* 4 visualization charts for both streaming and static data.
* History of the searches to see the trend over a period.
* Page details of users for 5 tech companies
* My Handle page which displays the details of the specific user, data visualization options and an option to tweet from the application.

## Expected outcomes

* Build an application to analyse the crowd response for a topic.
* Data analysis to include both static and streaming data.
* Comapre the basic statistics with other companies’ twitter page.
* Tweet analysis of the home page.
* Entire application to be in MVC format.

# Technical Details

## Application Architecture

Below is the high-level application architecture diagram of the application.



## Technology Used

### Front End:

Python Tkinter : Python package for creating GUI applications.

### Back End:

Azure SQL: Cloud based SQL environment for storing the data and retrieving it for data visulaizations.

### Python Packages used:

Below are the list of libraries that are used to build this project in Python

1. tkinter

2. matplotlib

3. squarify

3. wordcloud

4. pandas

5. numpy

6. pyodbc

7. os

8. sys

9. textblob

10. json

11. pylab

12. scipy

13. warnings

14. hashlib

15. re

16. urllib

17. pil

18. io

19. webbrowser

### Domain Knowledge Requirements

Social Media Networking

### Tools Used:

Python IDLE

Visual Studio 2017

# Data

tweepy is the library provided by python to access the entire twitter RESTful API methods. All the streaming data extracted through ‘tweepy.streaming’ – StreamListener. All the streaming data will wait for the desired number of tweets to be read as and when the user tweets and then stores them in the database. The serch option (Static data) is extracted with ‘api.search’ which will look for tweets which are tweeted till that time and then load it to the database. User details extracted with help of ‘api.get\_user’ and then stored in the database.

## Data cleaning

One of the main challenges in extracting the data is removal of Non-BMP characters from the data. The twitter API returns a JSON file. The JSON file is accessed as a dictionary and the desired values are then stored in the database. A special segment of code is written to replace the non-BMP characters to make them readable.

## Database

One of the main challenges in extracting the data is removal of Non-BMP characters from the data. The twitter API returns a JSON

|  |  |
| --- | --- |
| **Table** | **Description** |
| LOGIN | Table for storing login details |
| TWEET\_MASTER | Table for storing the search history by topic |
| TWEET\_SEARCH | Table for storing the search history by topic and time |
| TWEET\_STREAM | Table for storing the tweet data |
| PAGES | Table for storing the page details of each page |
| PAGES\_TWEET | Table for storing the tweets of each page |
| PAGES\_REPLY | Table for storing the replies for the latest tweet |

**NOTE – For running the application:**

1. Please add your system IP to the SQL server firewall in <https://portal.azure.com> before running the application. (Document and a video has been shared on adding the client IP to the firewall in the same folder named “**Firewall.mp4**” and “**Firewall changes.docx**”)

2. To run the application open the Pixel\_Tweets folder and run the file “**UI.py**”

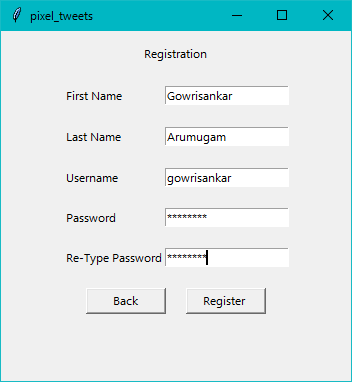
# Login

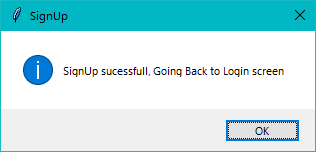
## Login Screen:

Login screen will be displayed when the application is started and will be a common screen for every individual.

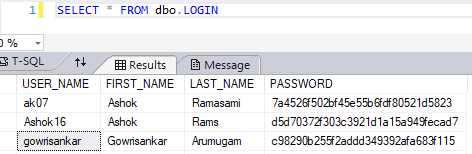
### Signup

When an individual login he/she must enter their User Name and password in the Text box and click on Login. The user will have an option to ‘Signup’ if it’s a new user.





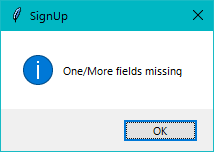
Record has been inserted into the database successfully.



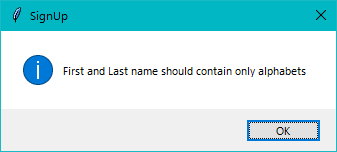
**Signup Screen – Validations:**

Once all the required fields are filled the following validations will be executed to determine whether the entered details are valid or invalid.

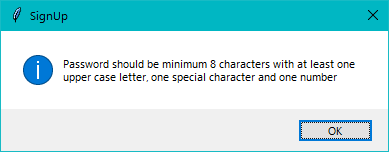
1. Multiple text boxes left blank : “One/More fields missing”



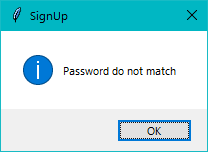
2. First Name and Last name can only take alphabets: “First and Last name should contain only alphabets”



3. Password should be entered as per the password standards : “Password should be minimum 8 characters with at least one upper case letter, one special character and one number”



4. Password and renter password values do not match : “Passwords do not match”



**Py files:**

UI.py

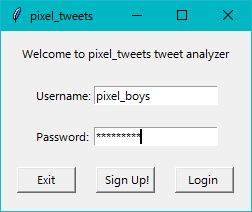
**Database files:**

LOGIN

### Login

Please use the following details to login.

|  |  |  |
| --- | --- | --- |
| **S. No** | **User Name** | **Password** |
| 1 | Pixel\_boys | [Msdhoni@7](mailto:Ananth@1234) |
| 2. | James | Itmd#513 |

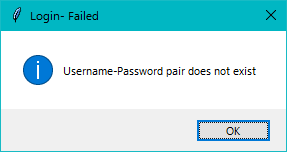


Users who are already registered can login by inputting the Username and the password. The entered password is converted to a Hash value and is validated against the existing password in the database. The password is stored in a Hash value format. If the values match the user is taken to the Home screen, else a pop-up displays prompting the user to enter a valid Username/Password.

**Login Screen – Validations:**

Once all the required fields are filled the following validations will be executed to determine whether the entered details are valid or invalid.

Invalid Username/password : “Username-Password pair does not exist”



**Py files:**

UI.py

**Database files:**

LOGIN

# Home

## Home Screen:

Once logged in the user will be taken to the Home screen. The home screen will have 5 options available.

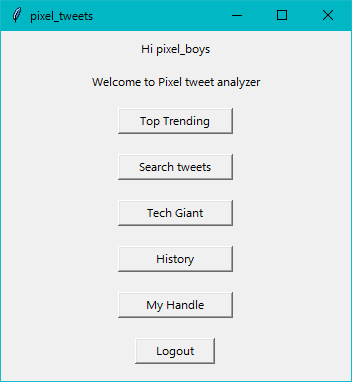
1. Top Trending

2. Search Tweets

3. Tech Giants

4. History

5. My Handle



### Python reference

UI.py

### Database tables

-

# Top Trending

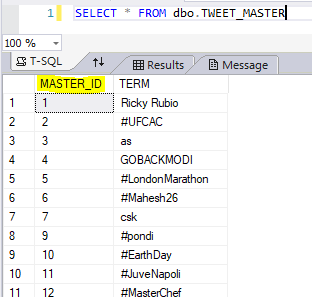
## Screen:

The Top trending screen will have option to enter the number of tweets to analyse, and the user is expected to select a tweet topic from a list box which shows top trending tweets in United States. Once the user enters the values and clicks on analyse, the application will fetch the tweets (no. of tweets entered by the user) which are tweeted by the users after the button has been clicked.

If the topic selected is searched for the first time a new record will be inserted to the table TWEET\_MASTER and TWEET\_SEARCH. TWEET\_MASTER will have a track of all the search topics with the help of MASTER\_ID. TWEET\_SEARCH will have track of all the searches under that specific topic. The MASTER\_ID will remain the same while the SEARCH\_ID will be incremented.

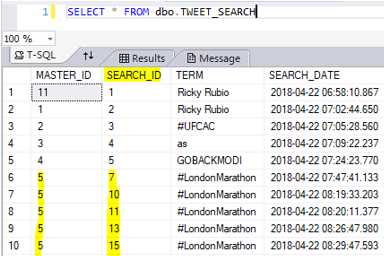
TWEET\_MASTER:

SELECT \* FROM dbo.TWEET\_MASTER



TWEET\_SEARCH:

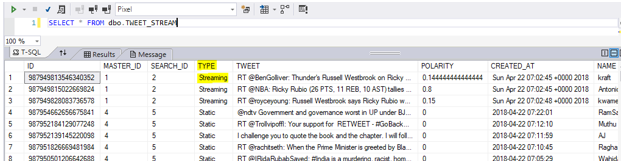
SELECT \* FROM dbo.TWEET\_SEARCH



The data/ tweets will then be loaded to the table TWEET\_STREAM. The table TWEET\_STREAM will have an additional column which states if the data belongs to Streaming search or a Static search. All the required details are then saved to the file.

TWEET\_STREAM:

SELECT \* FROM dbo.TWEET\_STREAM



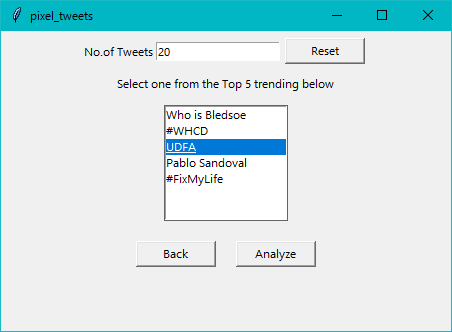
Once the data has been loaded to the database, the following 4 options will be available for analysis.

1. Sentiment analysis – Pie chart

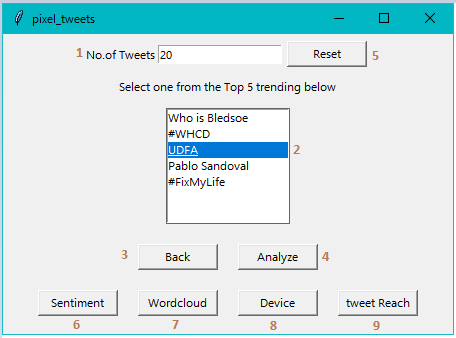
2. Reach of the sentiment – Scatter plot

3. Source of the tweets – Tree Map

4. Frequently used words – Word Cloud



The below 4 buttons generate charts which will be useful for getting a rough idea of the trend of the tweet topic.



### Label Description:

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Name** | **Description** |
| 1 | No. of Tweets | Text Input, enter the number of tweets to be analysed. |
| 2 | Top Trending | Top 5 trending topics in United States – Select one topic to analyse. |
| 3 | Back | Clicking on this button takes the user to the previous screen. |
| 4 | Analyse | Clicking on this button will load the tweets to the database and 4 new buttons (6-9) will appear. |
| 5 | Reset | Clears the N. of Tweets text input box. |
| 6 | Sentiment | Clicking on this button will create a chart which shows the sentiment of the tweet as a pie chart. |
| 7 | Word Cloud | Shows a word cloud with frequently used words. |
| 8 | Device | Shows a tree map with a distribution |
| 9 | Tweet Reach | Takes back to Login page and updates Logout time |

### Python Reference:

|  |  |
| --- | --- |
| **S. No.** | **Python module** |
| 1. | DB\_Operations.py |
| 2. | UI.py |
| 3. | Tweet.py |
| 4. | Sentiment.py |
| 5. | Scatterplot.py |
| 6. | Treemap.py |
| 7. | Wordcloud.py |

### Database Tables:

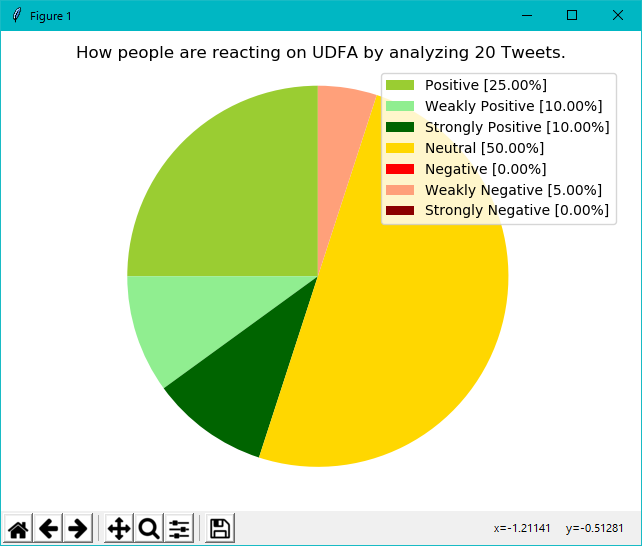
|  |  |  |
| --- | --- | --- |
| **S. No.** | **Table Name** | **Referred For** |
| 1. | TWEET\_MASTER | Keeping track of all the searches topics. |
| 2. | TWEET\_SEARCH | Keeping track of all the searches done and when the search has been made. |
| 3. | TWEET\_STREAM | Store the tweet data into the database for retrieval of the data for plotting. |

### Exception handling

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Reference** | **Error If** |
| 1. | No. of tweets, Trending topics | When no selection has been made or a value is not entered in the No. of tweets field or an invalid value is given in No. of tweets field |

## Sentiment Analysis – Pie Chart:

The pie chart shows overall crowd sentiment for the tweet topic as a pie chart. The data is taken from the Azure SQL database based on the SEARCH\_ID, which is unique to that search done by the user. The cahrt is also saved in the local as a copy for future references.

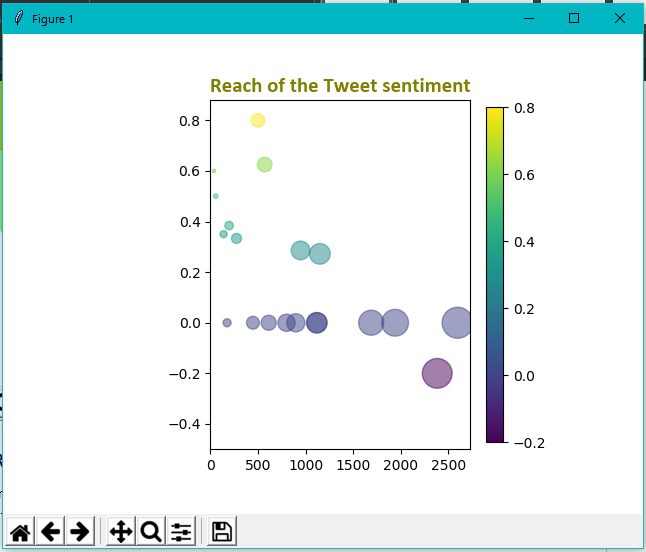


### Python reference

Sentiment.py

## Sentiment Reach – Scatter Plot:

The reach of the sentiment can be found out from the scatter plot. The scatter plot has color which shows the polarity (sentiment) of the tweet with yellow being the positive and indigo being the negative. The size of the circle shows the number of followers the user who tweeted the tweet has, so bigger the circle; it is most likely to reach more users.

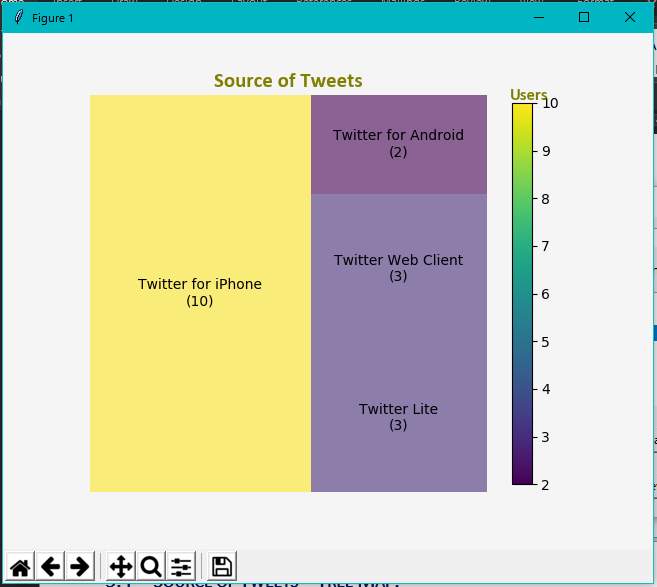


### Python reference

scatterplot.py

## Source of Tweets – Tree Map:

This chart shows the distribution of the source of the tweets, as a tree map. This chart can be used to understand the most commonly used devices the users use for tweeting for that topic. Since this may vary with tweet, business can know the target users for the topic.

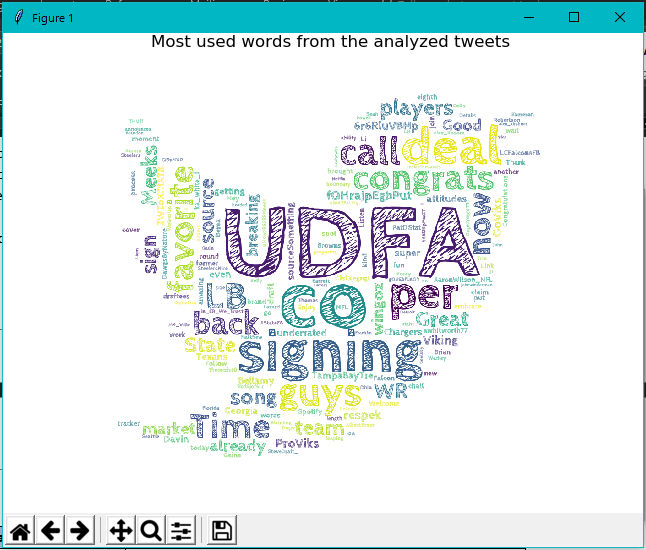


### Python reference

treemap.py

## Frequency of Words – Word Cloud:

The most frequently used words will be shown in the word cloud map. While analysing the trending topics, the word cloud can be used to find out what words have been used frequently. For example, analysing a team tweet, we can find out which player has been mentioned the most. This kind of scenarios will be useful while doing an analysis of what the users are mentioning about while responding about a tweet topic.

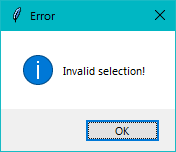


### Python reference

wordcloud.py

## Exception Handling:

1. When no selection has been made or a value is not entered in the No. of tweets field or an invalid value is given in No. of tweets field : “No Selection made!”



# Search Tweets

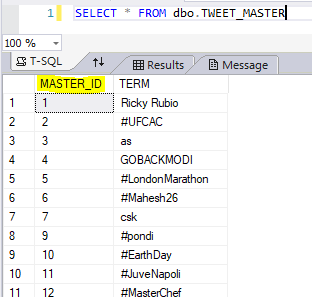
## Screen:

The search tweets screen will have option to enter the term to analyse and the number of tweets to analyse. Once the user enters the values and clicks on analyse, the application will fetch the number of tweets entered by the user.

If the topic entered is searched for the first time a new record will be inserted to the table TWEET\_MASTER and TWEET\_SEARCH. TWEET\_MASTER will have a track of all the search topics with the help of MASTER\_ID. TWEET\_SEARCH will have track of all the searches under that specific topic. The MASTER\_ID will remain the same while the SEARCH\_ID will be incremented.

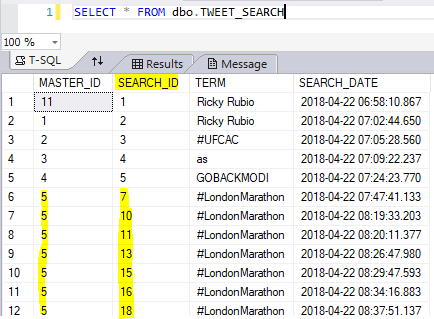
TWEET\_MASTER:

SELECT \* FROM dbo.TWEET\_MASTER



TWEET\_SEARCH:

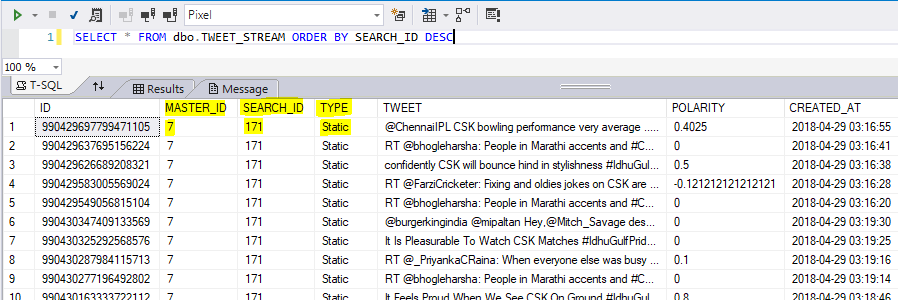
SELECT \* FROM dbo.TWEET\_SEARCH



The data/ tweets will then be loaded to the table TWEET\_STREAM. The table TWEET\_STREAM will have an additional column which states if the data belongs to Streaming search or a Static search. All the required details are then saved to the file.

TWEET\_STREAM:

SELECT \* FROM dbo.TWEET\_STREAM



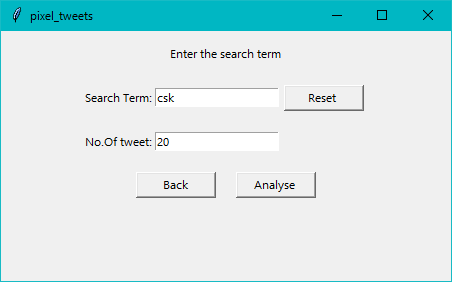
Once the data has been loaded to the database, the following 4 options will be available for analysis.

1. Sentiment analysis – Pie chart

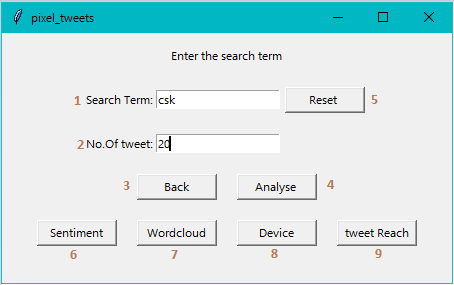
2. Reach of the sentiment – Scatter plot

3. Source of the tweets – Tree Map

4. Frequently used words – Word Cloud



The below 4 buttons generate charts which will be useful for getting a rough idea of the trend of the tweet topic.



### Label Description:

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Name** | **Description** |
| 1 | Search Term | Text Input, enter the topic to be analysed. |
| 2 | No. of tweets | Text Input, enter the number of tweets to be analysed. |
| 3 | Back | Clicking on this button takes the user to the previous screen. |
| 4 | Analyse | Clicking on this button will load the tweets to the database and 4 new buttons (6-9) will appear. |
| 5 | Reset | Clears the N. of Tweets text input box. |
| 6 | Sentiment | Clicking on this button will create a chart which shows the sentiment of the tweet as a pie chart. |
| 7 | Word Cloud | Shows a word cloud with frequently used words. |
| 8 | Device | Shows a tree map with a distribution |
| 9 | Tweet Reach | Takes back to Login page and updates Logout time |

### Python Reference:

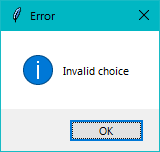
|  |  |
| --- | --- |
| **S. No.** | **Python module** |
| 1. | DB\_Operations.py |
| 2. | UI.py |
| 3. | search.py |
| 4. | Sentiment.py |
| 5. | Scatterplot.py |
| 6. | Treemap.py |
| 7. | Wordcloud.py |

### Database Tables:

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Table Name** | **Referred For** |
| 1. | TWEET\_MASTER | Keeping track of all the searches topics. |
| 2. | TWEET\_SEARCH | Keeping track of all the searches done and when the search has been made. |
| 3. | TWEET\_STREAM | Store the tweet data into the database for retrieval of the data for plotting. |

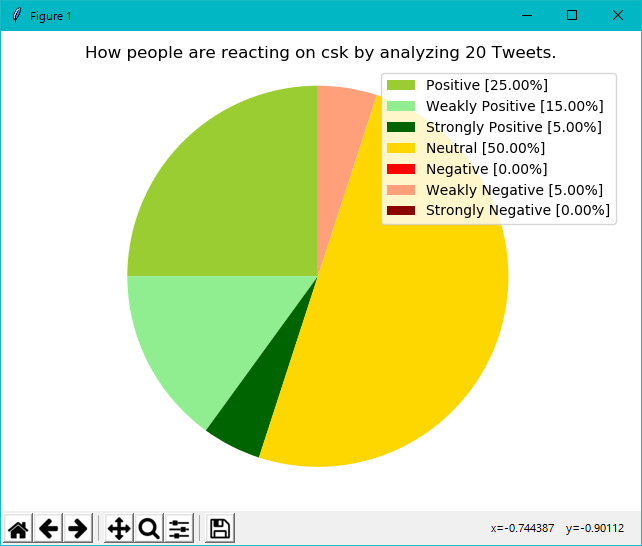
### Exception handling

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Reference** | **Error If** |
| 1. | Search Term, No. of tweets | When no value is entered in the No. of tweets/Search Term field or an invalid value is given in search term/No. of tweets field |



## Sentiment Analysis – Pie Chart:

The pie chart shows overall crowd sentiment for the tweet topic as a pie chart. The data is taken from the Azure SQL database based on the SEARCH\_ID, which is unique to that search done by the user. The cahrt is also saved in the local as a copy for future references.

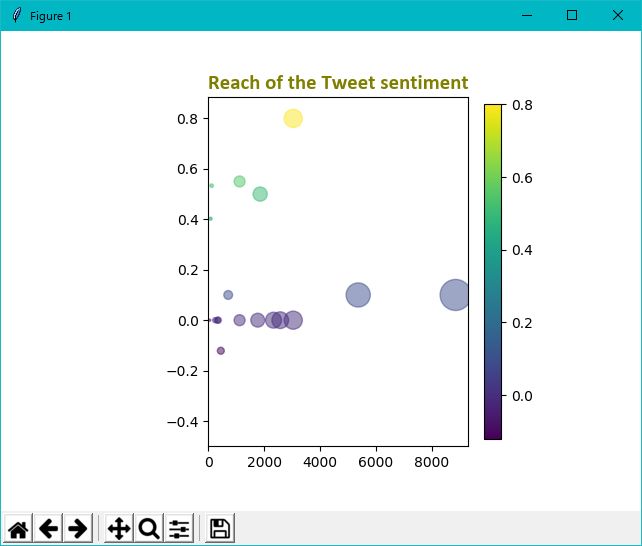


### Python reference

Sentiment.py

## Sentiment Reach – Scatter Plot:

The reach of the sentiment can be found out from the scatter plot. The scatter plot has color which shows the polarity (sentiment) of the tweet with yellow being the positive and indigo being the negative. The size of the circle shows the number of followers the user who tweeted the tweet has, so bigger the circle; it is most likely to reach more users.

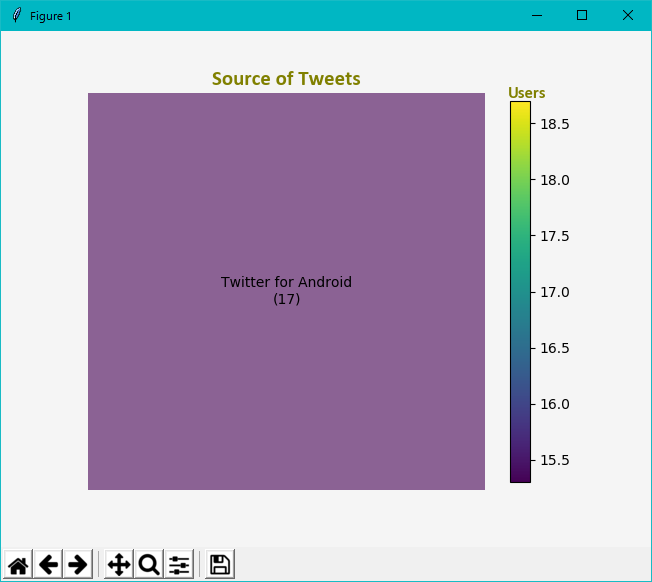


### Python reference

scatterplot.py

## Source of Tweets – Tree Map:

This chart shows the distribution of the source of the tweets, as a tree map. This chart can be used to understand the most commonly used devices the users use for tweeting for that topic. Since this may vary with tweet, business can know the target users for the topic.

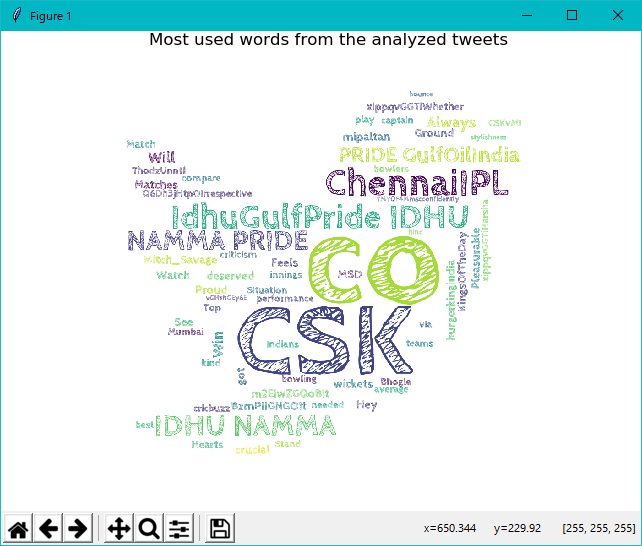


### Python reference

treemap.py

## Frequency of Words – Word Cloud:

The most frequently used words will be shown in the word cloud map. While analysing the trending topics, the word cloud can be used to find out what words have been used frequently. For example, analysing a team tweet, we can find out which player has been mentioned the most. This kind of scenarios will be useful while doing an analysis of what the users are mentioning about while responding about a tweet topic.



### Python reference

wordcloud.py

# Tech Gaints

This part of the application is created for comparison between the tech companies and displays the details about the firm chosen.

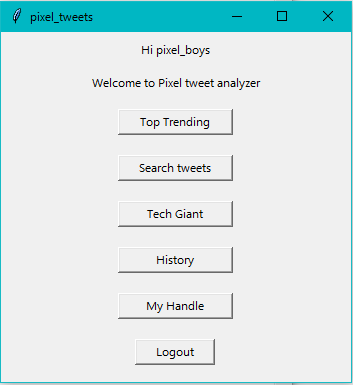
The main objective is to find out how the tweets posted by the firm is perceived globally and its impact on the products or the services they offer. Now a day’s social media serves a platform through which information is spread around the world also it serves as a mass media for marketing the products or the services that company offers, plans to offer. The more reach of the tweet about a product the more sales occurs for the product.

This part of the application has the tech companies (Microsoft, Google, IBM and Twitter) and IIT-Chicago. The operations the user could perform are plot a comparison chart between the companies based on the details like followers, tweets posted and likes. The other operation is to analyse the latest 25 tweets posted by the firm in twitter and then plot the number of likes for those tweets along with the retweet count. The last option available is to analyse the sentiment for all the replies posted worldwide for the latest tweet posted by the firm.

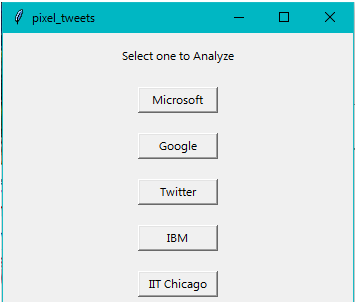
## Tech Industry Firms

Once the user clicks on the button “Tech Giant” there occurs background process where the connection to the twitter API is made and the details about the companies Microsoft, Google, IBM, Twitter, IIT-Chicago like followers count , official twitter handle name, likes , tweets posted, retweets, twitter profile picture, web link are loaded into the database. For every instance the button is clicked the table “PAGES” in the database is truncated and then the data is loaded. This is done to get the accurate details about the firm since the details can change over time. Also the home screen is provided with a logout button which when clicked asks for a confirmation from the user if he /she need to logout from the application if ‘OK’ is pressed then the user is navigated to ‘Login’ screen else the application stays in the same screen.

In the front end of the application it redirects to the next screen with a button for each firm.



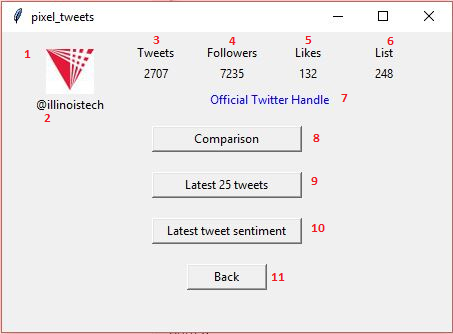
Home Screen- IM1



Tech Giant-IM2

### Details Screen

This screen displays the details about the firm selected by the user with 3 other buttons for plotting the sentiment, tweet reach, comparison plot along with a back button to navigate to the previous screen. Also there is a hyperlink which on clicking opens the official twitter page in the default web browser.



6.2.1 Details Screen-IM1

### Label Description:

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Name** | **Description** |
| 1 | Twitter label | Twitter profile pic of the selected firm |
| 2 | Twitter Username | Twitter username of the firm |
| 3 | Tweets | The number of tweets posted by the firm in Twitter |
| 4 | Followers | The number of followers the firm has |
| 5 | Likes | Number of overall likes for the tweets posted by the firm |
| 6 | List | Popularity of the firm in twitter |
| 7 | Hyperlink | Hyperlink to firms official twitter page |
| 8 | Comparison button | Button to plot the comparison chart |
| 9 | Tweet Trend | Button to plot the likes and retweets for last 25 tweets posted by the firm |
| 10 | Latest Tweet | Button to plot the sentiment chart for the latest tweet posted by the firm |
| 11 | Back | Button to navigate to the previous screen |

### Python Reference:

|  |  |
| --- | --- |
| **S. No.** | **Python module** |
| 1. | DB\_Operations.py |
| 2. | comparison.py |
| 3. | image.py |
| 4. | in\_sentiment.py |
| 5. | tgtrend.py |
| 6. | users.py |
| 7. | page\_tweets.py |

### Database Tables:

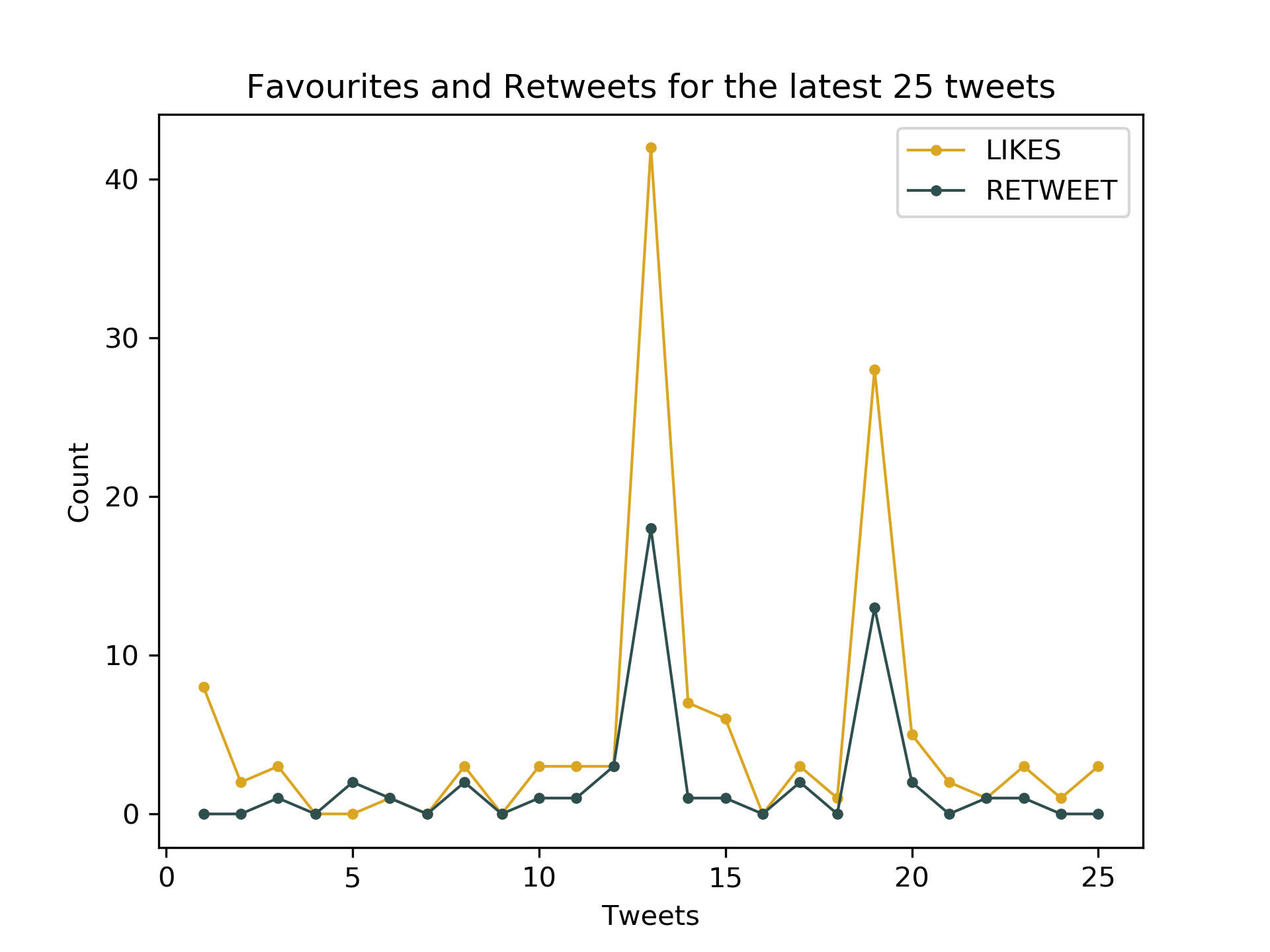
|  |  |  |
| --- | --- | --- |
| **S. No.** | **Table Name** | **Referred For** |
| 1. | PAGES | Getting details of user pages. |
| 2. | PAGES\_TWEET | Tweet data to plot the trend chart to analyse the latest 25 tweets |
| 3. | PAGES\_REPLY | Tweet data to analyse the sentiment of the replies for the last tweet posted. |

### Comparison plot

This plot is all about plotting the followers, likes and retweets of the firm selected by the user versus the other list of companies. Since the followers for some firms are in millions the values are scaled and then plotted. It also has a table below the chart which displays the exact details about the firms by retrieving it from the database table “PAGES”. This part of the application is linked with the button “Comparison”.

### Tweet trend

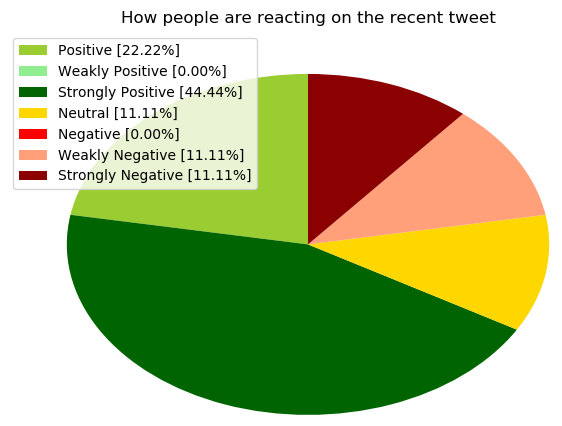
For this plot the latest 25 tweets posted by the firm are taken and based on their unique id the details like retweets and the number of times the tweet has been liked by other twitter users is taken with the help of the API and then stored in the table “PAGES\_TWEET”. Now the data from the table is used to plot the trend chart. This part of the application is linked with the button “Latest 25 Tweet”.



6.2.3 Trend Plot - IM1

### Latest Tweet Sentiment

For this plot the replies for the latest tweet posted by the firm is retrieved through the API and is stored into the table “PAGES\_REPLY” in the database. Then the data is read one by one to get calculate its polarity value and finally the sentiment plot is plotted. This functionality is linked to the button “Latest tweet sentiment”

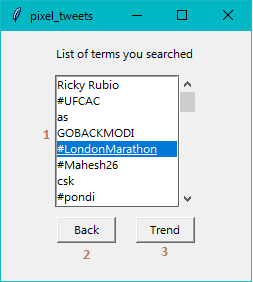


6.2.4 Sentiment plot – IM1

# History Screen

This screen provides the polarity trend for the term searched in twitter. For example, if the user searched for a single term multiple times with some time gap between each search then the polarity for the search term at each instance of the search could vary. This screen lists all the terms that have been analysed through both the streaming and search API in a list box.

The trend can be used to get an estimate of how the users have reacted for a topic over a period of time and how the sentiment has either moved towards positive or negative. When corelated with the latest happenings, it helps us understand what impact it has made on users.



### Label Description:

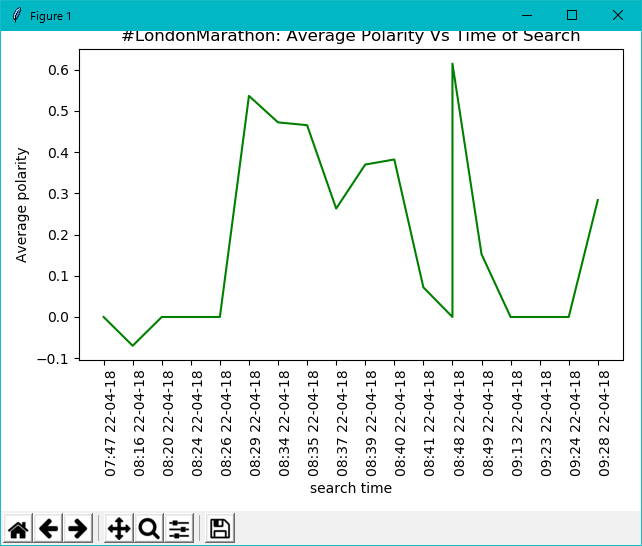
|  |  |  |
| --- | --- | --- |
| **S. No.** | **Name** | **Description** |
| 1 | List of tweets searched | List of all the searches made till now. |
| 2 | Back | Button to navigate to the previous screen |
| 3 | Trend | Analyse the trend over the period of time. |

### Python Reference:

|  |  |
| --- | --- |
| **S. No.** | **Python module** |
| 1. | DB\_Operations.py |
| 2. | Trend.py |

### Database Tables:

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Table Name** | **Referred For** |
| 1. | TWEET\_SEARCH | Getting details of all the searches made. |
| 2. | TWEET\_STREAM | Get the sentiment polarity for the tweet data present in the database. |
| 3. | TWEET\_MASTER | Get the list of searches the user has made till now. |

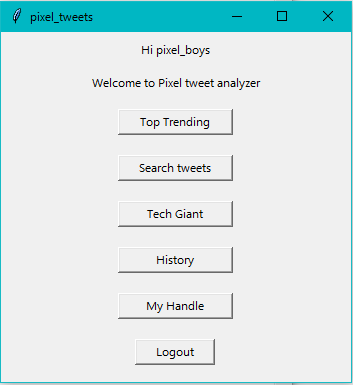


# My twitter Handle

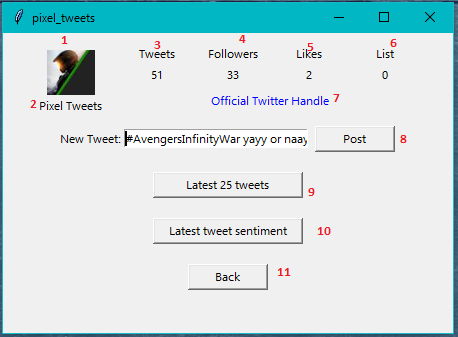
Like the “Tech Giants” screen this part also displays the details like followers, tweets posted, likes twitter name, twitter profile picture except that it doesn’t have the option to compare with other tech firms as it deals with the personal twitter handle. It provides options to view the chart for the last 25 tweets likes and retweets, and plot the sentiment for the replies of the last tweet posted and a hyperlink the twitter webpage. In addition to that there is also an option to post a tweet into the twitter account directly from the application through the API. The major reason for this part of the application to be built is to provide the user a medium not only to analyse the tweets about other topics and view the details about top firms but also to view and get to know the updates of his twitter handle as well.

### My Handle Screen

Once the button “My Handle” button is pressed the details about the user is retrieved through twitter API and stored in database table “PAGES”. This data is then pulled and displayed in a tkinter window.



7.1.1 Home Screen –IM1



7.1.1 Details Screen-IM2

### Python Reference:

|  |  |
| --- | --- |
| **S. No.** | **Python module** |
| 1. | DB\_Operations.py |
| 2. | Posttweet.py |
| 3. | image.py |
| 4. | in\_sentiment.py |
| 5. | tgtrend.py |
| 6. | users.py |

### Database Tables:

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Table Name** | **Referred For** |
| 1. | PAGES | Getting details of user pages. |
| 2. | PAGES\_TWEET | Tweet data to plot the trend chart to analyse the latest 25 tweets |
| 3. | PAGES\_REPLY | Tweet data to analyse the sentiment of the replies for the last tweet posted. |

### Exception handling

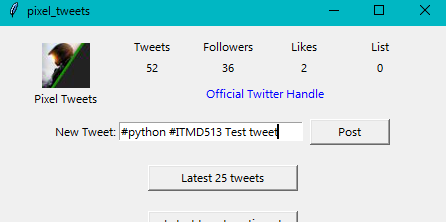
|  |  |  |
| --- | --- | --- |
| **S. No.** | **Reference** | **Error If** |
| 1. | New Tweet | Empty |
| 2. | New Tweet | Post length greater than 140 characters |

### Label Description:

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Name** | **Description** |
| 1 | Twitter label | Twitter profile pic of the selected firm |
| 2 | Twitter Username | Twitter username of the firm |
| 3 | Tweets | The number of tweets posted by the firm in Twitter |
| 4 | Followers | The number of followers the firm has |
| 5 | Likes | Number of overall likes for the tweets posted by the firm |
| 6 | List | Popularity of the firm in twitter |
| 7 | Hyperlink | Hyperlink to firms official twitter page |
| 8 | Post | Button to post the contents typed in the entry field into the user’s twitter handle |
| 9 | Tweet Trend | Button to plot the likes and retweets for last 25 tweets posted by the firm |
| 10 | Latest Tweet | Button to plot the sentiment chart for the latest tweet posted by the firm |
| 11 | Back | Button to navigate to the previous screen |

### Tweet post

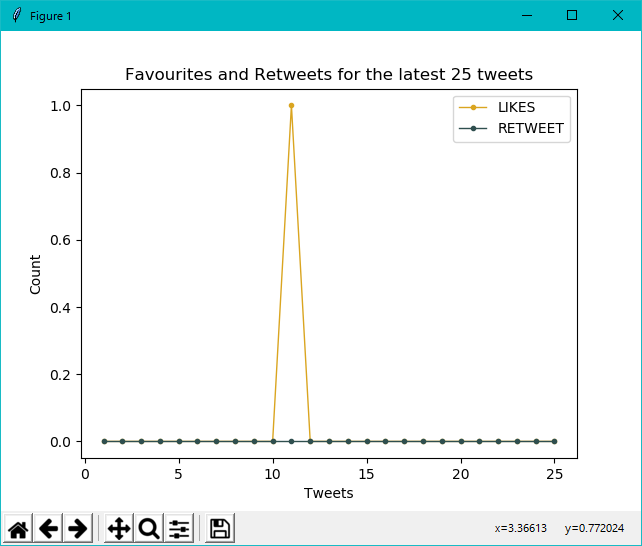
This is an entry field which allows the user to post a tweet into his/her twitter account directly from this application. Same as the post length of 140 characters in twitter we have also restricted the length of the post to 140 characters. Once the post button is clicked then the content in the entry field is posted in twitter.





### Tweet trend

For this plot the latest 25 tweets posted by the user are taken and based on their unique id the details like retweets and the number of times the tweet has been liked by other twitter users is taken with the help of the API and then stored in the table “PAGES\_TWEET”. Now the data from the table is used to plot the trend chart. This part of the application is linked with the button “Latest 25 Tweet”.



### Latest Tweet Sentiment

For this plot the replies for the tweet posted by the firm is retrieved through the API and is stored into the table “PAGES\_REPLY” in the database. Then the data is read one by one to get calculate its polarity value and finally the sentiment plot is plotted. This functionality is linked to the button “Latest tweet sentiment”.

