A PROJECT REPORT
ON

"TWITTER DATA ANALYSIS"

Group 4A

Submitted to

International Institute of Information Technology Bangalore

In Partial Fulfilment
of the Requirement for the
Award of

Master of Technology

BY

Aman Jain	**MT2017014**
Amar Prakash Mishra	**MT2017015**

UNDER THE GUIDANCE OF
PROF. Shrisha Rao

Acknowledgements

We are profoundly grateful to **Prof. Shirsha Rao** for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion.

Without his active co-operation and guidance, it would have become very difficult to complete task in time. We are highly indebted to him for his invaluable guidance and ever ready support in successfully completing this project in stipulated time. His persisting encouragement and perpetual motivation, everlasting patience and excellent expertise in discussions during the progress of the project work has benefited us a lot, which is beyond expressions.

At last we must express our sincere heartfelt gratitude to all the staff members of Computer Engineering Department who helped me directly or indirectly during this course of work.

AMAN JAIN MT2017014
AMAR PRAKASH MISHRA MT2017015

Contents

Chapter 1

Introduction

Goal of this project is to track what users are saying about your product, organization, political party by collecting all the tweets related to your Keyword. This keyword can be anything Product, name of the organization, name of the person etc. What are the opinion of the people about your product, How much they are liking your product, what are the topics that are on trend related to your product. what are the Hashtags people are using for your product

If you are getting some response then from which Geographical Location you are getting most of the response then they can do more advertisement from where they are getting most response. Not only this you can compare your company/organization/political party with the opposite company/party to see how opposite team is growing for the same product what is the satisfaction of the user for the other party .

Chapter 2

Algorithms

2.1 Fetch the tweets

1. It will get the user id defined by us and token key and token secret.
2. It will fetch the keywords related to a user and store them in the array.
3. Then it will take the consumer key and consumer secret provided by the APP and verify the application.
4. It will fetch and filter the tweets using FilterQuery Class and filter() method provided by tweeter 4j API.
5. At every tweet it will call Listener event which will push the tweets in the RabbitMQ.
6. Once all tweets pushed and they further will be stored in the HashMap name as streamMap.
7. Now if a user will add/delete some entity so now it needs to stop the current stream known as shutdown but before doing shutdown we need to check whether its stream is currently going on or not that we can check via streamMap by passing userid.
8. If it is null that means there was no stream earlier so we can start new stream for fetching the tweets without shutdown.
9. otherwise we need to shutdown the earlier stream first then only we can start the new one.

2.2 People Talking

For every tweet do the following below steps:

1. Find all the entities available in the tweet by comparing with the entities already available along with the userID.
2. now it will find the Time at which this tweet has been posted.
3. convert this time in the hours.
4. then it will check in the table.
5. if there is already one entry in the table with same entity and same time then it will simply increment the count in the same row.
6. otherwise it will create one more new entry and initialize count as 1.

2.3 Geographic Location

For every tweet do the following steps:

1. Find all the entities available in the tweet by comparing with the entities already available along with the userID.
2. First it will check whether the tweet has some location or not because There is chance that sometimes location is not provided in the tweet eg. GPS off (Global Positioning System).
3. otherwise do the steps from 3.1 to 3.3
    a. now it will find the Time at which this tweet has been posted.
    b. get the Longitude and Latitude using the standard method provide in the Twitter 4j API.
    c. Populate the data in the table.

2.4 Number of hashtags

For every tweet do the following steps

1. Find all the entities available in the tweet by comparing with the entities already available along with the userID.
2. Check if the tweet is reTweeted then we dont need to do anything because that tweet has been already processed.

3. Otherwise find out all the hashtag entities by getHashtagEntities() method and store into one array.
4. Convert time into hours.
5. For all hashtagEntities do the following steps
    a. Convert into text.
    b. For all entities which are present in tweet do the following steps.
        i. Search onto the table if there is no entry present for that particular entity at that particular time, then intialize count by 1
        ii. Otherwise increment the count by 1.
        iii. Again store onto the hashCounter table

2.5 Most Retweeted tweet

For every tweet do the following steps

1. Find all the entities available in the tweet by comparing with the entities already available along with the userID.
2. If Tweet does not have any retweeted tweet inside of it then we dont need to do anything
3. Otherwise find out the status of retweeted tweet Find out the count of retweeted tweet if it is less than MIN RETWEETS FOR TRENDING then we will simply return
4. We can set MIN RETWEETS FOR TRENDING by ourself
5. We have fixed MIN RETWEETS FOR TRENDING as 10
6. Otherwise find out list of all entries for retweeted tweet id if (size of list is zero).
    a. Set the retweet count of all entities of tweets
    b. And also add to the list so that later we will be able to store into table
    c. otherwise
    a. Set retweet count for all entries of list
    b. Finally store the list into the table

2.6 Relevant topics

Means , what are all the topics about which people are talking. For example if the entity is iphone then the topics can be Camera , sound , speaker , screen , flash light etc.
There are some words which can not be a topic generally for any entity those all words are known as stop words like i, me, you, is, am, and, but, therefore etc.

1. Find all the entities available in the tweet by comparing with the entities already available along with the userID.
2. Check if tweet is not retweeted then do following :
    i. Tokenize the entire text into tokens and find out the time
    ii. for each token
        a. check if token is starting with alphabet and does not contain any stop word then consider that token as a topic.
        b. Then in the topics for each entities check if that topic has been stored earlier then increment the count by 1, otherwise sipmly store that topic and initialize count as 1

2.7 Sentiment Analysis

The process of identifying and categorizing opinions expressed
in a piece of text (tweets) Here we have taken the batch size = 10
We will get the score from 0 to 4 from sentiment analyzer which is provided by stanford NLP library

id	time	entity_id	Score	count
1	10:00AM	1	1	2
2	10:00AM	4	1	2
3	10:00AM	6	1	2
4	10:00AM	2	1	1
5	11:00AM	2	2	1
6	10:00AM	1	3	4

Calculation of Average Score :

Example - Suppose for entity_id = 1 , we want to know the score At 10:00 AM we have two rows for entity_id = 1

$$AvgScore = Sum\_of(score * count)/totalcount$$

$$Avgscore = ((1*2)+(3*4))/(2+4)$$

$$Avgscore = 14/6$$

$$Avgscore = 2.33$$

(Approximately Neutral)So , at 10:00 AM for entity_id = 1 , We will show opinion as Neutral

Chapter 3

Implementation of Database

TABLE UserLogin
Whenever the user will login through our website into twitter , you will get token key and token secret by Oauth protocol and those token key and token secret will be stored onto our following table . indicates that there can be any number of users .

user_id	access_token	access_token_secret

TABLE Entity
User is allowed on our website to create various entities . For example user might want to know review about apple company . So here entity_name is apple . Same user can also know about Google . In this table entity_id will be the primary key .

entity_id	entity_name	user_id

TABLE Followers
One entity can have any number of handles . For example if entity is apple then handle can be iphone-7 , iphone-8 . This table is used mainly for finding out the number of follower for particular entity .

id	handle	entity_id

TABLE Keyword
One entity can have any number of keywords . Based on the keywords provided by the user it will search for a particular entity .

id	keyword	entity_id

TABLE Tweet
Whenever you will get the tweet you will store the tweet into this tweet table .
In this table , there are some attributes which are -
id - id which will be allocated by you .

tweet_id id which twitter has given .
entity_id tweet is related to which entity .
Text what is the text present inside the tweet .
retweet_count - How many times this tweet has been retweeted.
created_at what is the time at which this tweet has been created .
screen_name Who has created this tweet .

id	tweet_id	entity_id	text	retweet_count	created_count	screen_name

TABLE Sentiment
After taking the tweet from the rabbitmq one by one , find out sentiment score by using stanford NLP or some other library and store onto the with time and entity_id . If for the same time and for the same entity_id you are getting the same score then simply increment the count attribute by 1 instead of creating a new row .

id	entity_id	time	score	count

TABLE Topic
In this table , you will store all the topics which have been talked for the particular entity . Created_at attribute is saying the time at which it has been created . Topic attribute indicates the topic name. For example , if entity_name is mobile then the topics can be camera or flash light or speaker etc. If for the same entity and for the same time-interval the particular topic comes more than one time then increment count by one .

id	entity_id	topic	created_at	count

TABLE PeopleTalking
In this table you are storing all persons which are tweeting for particular entity at the same time . mention_time indicates the time at which person is tweeting . Count will tell the number of persons for a same time and same entity .

id	entity_id	mention_time	count

TABLE Hashtag
In this table , hash_tag attribute indicates the name of the hash_tag . Created_at attribute indicates the time at which it has been posted and increment the count attribute if for the same time same hash_tag is coming more than one time .

id	entity_id	hash_tag	created_at	count

TABLE Handle

In this table , screen_name attribute indicates handle name and time attribute stores the time . You increment the count attribute if at the same time , same handle is coming more than one time .

id	screen_name	time	count

TABLE Location

In this table , you will store entity_id for which this tweet is related and at which time this tweet has been created (in created_at attribute) and store the latitude and longitude of location from where this tweet has been posted .

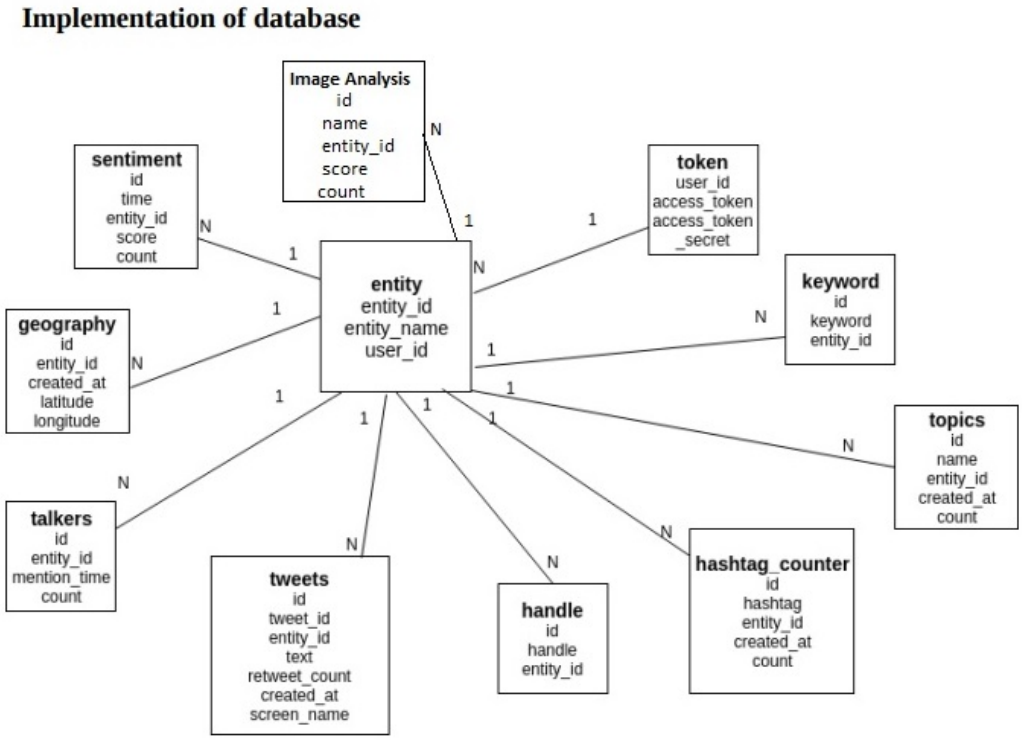
id	entity_id	created_at	latitude	longitude

TABLE ImageRecognition

In this table , you are storing the name of image after recognitin and at time at which it has been posted and if for the same time , the same image name is coming more than 1 then increment count by 1 every time .

id	entity_id	name	created_at	count

In this project totally we need 12 tables . class diagram for those tables is given below .



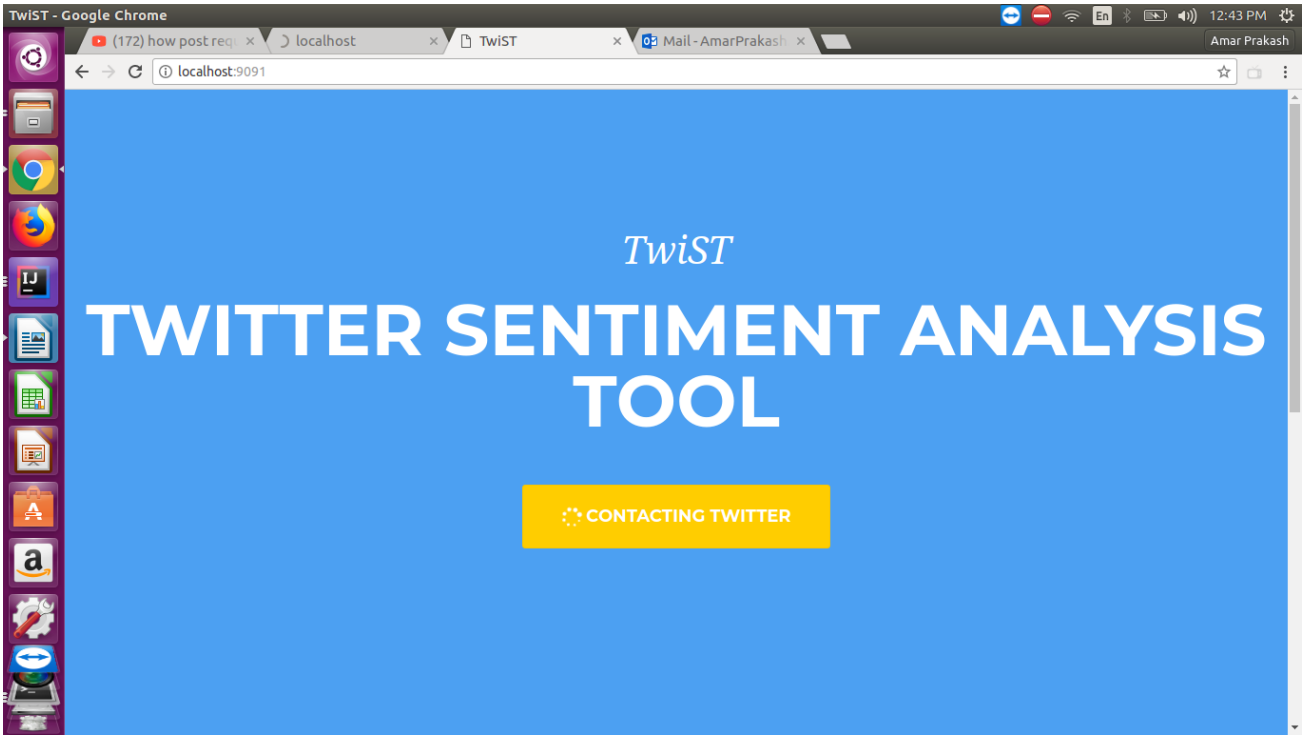
Chapter 4

Work Flow Of Project

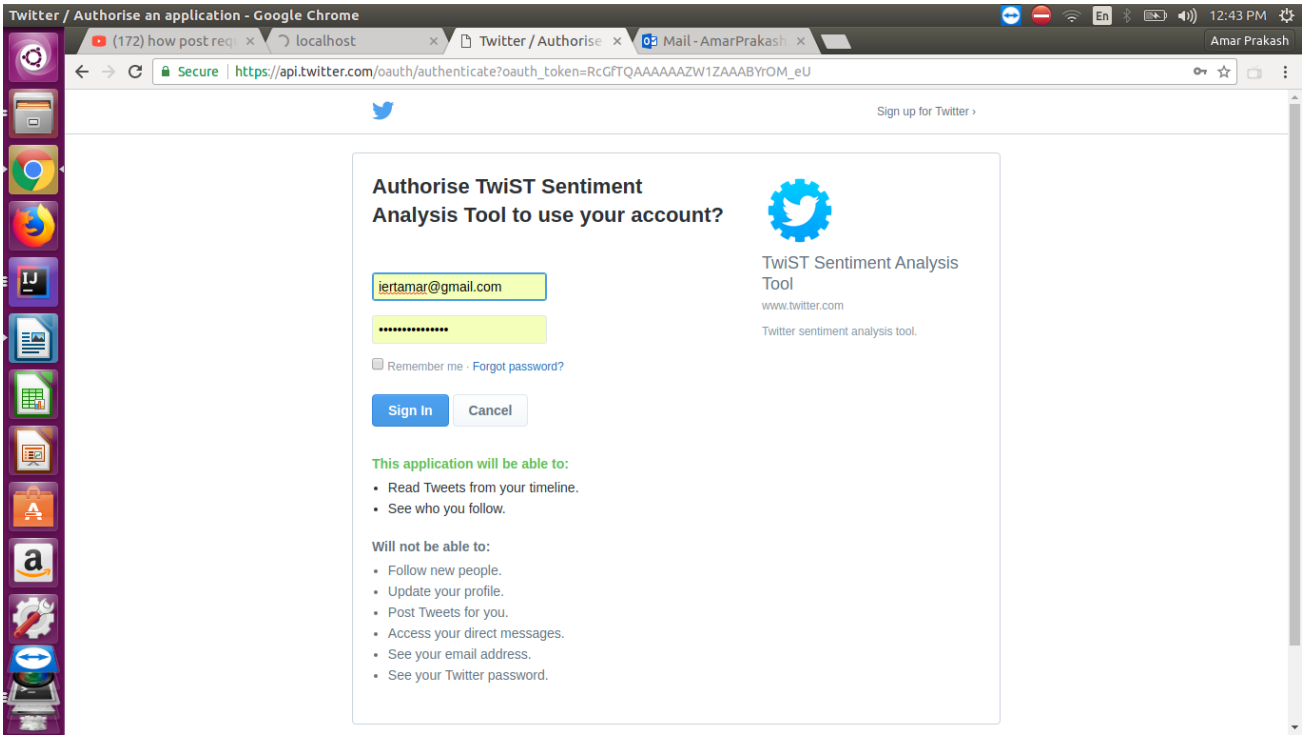
4.1 Architecture

There Are N number of users, these users will come to our web API and contact to the twitter where twitter will verify the web API using consumer key and consumer token and then twitter will provide their own interface to the user where user will enter their username and password for which twitter will authenticate the user and it will generate the access key and access token then on our web API it will return the user to the datafetch API where Datafetch will provide all the entity through database and display it through the web API. It will go to the twitter Stream API where it will send the stream of the tweet and these stream of the tweet will be sending to the RabbitMQ then from this RabbitMQ,tweets will be fetch one by one and send them to the analysis module once they generate the result then send their result to the database and again datafetch will fetch it from the database and display it to the user based on requirement.

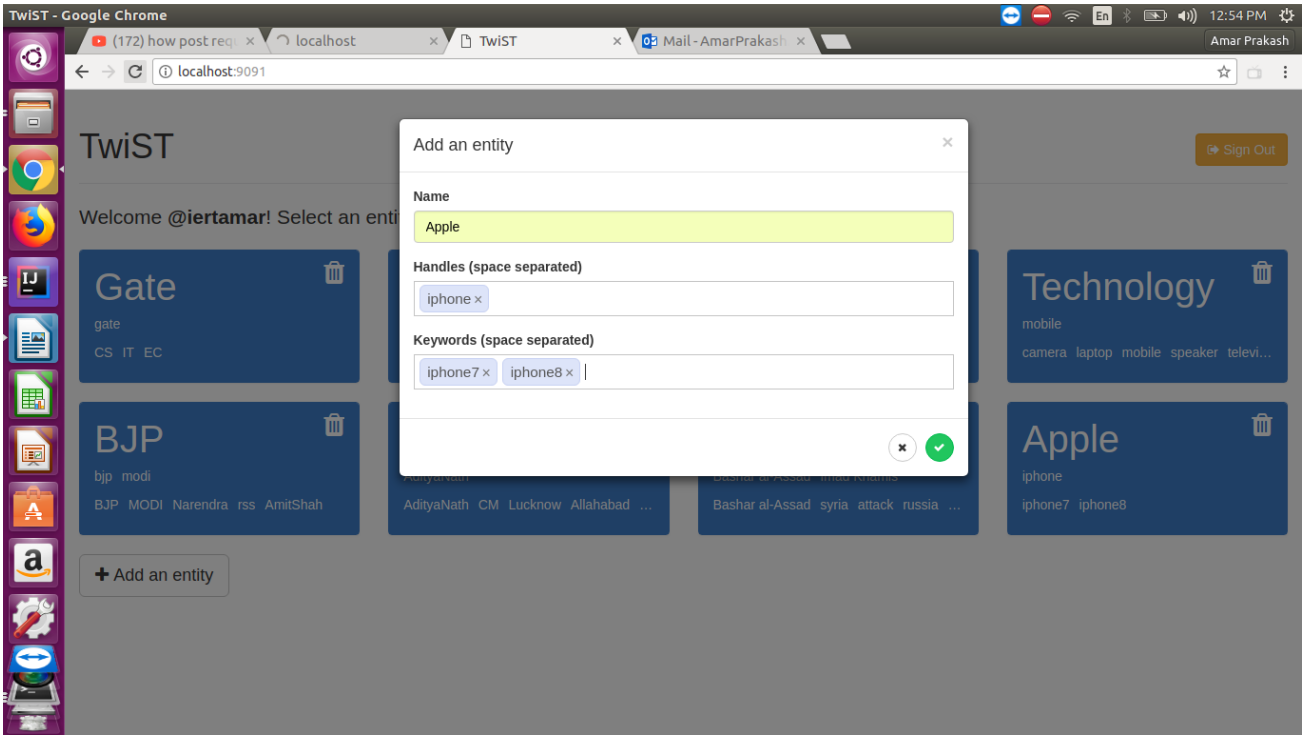
4.2 Contacting twitter and Input to the Application



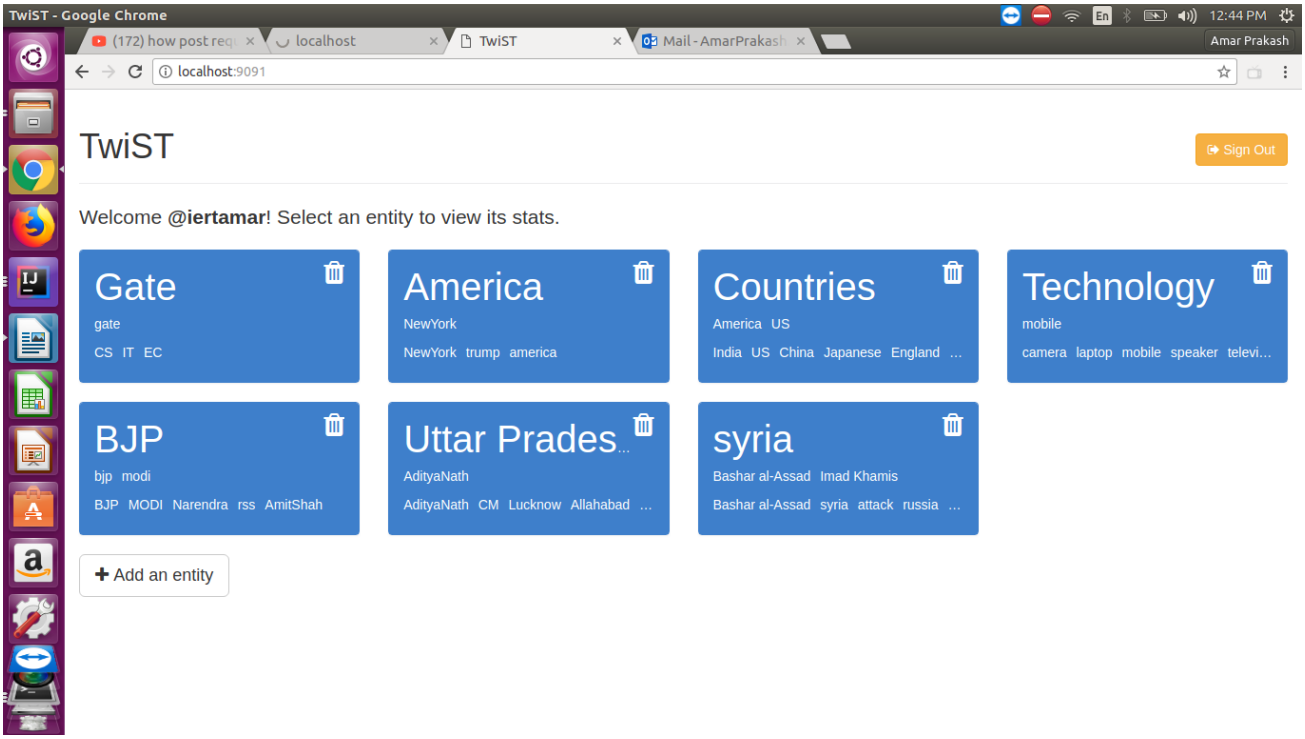
This figure shows the front page of our project when user will click on log in button . It will first contact to twitter to verify twitter API .



This figure shows that after verifying twitter api we will take email id and password from the user to verify whether user is a valid user or not by using OAuth protocol .



This figure shows that after verifying user how will we take the input from the user .

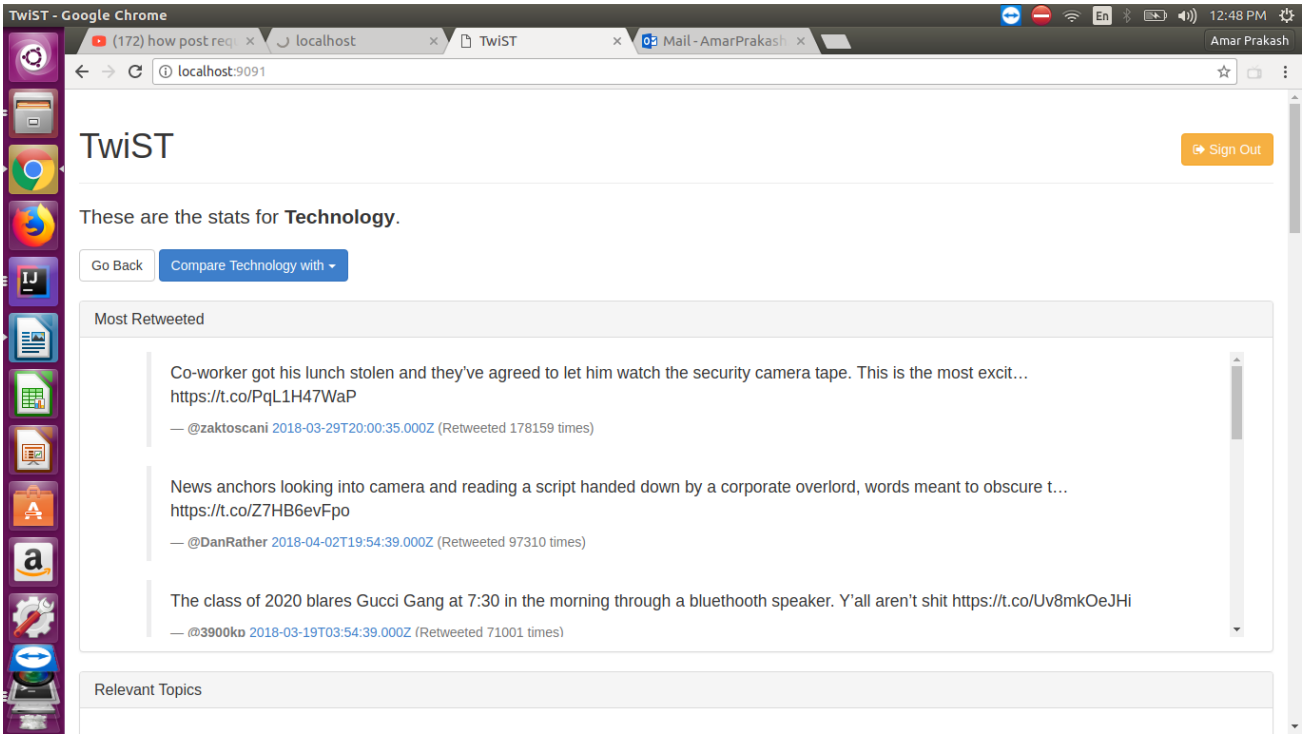


This figure shows all the entities added by particular user .

Chapter 5

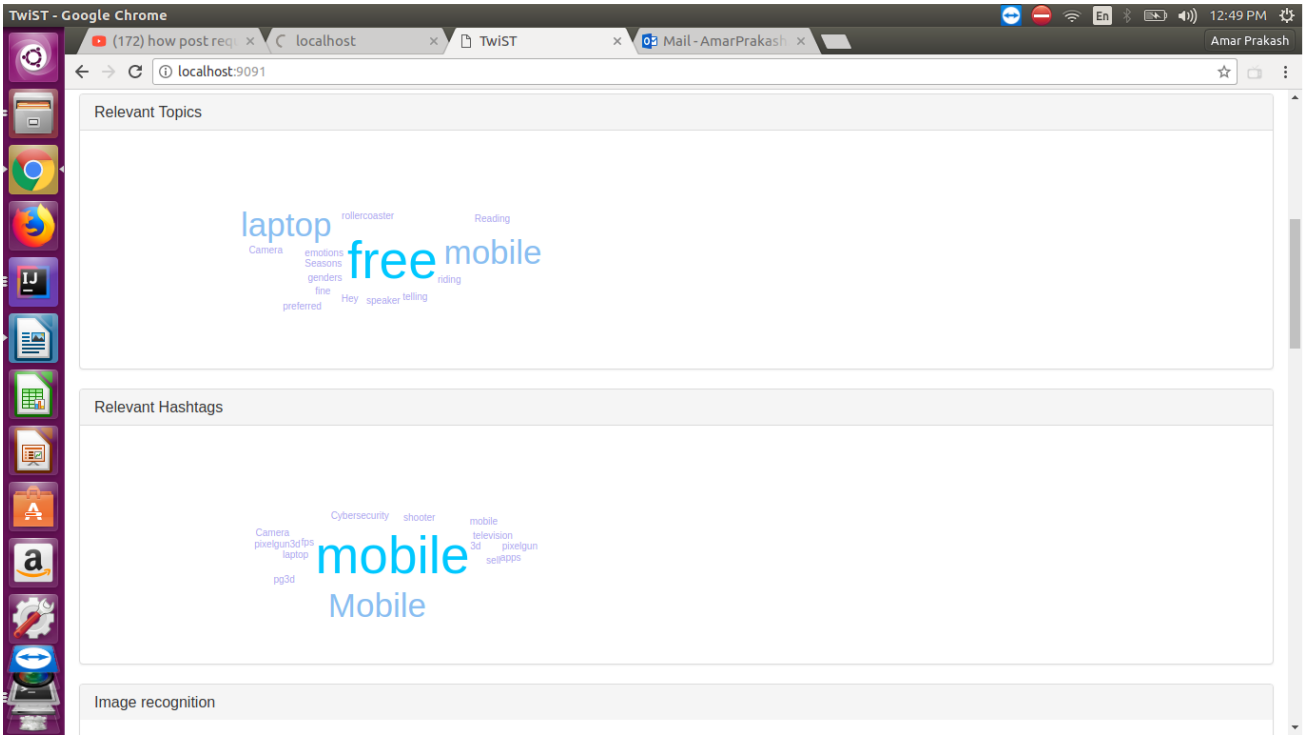
Screenshots of Project

Following are some screen shots which shows output for any particular entity provided by user .



MOST RETWEETED

This is the screen shot of most retweeted tweet which shows all those tweets which are frequently retweeted by people for a particular entity .



RELEVANT TOPICS

This is the screen shot of relevant topics which shows what are all the topics about which people are frequently talking . Some topic names are showing bigger size means those topic names are most frequently used by the people.

RELEVANT HASHTAGS

This is the screen shot of relevant hashtags which shows what are all the hashtags which are present in the tweet for any particular entity.Some hashtags are showing bigger size means those hashtags are most frequently present on the tweets .

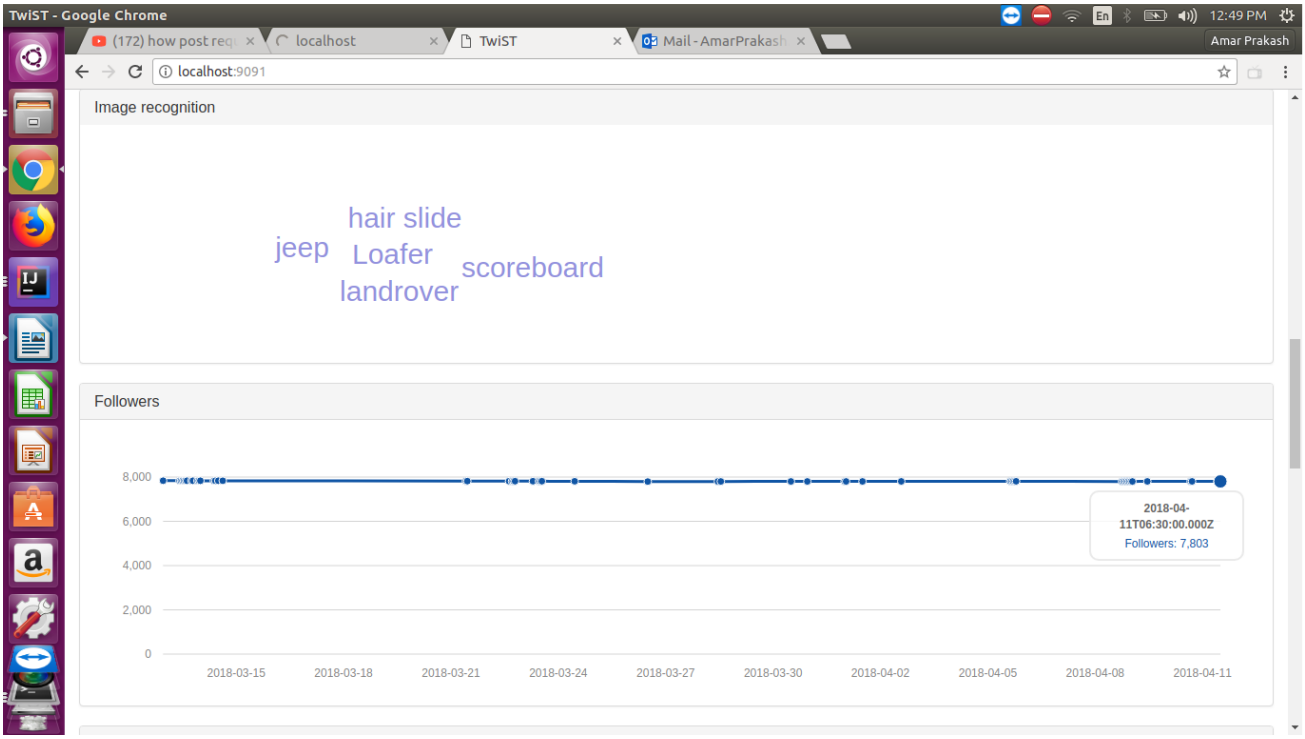
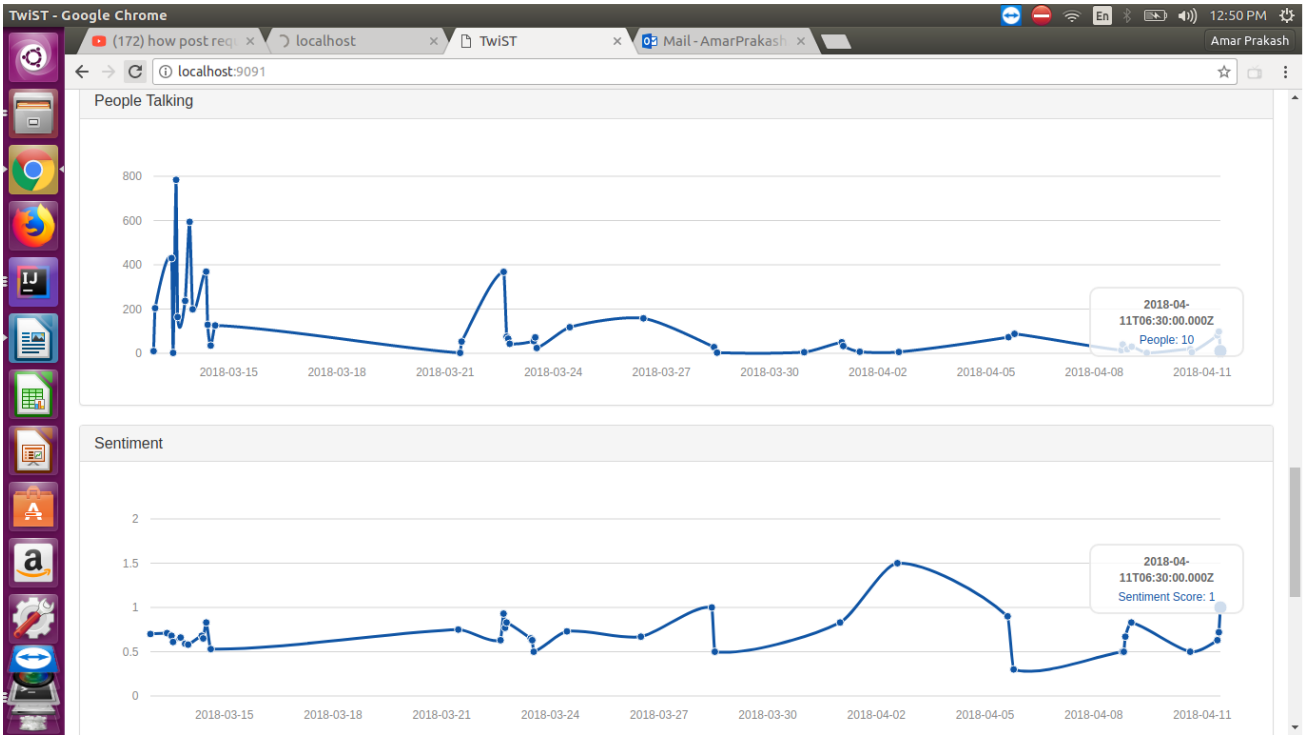


IMAGE RECOGNITION

This is the screen shot of name of images which shows what are all the images which are frequently uploaded by people for any particular entity. Some image names are showing bigger size means type of images are most frequently uploaded by the people.

FOLLOWERS

This is the screen shot of followers which shows that how many people are following for any particular entity provided by the user . In the X-axis it is showing dates which have 3 days interval and in the Y-axis shows the number of people which are following .

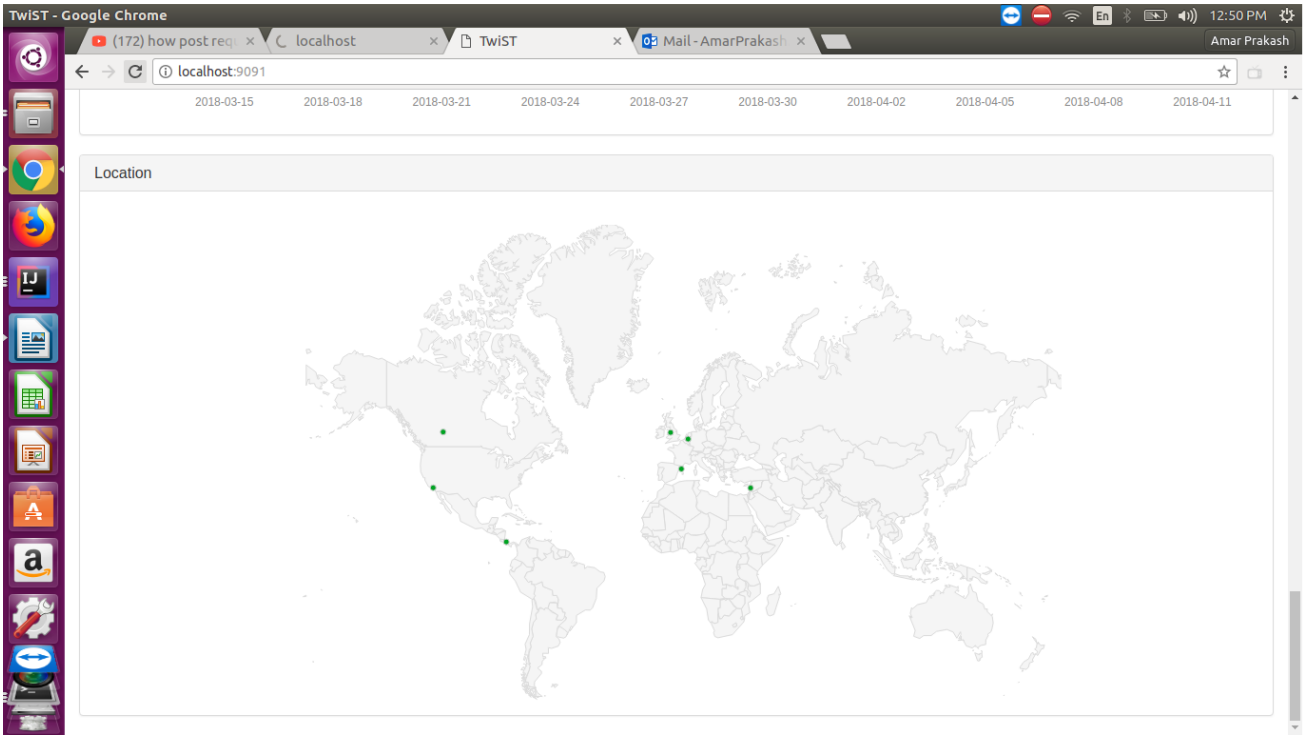


PEOPLE TALKING

This is the screen shot of the people talking. It shows how many people are talking about your keyword in a defined time interval.

SENTIMENT ANALYSIS

This is the screen shot of sentiment of the people. The process of identifying and categorizing opinions expressed in a piece of text (tweets).



LOCATION

This is the screen shot of followers which shows the location of the user. If you are getting some response then from which Geographical Location you are getting most of the response

Chapter 6

Tools and Technology

Web APP HTML , CSS , Java-Script , AngularJS
Data Fetch API JAVA , Play Framework
RabbitMq Queuing Technology (Open Source Software)
Maven tool
Stanford NLP library for sentiment analysis
twitter4j library for fetching the tweets
TensorFlow library for image recognition

Chapter 7

Appendices

7.1 What is twitter 4j API

. Twitter4J is an unofficial Java library for the Twitter API. With Twitter4J, you can easily integrate your Java application with the Twitter service. Twitter4J is an unofficial library

7.2 What is RabbitMq?

RabbitMQ is a messaging broker - an intermediary for messaging. It gives your applications a common platform to send and receive messages, and your messages a safe place to live until received.

Messaging enables software applications to connect and scale. Applications can connect to each other, as components of a larger application, or to user devices and data. Messaging is asynchronous, decoupling applications by separating sending and receiving data.

RabbitMQ offers a variety of features to let you trade off performance with reliability, including persistence, delivery acknowledgements, publisher confirms, and high availability.

7.3 What is consumer key, consumer secret, token key and token secret

Consumer key and consumer secret are the keys for the Authentication of the application so that Twitter can verify that this application is running by the authenticated user

token key and token secret, these are the identity for the user when user will enter his/her username and password through our application to the twitter then twitter

will generate token key and token secret and return to the application

7.4 How to get consumer key, consumer secret, Token key and Token secret

i. Log into the Twitter Developers section.

ii. If you don't already have an account, you can login with your normal Twitter credentials Go to "Create an app"

iii. Fill in the details of the application you'll be using to connect with the API

iv. Your application name must be unique. If someone else is already using it, you won't be able to register your application until you can think of something that isn't being used. Click on Create your Twitter application

v. Details of your new app will be shown along with your consumer key and consumer secret.

vi. If you need access tokens, scroll down and click Create my access token

vii. The page will then refresh on the "Details" tab with your new access tokens. You can recreate these at any time if you need to.

References

[1] *Agarwal, A., Xie, B., Vovsha, I., Rambow, O., Passonneau, R.: Sentiment analysis of twitter data. In: Proc. ACL 2011 Workshop on Languages in Social Media*

[2] *Gimpel, K., Schneider, N., OConnor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., Smith, N.: Part-of-speech tagging for twitter: Annotation, features, and experiments. Tech. rep., DTIC Document*

[3] *Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford*

[4] *Lin, C., He, Y.: Joint sentiment/topic model for sentiment analysis. In: Proceeding of the 18th ACM Conference on Information and Knowledge Management*