# Principles of Big Data

## Project- Phase1

https://github.com/SAISRIHARSHAS/PB-Project


## Team -16

Team Members:

Sai Sriharsha Sudulaguntla (16233059)
Sankarasetty Avinash (16233012)
Sai Mohith Reddy Chagamreddy (16233203)
Lava Kumar  Surparaju

# Project Overview:

Develop a system to store, analyse, and visualize a social network's (e.g. Twitter's) data.

This project is divided into three development phases.

## Phase 1 includes collection of tweets, running the word count in Apache Spark.

## Tasks Performed:

1. Created Developer account on Twitter.com, for access keys and secret tokens.
2. We used tweepy API to collect streaming tweets in python code, to write tweet text to a document.

codeForTweets.py - C:\Users\LENOVO\Desktop\Phase1\Code\codeForTweets.py (3.5.2)

File Edit Format Run Options Window Help

```
from tweepy import Stream
from tweepy import OAuthHandler
from tweepy.streaming import StreamListener
import time
import json


ckey="ycCB7onSCfcA1e9K0ZUzVY2GA"
csecret="bzZBxbycwIqyB97UOA5llVZcG7OFczlwVU0Bvmvp1tUOoGw93W"
atoken="768298204567711744-8REnbwuK8dPT8LLRMm2hlMFbesnWbWK"
asecret="uX4si9CirS4pp8OhFME3ZaIwyfvT7bPyFcZh0KRyCWjZX"

class listener(StreamListener):
    def on_data(self, data):

        try:
            tweet_data = json.loads(data)
            tweet = tweet_data["text"]
            writeTweets = open('tweet_p1.json','a')
            writeTweets.write(tweet)
            writeTweets.close()
            return(True)

        except Exception as e:
            time.sleep(1)

        def on_error(self, status):
            print status

auth = OAuthHandler(ckey, csecret)
auth.set_access_token(atoken, asecret)

twitterStream = Stream(auth, listener())
twitterStream.filter(track=["#MissAmerica,#Patriots,#WWEBacklash"])
```

3. Used OAuth to authorize the python code for the purpose of connecting to the twitter.
4. Collected the tweets with using current Trending keywords, used StreamListner().

# Word Count:

1. Used IntelliJ IDEA tool to do the word count in Apache Spark using Scala.

input × wordcount.scala ×

```scala
/**
  * Created by Avinash on 9/8/2016.
  */

import org.apache.spark.{SparkContext, SparkConf}
object wordcount {
  def main(args: Array[String]) {
    System.setProperty("hadoop.home.dir", "E:\\winutils");
    val sparkConf = new SparkConf().setAppName("SparkWordCount").setMaster("local[*]")
    val sc = new SparkContext(sparkConf)
    val input = sc.textFile("C:\\Users\\Avinash\\Desktop\\tweet_p1.txt")
    val wc=input.flatMap(line=>{line.split(" ")}).map(word=>(word,1)).cache()
    val output=wc.reduceByKey(_+_).sortByKey()
    output.saveAsTextFile("C:\\Users\\Avinash\\Desktop\\Output1")
    val o=output.collect()
    var s:String="Words:Count \n"
    o.foreach{case(word,count)=>{
      s+=word+" : "+count+"\n"
    }}
    //println(s)
  }
}
```

# Output:

Received the word count, stored in output folder.

Output sample snippet:

```
(#PPVAB1@NEPD_Loyko,1)
(#PPVAB1RT,6)
(#Packers@AJStylesOrg,1)
(#Pakistan,2)
(#PanicoNaBand,2)
(#PatriotDay,1)
(#Patriots,2448)
(#Patriots!!!!!!!RT,1)
(#Patriots#BackLash,1)
(#Patriots#Cowboys,1)
(#Patriots#MissAmerica,3)
(#Patriots#Patriots,1)
(#Patriots#WWE:,1)
(#Patriots#WWEBacklash,3)
(#Patriots#mondaymotivation,1)
(#Patriots,,1)
(#Patriots.,4)
(#Patriots...,1)
(#Patriots1st,1)
(#PatriotsA,1)
(#PatriotsCan,1)
(#PatriotsGame,1)
(#PatriotsHAHA,1)
(#PatriotsHe,1)
(#PatriotsI'm,1)
(#PatriotsIf,1)
(#PatriotsIt's,1)
(#PatriotsJust,1)
(#PatriotsMST,1)
(#PatriotsNEW,1)
(#PatriotsNation,11)
```