**Report**

**Theme of the Project:** The project is based on the concept of US Elections of 2016 and Trump’s stand in it from Twitter. The data is then stored in HDFS to be accessed by Spark API.

**Language used for Project**:

1. Server programming language: Java

2. Front end language: HTML5, CSS3 & BootStrap, JavaScript and D3.js(library)

**Libraries and API’s:**

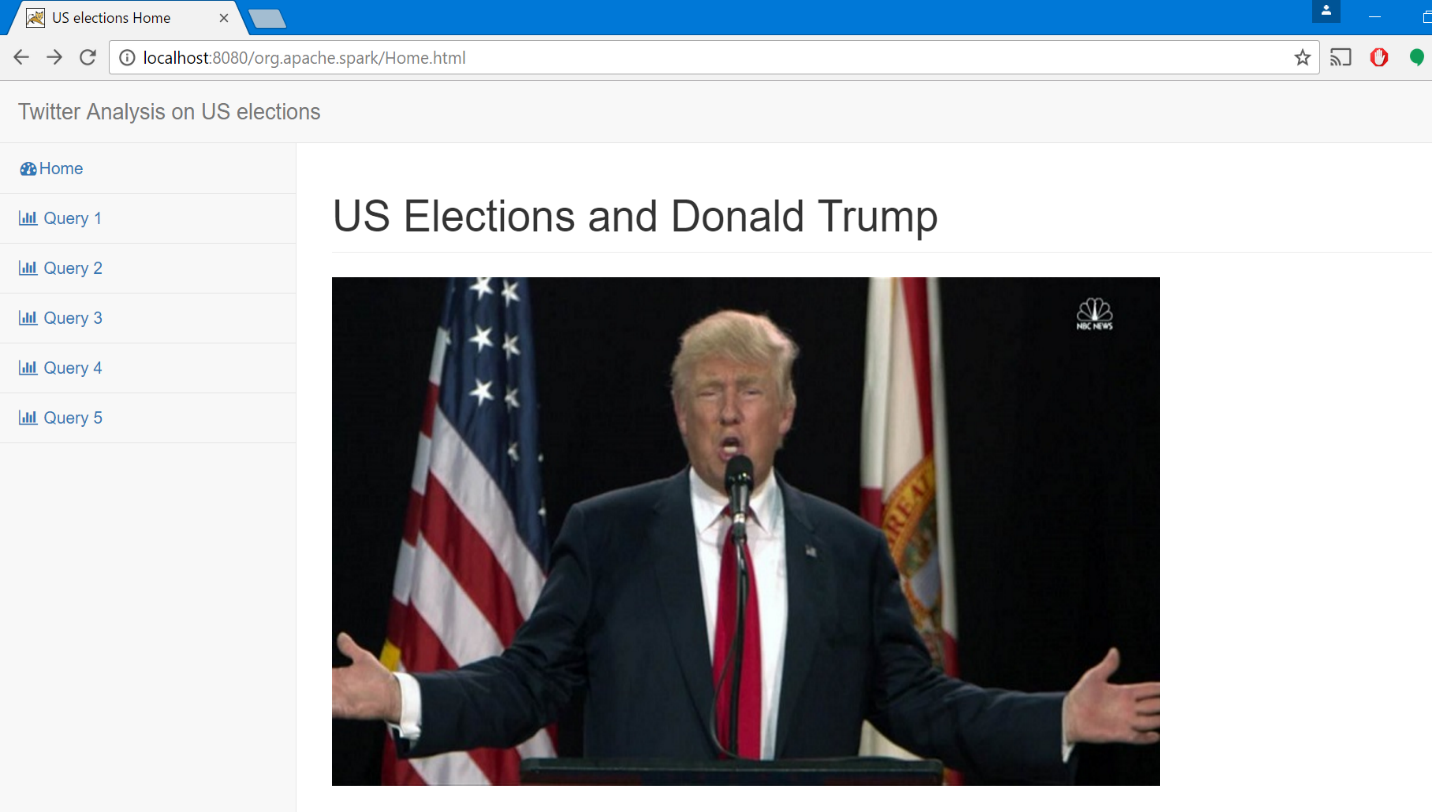
1. We have used D3.js to represent data in the form of charts in the front end. It is a library just like JQuery.

<script src=*"http://d3js.org/d3.v3.min.js"*></script>

<script src =*"https://cdnjs.cloudflare.com/ajax/libs/nvd3/1.8.1/nv.d3.min.js"*> </script>

**Home page:**

This is the page where we have created interface for all the queries to be excecuted.



**Query 1: Sentiment Analysis of Donald Trump’s stand in US elections**

1. API Used: Alchemy API of Watson Service from IBM

2. Description: This query gives the sentiment of overall collected data which is can be either positive or negative. The data that is shown is below screen shot shows that Donald Trump has not so good opinion from the users that the data consists of.

3. Implementation:

**protected** **void** query1PosNeg(JavaSQLContext sqlContext) {

// **TODO** Auto-generated method stub

**try**

{

File outputFile = **new** File(getServletContext().getRealPath("/")

+ "/q10.JSON");

FileWriter fw= **new** FileWriter(outputFile);

JavaSchemaRDD tweetdata = sqlContext.sql("SELECT DISTINCT text FROM tweetTable ");

String textTweet = tweetdata.toString();

System.***out***.println(textTweet);

AlchemyLanguage service = **new** AlchemyLanguage();

service.setApiKey("29131b737a57f19004e3728cb2c16900bd598919");

Map<String,Object> params = **new** HashMap<String, Object>();

params.put(AlchemyLanguage.***TEXT***, textTweet);

DocumentSentiment sentiment = service.getSentiment(params).execute();

System.***out***.println(sentiment);

String sentimentData = sentiment.toString();

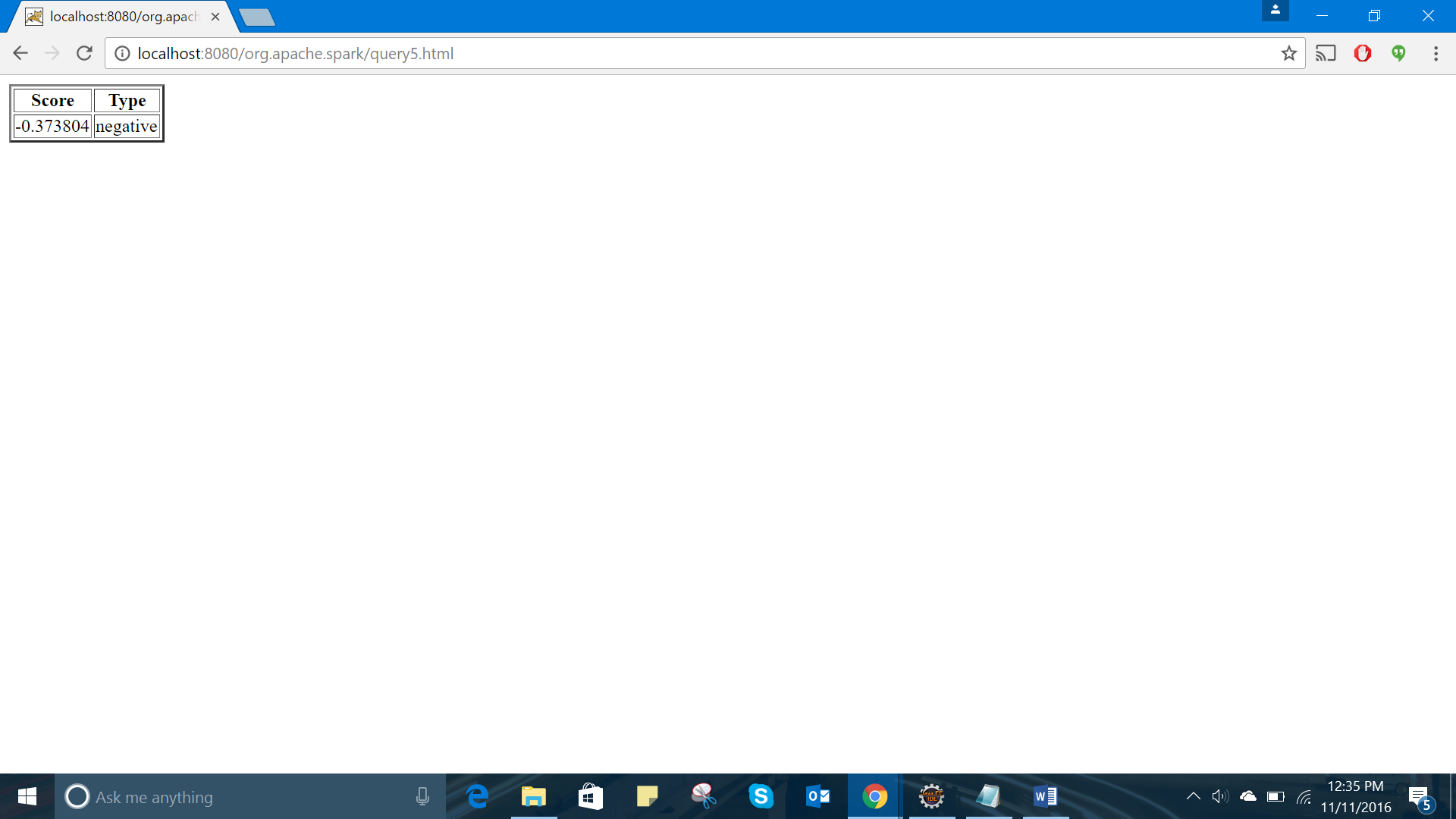
fw.write(sentimentData);

fw.close();

}

The above method in java establishes a connection with Alchemy API using the key and gets the text from tweet table to display the sentiment of the data.

4. Screen Shot:

It is in the format of a table

**Query 2: Popular Users with high Follower count**

1. Description: This chart describes about the “Popular Users with High Follower Count”. It checks in the data for the users who have Users with high follower count of all those which match the selected theme.

2. Implementation:

**try**

{

File outputFile = **new** File(getServletContext().getRealPath("/")+ "/q2.csv");

FileWriter fw= **new** FileWriter(outputFile);

JavaSchemaRDD count = sqlContext.sql("SELECT DISTINCT user.screen\_name, user.followers\_count AS c FROM tweetTable " +

"ORDER BY c");

List<org.apache.spark.sql.api.java.Row> rows = count.collect();

Collections.*reverse*(rows);

String rows123=rows.toString();

String[] array = rows123.split("],");

fw.append("Name");

fw.append(',');

fw.append("Count");

fw.append("\n");

**for**(**int** i = 0; i < 8; i++)

{

**if**(i==0)

{

fw.append(array[0].substring(2));

fw.append(',');

fw.append("\n");

System.***out***.println(array[0].substring(2));

}

**else** {

fw.append(array[i].substring(2));

fw.append(',');

fw.append("\n");

System.***out***.println(array[i].substring(2));

}

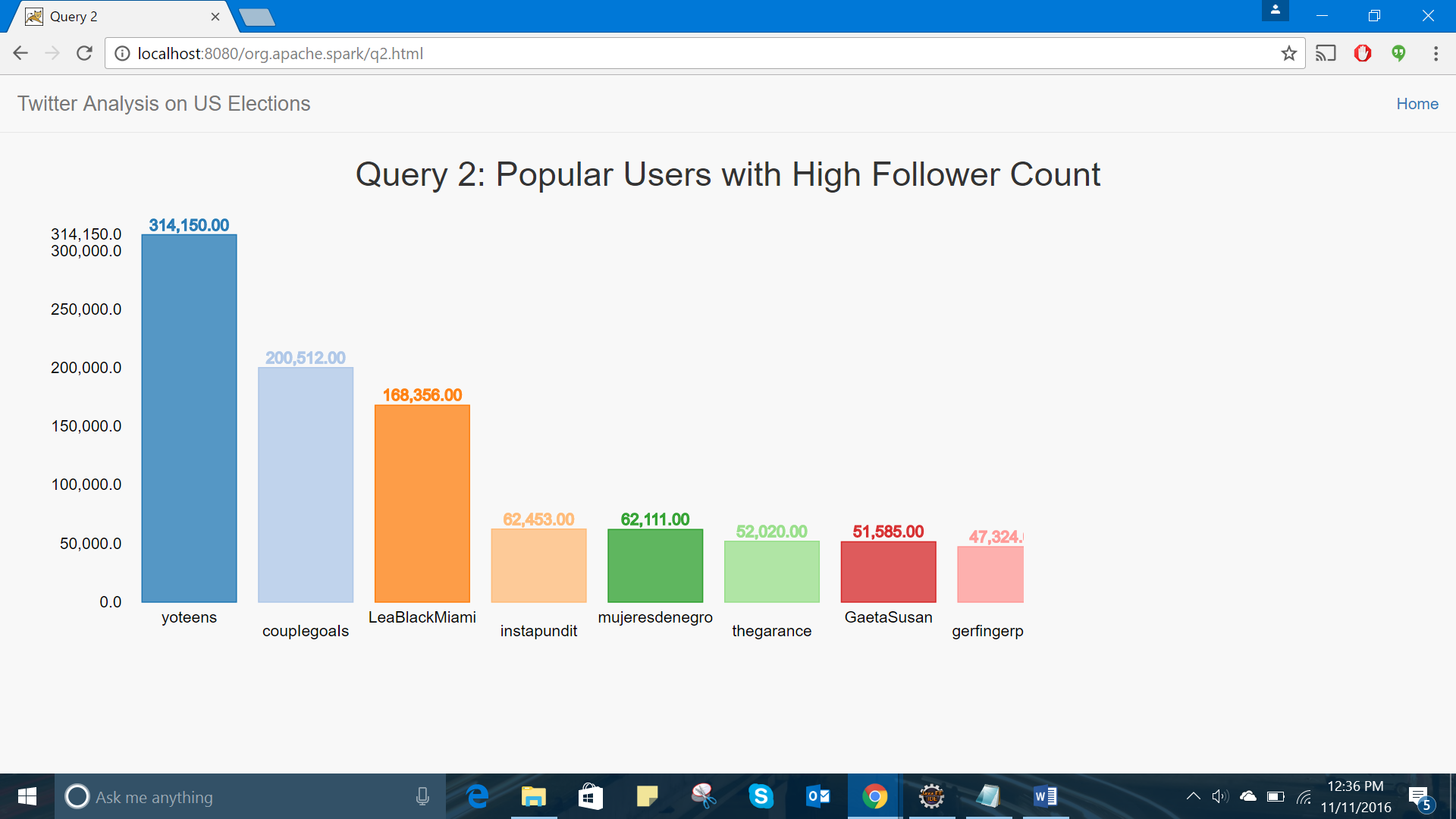
}

fw.close();

}

3. ScreenShot:

It is in the form of a bar chart



**Query 3: Most active Users with high tweet count**

1. Description: This query gives the users who tweet very often

2. Implementation:

**try**

{

File outputFile = **new** File(getServletContext().getRealPath("/")+"/q3.csv");

FileWriter fw= **new** FileWriter(outputFile);

JavaSchemaRDD count = sqlContext.sql("SELECT DISTINCT user.name,user.statuses\_count AS c FROM tweetTable " +

"ORDER BY c");

List<org.apache.spark.sql.api.java.Row> rows = count.collect();

Collections.*reverse*(rows);

String rows123=rows.toString();

String[] array = rows123.split("],");

fw.append("Name");

fw.append(',');

fw.append("Count");

fw.append("\n");

**for**(**int** i = 0; i < 8; i++)

{

**if**(i==0)

{

fw.append(array[0].substring(2));

fw.append(',');

fw.append("\n");

}

**else** {

fw.append(array[i].substring(2));

fw.append(',');

fw.append("\n");

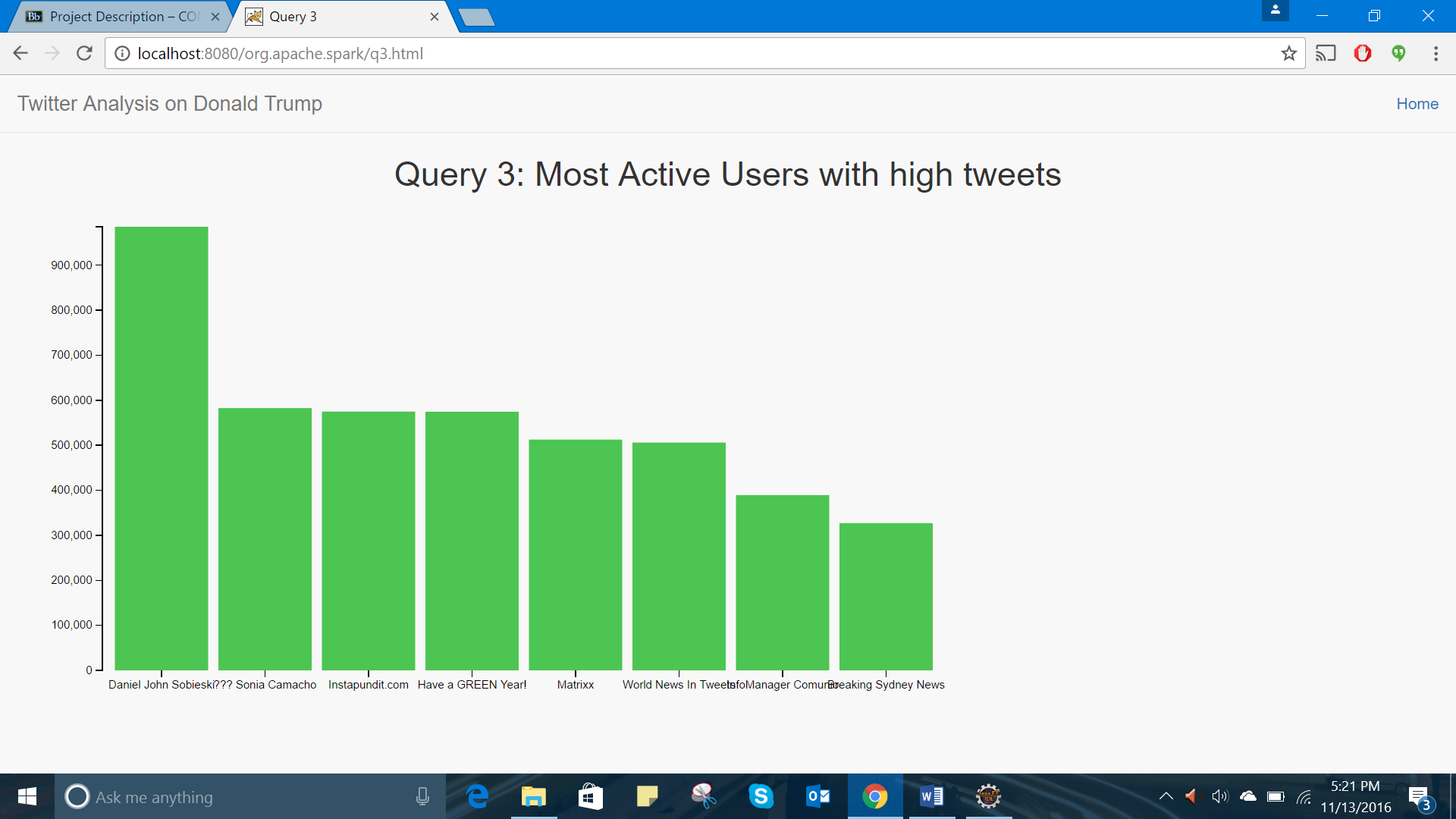
System.***out***.println(array[i].substring(2));

}

}

3. ScreenShot:

It is in the form of a bar chart with single color to represent active users with high tweets



**Query 4: Most common timings of users tweeting**

1. Description: This query gives the peak timings in which users tweet the most

2. Implementation:

**try**

{

File outputFile = **new** File(getServletContext().getRealPath("/")

+ "/q5.csv");

FileWriter fw= **new** FileWriter(outputFile);

JavaSchemaRDD count = sqlContext.sql("SELECT created\_at, COUNT(\*) AS c FROM tweetTable " +

"Group By created\_at " +

"order by c" );

List<org.apache.spark.sql.api.java.Row> rows = count.collect();

Collections.*reverse*(rows);

String rows123=rows.toString();

String[] array = rows123.split("],");

System.***out***.println(rows123);

fw.append("Time");

fw.append(',');

fw.append("Count");

fw.append("\n");

**for**(**int** i = 0; i < 9; i++)

{

**if**(i==0)

{

**continue**;

}

**else** **if**(i == array.length-1)

{

fw.append(array[i].substring(2,array[i].length()-2));

fw.append(',');

fw.append("\n");

}

**else** {

fw.append(array[i].substring(2));

fw.append(',');

fw.append("\n");

}

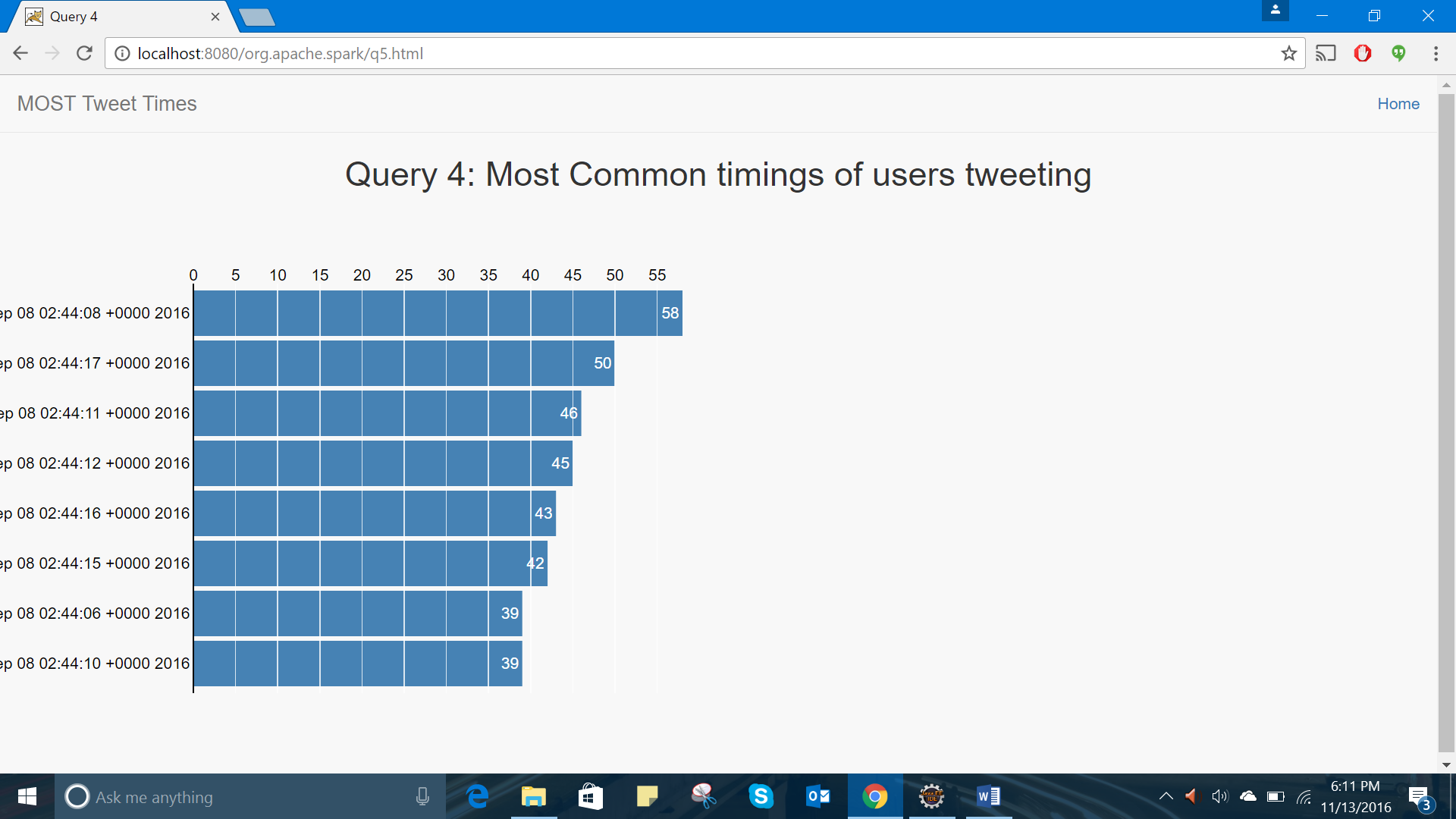
}

fw.close();

}

3. ScreenShot:

It is a timeline diagram with horizontal bar chart



**Query 5: Language that was most used to tweet about Donald Trump**

1. Description: This query gives the language in which the users tweet the most

2. Implementation:

FileWriter fw= **new** FileWriter(outputFile);

JavaSchemaRDD count = sqlContext.sql("SELECT lang, COUNT(\*) AS c FROM tweetTable " +

"Group By lang " +

"order by c");

List<org.apache.spark.sql.api.java.Row> rows = count.collect();

Collections.*reverse*(rows);

String rows123=rows.toString();

String[] array = rows123.split("],");

System.***out***.println(rows123);

fw.append("Language");

fw.append(',');

fw.append("Count");

fw.append("\n");

**for**(**int** i = 0; i < 9; i++)

{

**if**(i==0)

{

**continue**;

}

**else** **if**(i == array.length-1)

{

fw.append(array[i].substring(2,array[i].length()-2));

fw.append(',');

fw.append("\n");

}

**else** {

fw.append(array[i].substring(2));

fw.append(',');

fw.append("\n");

}

}

fw.close();

3. ScreenShot:

It is a bar chart representing language used for tweeting

