Codelab RESUBMISSION(DATA DRIVEN APPLICATION) .

Name: Muhamad Mueez Amer Iqbal

Student Number : 415772

Description:

We have been approached to build up an application that cross examines tweets. We have two alternatives for finishing this task. For the two choices the application ought to be intended to utilize capacities, and pass contentions between these as proper. Where conceivable the application ought to be likewise executing object arranged programming techniques. The last application have to be conveyed through a working intuitive GUI fabricated utilizing open Frameworks. This GUI ought to permit the client to communicate by means of mouse or potentially console input. The GUI is also included in it and the open frame work is also there with the code. The code is viewable and run able. Shows all the function of the commands mentioned there.

There are Six models that have effectively been given to us and four more should be implemented by us. The six are as below:

• Count the quantity of tweets in the organizer

• Counting the quantity of tweets which includes "Money"

• Counting the quantity of tweets which includes "Politics"

• Printing on the screen any tweets which includes "Paris"

• Printing on the screen any tweets which includes "Dreamworks"

• Printing on the screen any tweets which includes "Uber"

And the other Four are as below:

* Counting the quantity of tweets which includes “Elections”.
* Counting the quantity of tweets which includes “Specific UK”.
* Printing on the screen any tweets which includes “BreakingNews”.
* Printing on the screen any tweets which includes “Hometown”.

**Project Idea:**

**Beginning with the primary week and a bit of the cycle have been played to include the arranging stage, picking where to run certain systems that may add to the overall progression in the improvement of the application. Following this, implies should be taken to address different people and portray what needs the application would need to satisfy. At whatever point this is done, starting models can be drawn up that prepare for the writing computer programs is to be in progress.**



**Specialized Description:**

**The essential concern of the application is that it ought to have the alternative to examine in data from the sheet sufficiently. This suggests that preceding whatever else can happen, the <fstream> addon ought to be initialised at the real start of the program. Moreover, we will scrutinize in data and not yielding anything to the sheet, so we will use the 'ifstream' work. Following this, the issue goes to how the total of the data will be taken care of inside the program. A bunch could be used, yet if the size of the data sheet to change, this would in like manner suggest that the size of the show would should be changed each time this is done. To keep an essential separation from this, we can rather make usage of a vector to store the characteristics. The date/period of the tweet can be associated with the genuine tweet likewise, crashed into the vector to store it, by then reiterated with 'Some time' circle until the completion of the archive has been reached. A string stream variable would moreover be required to interface these, which would be cleared during each pattern of the circle to make a way for the accompanying components.**

CODE:

#include<iostream>

#include<fstream>

#include<string.h>

#include<vector>

#include<string>

using namespace std;

int main()

{

int count = 0, i;

string line;

int a;

string stopApp;

string hometowntweetHolder;

vector<string> hometowntweet, hometowntweetHolder2;

int hometowncount1 = 0;

string UbertweetHolder;

vector<string> Ubertweet, UbertweetHolder2;

string DreamWorkstweetHolder;

vector<string> DreamWorkstweet, DreamWorkstweetHolder2;

string hashtagtweetHolder;

vector<string> hashtagtweet, hashtagtweetHolder2;

string specificUsertweetHolder;

vector<string> specificUsertweet, specificUsertweetHolder2;

int count1 = 0;;

string locationtweetHolder;

vector<string> locationtweet, locationtweetHolder2;

int count2 = 0;

string DonaldtweetHolder;

vector<string> Donaldtweet, DonaldtweetHolder2;

string BreakingNewstweetHolder;

vector<string> BreakingNewstweet, BreakingNewstweetHolder2;

string sepIssuetweetHolder;

vector<string> sepIssuetweet, sepIssuetweetHolder2;

string augusttweetHolder;

vector<string> augusttweet, augusttweetHolder2;

string UKtweetHolder;

vector<string> UKtweet, UKtweetHolder2;

string christmastweetHolder;

vector<string> christmastweet, christmastweetHolder2;

int christmascount1 = 0;

fstream data10("sampleTweets.csv");

if (data10.is\_open())

{

while (!data10.eof())

{

getline(data10, hometowntweetHolder);

hometowntweet.push\_back(hometowntweetHolder);

}

data10.close();

}

else

{

cout << "not found" << endl;

}

ifstream file("sampleTweets.csv");

ifstream data9("sampleTweets.csv");

if (data9.is\_open())

{

while (!data9.eof())

{

getline(data9, UbertweetHolder);

Ubertweet.push\_back(UbertweetHolder);

}

data9.close();

}

else

{

cout << "not found" << endl;

}

ifstream data8("sampleTweets.csv");

if (data8.is\_open())

{

while (!data8.eof())

{

getline(data8, DreamWorkstweetHolder);

DreamWorkstweet.push\_back(DreamWorkstweetHolder);

}

data8.close();

}

else

{

cout << "not found" << endl;

}

ifstream data("sampleTweets.csv");

if (data.is\_open())

{

while (!data.eof())

{

getline(data, hashtagtweetHolder);

hashtagtweet.push\_back(hashtagtweetHolder);

}

data.close();

}

else

{

cout << "not found" << endl;

}

ifstream data1("sampleTweets.csv");

if (data1.is\_open())

{

while (!data1.eof())

{

getline(data1, specificUsertweetHolder);

specificUsertweet.push\_back(specificUsertweetHolder);

}

data1.close();

}

else

{

cout << "not found" << endl;

}

ifstream data2("sampleTweets.csv");

if (data2.is\_open())

{

while (!data2.eof())

{

getline(data2, locationtweetHolder);

locationtweet.push\_back(locationtweetHolder);

}

data2.close();

}

else

{

cout << "not found" << endl;

}

ifstream data3("sampleTweets.csv");

if (data3.is\_open())

{

while (!data3.eof())

{

getline(data3, DonaldtweetHolder);

Donaldtweet.push\_back(DonaldtweetHolder);

}

data3.close();

}

else

{

cout << "not found" << endl;

}

ifstream data4("sampleTweets.csv");

if (data4.is\_open())

{

while (!data4.eof())

{

getline(data4, BreakingNewstweetHolder);

BreakingNewstweet.push\_back(BreakingNewstweetHolder);

}

data4.close();

}

else

{

cout << "not found" << endl;

}

ifstream data5("sampleTweets.csv");

if (data5.is\_open())

{

while (!data5.eof())

{

getline(data5, sepIssuetweetHolder);

sepIssuetweet.push\_back(sepIssuetweetHolder);

}

data5.close();

}

else

{

cout << "not found" << endl;

}

ifstream data6("sampleTweets.csv");

if (data6.is\_open())

{

while (!data6.eof())

{

getline(data6, augusttweetHolder);

augusttweet.push\_back(augusttweetHolder);

}

data6.close();

}

else

{

cout << "not found" << endl;

}

ifstream data7("sampleTweets.csv");

if (data7.is\_open())

{

while (!data7.eof())

{

getline(data7, UKtweetHolder);

UKtweet.push\_back(UKtweetHolder);

}

data7.close();

}

else

{

cout << "not found" << endl;

}

do {

cout << endl << "....................................Menu...................................." << endl;

cout << endl << "Press Number Which operation you want!" << endl;

cout << "1-Count the total number of tweets in the data set" << endl;

cout << "2-Count the number of tweets that include money" << endl;

cout << "3-Count the number of tweets that include politics" << endl;

cout << "4-Print to the screen any tweets that include Paris " << endl;

cout << "5-Print to the screen any tweets that include DreamWorks " << endl;

cout << "6-Print to the screen any tweets that include Uber " << endl;

cout << "7-Display tweets about Election " << endl;

cout << "8-Display specific UK tweets " << endl;

cout << "9-Display Breaking News tweets " << endl;

cout << "10-Total number of hometown tweets" << endl;

cin >> a;

switch (a)

{

case 1:

cout << endl << "...............1-Total Number of tweets in data set...................." << endl;

while (getline(file, line))

{

count++;

}

cout << "1-Total Numbers of Tweets in the file : " << count << endl;

break;

case 2:

cout << endl << "................2-Count the number of tweets that include money..........." << endl;

for (int j = 0; j < specificUsertweet.size(); j++)

{

string str2 = specificUsertweet[j];

if (str2.find("money") != string::npos)

{

specificUsertweetHolder2.push\_back(specificUsertweet[j]);

//cout<<endl<<specificUsertweet[j]<<endl;

count1++;

}

}

cout << "2-Total Numbers of Tweets that include money are : " << count1 << endl;

break;

case 3:

cout << endl << ".......................3-Count the number of tweets that include politics........................... " << endl;

for (int k = 0; k < locationtweet.size(); k++)

{

string str3 = locationtweet[k];

if (str3.find("politics") != string::npos)

{

locationtweetHolder2.push\_back(locationtweet[k]);

//cout<<endl<<locationtweet[k]<<endl;

count2++;

}

}

cout << "3-Total Numbers of Tweets that include politics are : " << count2 << endl;

break;

case 4:

cout << endl << ".......................4-Print to the screen any tweets that include Paris........................... " << endl;

for (int k = 0; k < locationtweet.size(); k++)

{

string str3 = locationtweet[k];

if (str3.find("paris") != string::npos)

{

locationtweetHolder2.push\_back(locationtweet[k]);

cout << endl << locationtweet[k] << endl;

}

}

break;

case 5:

cout << endl << "...............5-Display tweets that include DreamWorks...................." << endl;

for (int i = 0; i < DreamWorkstweet.size(); i++)

{

string str1 = DreamWorkstweet[i];

if (str1.find("DreamWorks") != string::npos)

{

DreamWorkstweetHolder2.push\_back(DreamWorkstweet[i]);

cout << endl << DreamWorkstweet[i] << endl;

//moneycount1++;

}

}

break;

case 6:

cout << endl << "...............6-Display tweets that include uber...................." << endl;

for (int i = 0; i < Ubertweet.size(); i++)

{

string str1 = Ubertweet[i];

if (str1.find("Uber") != string::npos)

{

UbertweetHolder2.push\_back(Ubertweet[i]);

cout << endl << Ubertweet[i] << endl;

//moneycount1++;

}

}

break;

case 7:

cout << endl << "...............7-Display tweets about election...................." << endl;

for (int i = 0; i < augusttweet.size(); i++)

{

string str1 = augusttweet[i];

if (str1.find("election") != string::npos)

{

augusttweetHolder2.push\_back(augusttweet[i]);

cout << endl << augusttweet[i] << endl;

//moneycount1++;

}

}

break;

case 8:

cout << endl << "...............8-Display specific UK tweets...................." << endl;

for (int i = 0; i < UKtweet.size(); i++)

{

string str1 = UKtweet[i];

if (str1.find("UK") != string::npos)

{

UKtweetHolder2.push\_back(UKtweet[i]);

cout << endl << UKtweet[i] << endl;

//moneycount1++;

}

}

break;

case 9:

cout << endl << "...............9-Display Breaking News Tweets...................." << endl;

for (int i = 0; i < BreakingNewstweet.size(); i++)

{

string str1 = BreakingNewstweet[i];

if (str1.find("breaking news") != string::npos)

{

BreakingNewstweetHolder2.push\_back(BreakingNewstweet[i]);

cout << endl << BreakingNewstweet[i] << endl;

//moneycount1++;

}

}

break;

case 10:

for (int j = 0; j < hometowntweet.size(); j++)

{

string str8 = hometowntweet[j];

if (str8.find("hometown") != string::npos)

{

hometowntweetHolder2.push\_back(hometowntweet[j]);

//cout<<politicstweet[j]<<endl;

hometowncount1++;

}

}

cout << endl << "10-Total number of hometown tweets= " << hometowncount1 << endl;

break;

}

cout << "Do you want to continue? (Y/N)" << endl;

cin >> stopApp;

} while (stopApp == "Y");

return 0;

**RESULT OF CODING:**

....................................Menu....................................

Press Number Which operation you want!

1-Count the total number of tweets in the data set

2-Count the number of tweets that include money

3-Count the number of tweets that include politics

4-Print to the screen any tweets that include Paris

5-Print to the screen any tweets that include DreamWorks

6-Print to the screen any tweets that include Uber

7-Display tweets about Election

8-Display specific UK tweets

9-Display Breaking News tweets

10-Total number of hometown tweets

................2-Count the number of tweets that include money...........

2-Total Numbers of Tweets that include money are : 65

Do you want to continue? (Y/N) .

**NOTE: PLEASE USE CAPITAL Y or N for running the program Y stands for Y to continue & N for discontinue**

**The GitHub link is attached:** [**https://github.com/MuhammadMueezA/CODELABRESUBMISIION**](https://github.com/MuhammadMueezA/CODELABRESUBMISIION)

**If the GitHub link does not work then you can check in google drive:**

**For Code:** [**https://drive.google.com/file/d/1DHakatGt3Uo-RTxQJhjkfvZ0DqELzEuh/view?usp=sharing**](https://drive.google.com/file/d/1DHakatGt3Uo-RTxQJhjkfvZ0DqELzEuh/view?usp=sharing)

**Running file:** [**https://drive.google.com/file/d/1-E6qGtdhV-WFJ762Yje\_RbzIWEOSC1vj/view?usp=sharing**](https://drive.google.com/file/d/1-E6qGtdhV-WFJ762Yje_RbzIWEOSC1vj/view?usp=sharing)**.**

**Please click above link for running file .**