GitHub Private Repository Link:

PatelApurv1230/DS_Crafters-Capstone: Word Cloud Generator (github.com)

Team Name: DS_Crafters

➢ Group Members:

Name	Id
Apurv Patel	202301230
Faran Gori	202301209
Nauman Shethwala	202301237
Astha Patel	202304032

Project ID: P4

Word Cloud Generator:

Description:

You need to build a Word Cloud Generator. The generator takes in a dynamic collection of documents (i.e., new documents keep coming in) and creates a word cloud based on the frequency of unique words in the document. The generator starts with an initial collection of N documents. You can control the generator to only the top k most frequently used words. You should also be able to stop the generator from selecting a pre-defined set of words W.

Algorithm: Not Count Map Generate

```
void not_count_word(NotCountWord, NCP)
{
    temp
    for each entry in directory NCP
        open file
        while not end of file
            read next word into temp
            for each character  c in temp
                if  c is equal to '.' or ',' or '"'  Then
                    remove c from temp
                    continue
                EndIf
                Modify c to lowercase
            repeat
            if  NotCountWord.count(temp) is 0  Then
                 insert temp into NotCountWord with count 1
            EndIf
        repeat
    repeat
}
```

Algorithm: Convert File Type(Pdf to txt)

```
void convert_file_type(npath_)
{
    string temp
    int npath_length = length of path¬_string
    if  (npath_length – 1)nt element of npath_ is equal to 'f' then
        temp is equal to npath_
        (npath_length – 1)th element of temp = 't'
        (npath_length – 2)th element of temp = 'x'
        (npath_length – 3)th element of temp = 't'
        string command = "pdftotext " + npath_ + " " + temp
        system(command.c_str())
        npath_ = temp
    EndIf
```

Main Algorithm:

```
1. Start
2. Declare variables:
    - NCP (string)
    - initial_map (unordered_map<string, integer>)
    - NotCountWord (unordered_map<string, integer>)
    - select_ (integer)
    - count_ (integer)
    - path_ (string)
    - first_Ndocuments (integer)
```

```
    - n_documents (string)
    - num_2 (integer)
3. Display "<<--------WELCOME TO WORD CLOUD GENERATOR--------->>"
4. Display "Press 1 to not count frequency of word which is in pre-defined file"
   Display "Press 2 to not count word from particular file"
   Display "Press 3 to not count word from particular word set :-"
5. Input select_
6. Switch select_:
    case 1:
        Set NCP = "Pre-defined_Words"
        Call not_count_word(NotCountWord, NCP) function
    case 2:
        Display "<---You selected option 2--->"
        Display "Enter Not Count file Path :- "
        Input NCP
        Call convert_file_type(NCP)
        Call not_count_word(NotCountWord, NCP) function
    case 3:
        Display "<---You selected option 3--->"
        Display "Enter the Set of Word :- "
        Initialize num = 1
        while num != 2:
            Display "Please! Press 1 to add word in set or Press 2 exit:- "
            Input num
            Switch num:
                case 1:
                    Display "<---You selected option 1--->"
                    Display "Enter the word :- "
                    Input nc_word
                    Call NotCountWord.insert(pair<string, int>(nc_word, 1))
                case 2:
                    Exit loop
                default:
                    Display "Please Choose Valid Option!"
        Display "Please Choose Valid Option!"
7. Input first_Ndocuments
8. Set count_ = 0
9. while count_ != first_Ndocuments:
        Display "Enter the path of file :- "
        Input path_
        for each file in directory_iterator(path_):
            Convert file to string (convert_)
            Call convert_file_type(convert_)
            Open file as initialDocuments
            while initialDocuments is not empty:
                Read word into n_documents
                for each character c in n_documents:
                    If c is '.', ',' or '"':
                        Remove c from n_documents
                        Continue loop
                    Convert c to lowercase
                If NotCountWord.count(n_documents) == 0:
```

```
                    If initial_map.count(n_documents) == 0:
                        Insert n_documents into initial_map with frequency 1
                    Else:
                        Increment frequency of n_documents in initial_map
            Increment count_
            Display "count_ convert_ document(s) scanned successfully"
10. Initialize num_2 = 1
11. while num_2 != 4:
        Display "Press 1 to add file for counting frequency, Press 2 to add a word which
will not be counted for frequency"
        Display "Press 3 to get top k most frequently used words, Press 4 to Exit : "
        Input num_2
        Switch num_2:
            case 1:
                Input path_new
                for each file in directory_iterator(path_new):
                    Convert file to string (convert_)
                    Call convert_file_type(convert_)
                    Open file as new_Documents
                    while new_Documents is not empty:
                        Read word into new_documents
                        for each character c in new_documents:
                            If c is '.', ',' or '"':
                                Remove c from new_documents
                                Continue loop
                            Convert c to lowercase
                        If NotCountWord.count(new_documents) == 0:
                            If initial_map.count(new_documents) == 0:
                                Insert new_documents into initial_map with frequency 1
                            Else:
                                Increment frequency of new_documents in initial_map
            case 2:
                Input new_nc_word
                If initial_map.count(new_nc_word) > 0:
                    Remove new_nc_word from initial_map
                Insert new_nc_word into NotCountWord with frequency 1
            case 3:
                Sort initial_map by frequency into sorted_map
                Initialize i = 0, k
                Display "<---You selected option 3--->"
                Display "Please! Enter the value of K :- "
                Input k
                for each pair in sorted_map in reverse order:
                    Display pair.second with pair.first
                    Increment i
                    If i == k - 1:
                        Break loop
            case 4:
                Display "Thank you for Visiting , Have a Good Day !"
            default:
                Display "Please Choose Valid Option!"
12. End
```

Time Complexity (for Console Output)(Worst case):

Functions	Time Complexity	Space Complexity
For Add NotCountWord	**O(n)**	**O(n)**
For Add Word in map	**O(n)**	**O(n)**
Sort Data	**O(nlog(n))**	**O(n)**
PrintData	**O(n)**	**O(1)**
Overall	**O(nlog(n))**	**O(n)**

In Average Case: O(1)

Where n is no of words.

Choice of Data Structure:

- Unlike other data structures (arrays, linked list, stack, queue) Unordered_map has been implemented by hash table. At the average level in unordered map gives constant time **(O(1))** complexity of operations like insertion, deletion and search.
- But in worst case when value load factor becomes very high and collision occurs this unordered map gives linear time complexity.
- Although it gives linear time complexity in worst case, we choose this data structure because both common linear data structure takes linear time complexity in worst case and greater than constant (O(1)) time complexity in average case.
- Stack and queue are not work as constant (O()) time complexity in average case according to insert in sorted manner and add word according it's second part (frequency part-integer).
- We also use multimap to sort word according to its frequency.
- In this challenge we have to increase frequency of word which is same as a string which has been stored in initial_map. Thus, if a same key comes in map it increased only frequency of the original stored string and does not store again so in this case possibility of collision in map is less.

Functionalities:

- In our program three options to not count frequencies of particular words
  Choices:

  (_1_) Not count frequency of word which is in pre-defined file (Our file).

  (_2_) Not count word from User provided file.

```
 <<----------WELCOME TO WORD CLOUD GENERATOR---------->>
Press 1 to not count frequency of word which is in pre-defined file,
Press 2 to not count word from particular file and
Press 3 to not count word from particular word set :- 2

<---You selected option 2--->
Enter Not Count file Path :- E:\t4
```

  (_3_) Not count word from User provided word set in console.

- Further ask user to enter total number of initial documents and then ask user to enter the directory path of file and open each file which is in directory.
- Our code ask user to enter new file directory until the total scanned file is not equal to user provided number of initial documents.

```
Enter the total number of initial documents :- 2
-------------------------------------------------
Enter the path of file :- E:\t2
1 E:\t2\f1_.txt document scanned succesfully
2 E:\t2\f3.txt document scanned succesfully
```

- After user provided initial documents scanned then code provide more four choices
  Choices:

  (_1_) To add new file for count frequency.

```
Press 1 to add file for counting frequency,
Press 2 to add a word which will not be counted for frequency
Press 3 to get top k most frequently used words
Press 4 to Exit : 1

<---You selected option 1--->
Enter the path of file :- C:\Users\DELL\Desktop\DSA\new\DS_Crafters-Capstone\test
C:\Users\DELL\Desktop\DSA\new\DS_Crafters-Capstone\test\ok.txt document scanned succesfully
```

  (_2_) To add a new word in set which will not count in frequency.

```
Press 1 to add file for counting frequency,
Press 2 to add a word which will not be counted for frequency
Press 3 to get top k most frequently used words
Press 4 to Exit : 2

<---You selected option 2--->
Enter the new word for not counting :- Apurv
```

(_3_) To print top k most frequently used words.
(_4_) To exit from code.

Extra Functionalities:

- Our code can read PDF also. If we give the directory path of PDF files this code converts first PDF files to text files then scanned text files.

```
--------------------------------------------------
Enter the path of file :- E:\t1


MiKTeX requires Windows 10 (or greater): https://miktex.org/announcement/legacy-windows-deprecation
1 E:\t1\f1_.txt document scanned succesfully


MiKTeX requires Windows 10 (or greater): https://miktex.org/announcement/legacy-windows-deprecation
2 E:\t1\f2_.txt document scanned succesfully
```

- If in file words which contains ‘,’, ‘.’, “” at the end our code recognizes it by removing this extra character.
- In console colour input and output prints.
- If once user prints the top most frequently used words then our code instead of exit provides user to regenerate the new Output.

```
Press 1 to add file for counting frequency,
Press 2 to add a word which will not be counted for frequency
Press 3 to get top k most frequently used words
Press 4 to Exit : 3

<---You selected option 3--->
Enter the value of K :- 20
------------------------------------------------------------
|Frequency || Word                                         |
------------------------------------------------------------
|12        || non                                          |
|11        || quod                                         |
|9         || esse                                         |
|8         || cum                                          |
|7         || est                                          |
|6         || ut                                           |
|6         || ita                                          |
|6         || et                                           |
|5         || sed                                          |
|5         || si                                           |
|5         || mihi                                         |
|5         || enim                                         |
|5         || quae                                         |
|5         || ad                                           |
|4         || nihil                                        |
|4         || ipsum                                        |
|4         || autem                                        |
|4         || te                                           |
|4         || quidem                                       |
|3         || modo                                         |
------------------------------------------------------------
Press 1 to add file for counting frequency,
Press 2 to add a word which will not be counted for frequency
Press 3 to get top k most frequently used words
Press 4 to Exit :
```

Input & Output:

```
 <<----------WELCOME TO WORD CLOUD GENERATOR---------->>
Press 1 to not count frequency of word which is in pre-defined file,
Press 2 to not count word from particular file and
Press 3 to not count word from particular word set :- 2

<---You selected option 2--->
Enter Not Count file Path :- E:\t2

Enter the total number of initial documents :- 1
-----------------------------------------------
Enter the path of file :- E:\t3
1 E:\t3\sample-2mb-text-file.txt document scanned succesfully


Press 1 to add file for counting frequency,
Press 2 to add a word which will not be counted for frequency
Press 3 to get top k most frequently used words
Press 4 to Exit : 3
```

```
<---You selected option 3--->
Enter the value of K :- 25
-------------------------------------------------------------
|Frequency || Word                                          |
-------------------------------------------------------------
|7650      || sed                                           |
|6512      || in                                            |
|6224      || amet                                          |
|6173      || sit                                           |
|5258      || ut                                            |
|5247      || id                                            |
|5068      || eget                                          |
|4702      || et                                            |
|4648      || nunc                                          |
|4565      || vitae                                         |
|4413      || at                                            |
|4075      || enim                                          |
|3843      || eu                                            |
|3780      || egestas                                       |
|3706      || pellentesque                                  |
|3615      || diam                                          |
|3555      || viverra                                       |
|3532      || quis                                          |
|3510      || ac                                            |
|3392      || arcu                                          |
|3374      || non                                           |
|3284      || massa                                         |
|3282      || tellus                                        |
|3257      || nulla                                         |
|3223      || mauris                                        |
-------------------------------------------------------------
```

```
Press 1 to add file for counting frequency,
Press 2 to add a word which will not be counted for frequency
Press 3 to get top k most frequently used words
Press 4 to Exit : 2

<---You selected option 2--->
Enter the new word for not counting :- sed

Press 1 to add file for counting frequency,
Press 2 to add a word which will not be counted for frequency
Press 3 to get top k most frequently used words
Press 4 to Exit : 2

<---You selected option 2--->
Enter the new word for not counting :- in

Press 1 to add file for counting frequency,
Press 2 to add a word which will not be counted for frequency
Press 3 to get top k most frequently used words
Press 4 to Exit : 3
```

```
<---You selected option 3--->
Enter the value of K :- 25
------------------------------------------------------------
|Frequency || Word                                         |
------------------------------------------------------------
|6224      || amet                                         |
|6173      || sit                                          |
|5258      || ut                                           |
|5247      || id                                           |
|5068      || eget                                         |
|4702      || et                                           |
|4648      || nunc                                         |
|4565      || vitae                                        |
|4413      || at                                           |
|4075      || enim                                         |
|3843      || eu                                           |
|3780      || egestas                                      |
|3706      || pellentesque                                 |
|3615      || diam                                         |
|3555      || viverra                                      |
|3532      || quis                                         |
|3510      || ac                                           |
|3392      || arcu                                         |
|3374      || non                                          |
|3284      || massa                                        |
|3282      || tellus                                       |
|3257      || nulla                                        |
|3223      || mauris                                       |
|3181      || aliquam                                      |
|3102      || tincidunt                                    |
------------------------------------------------------------
```

```
Do you want to save file to output
Press 1 Yes
Press 2 No
1
```

➔ "Output.txt" uploaded on GitHub.

2)

```
 <<----------WELCOME TO WORD CLOUD GENERATOR---------->>
Press 1 to not count frequency of word which is in pre-defined file,
Press 2 to not count word from particular file and
Press 3 to not count word from particular word set :- 3

<---You selected option 3--->
Enter the Set of Word :-
Press 1 to add word in set or
Press 2 exit :- 1

<---You selected option 1--->
Enter the word :- hi

Press 1 to add word in set or
Press 2 exit :- 1

<---You selected option 1--->
Enter the word :- is

Press 1 to add word in set or
Press 2 exit :- 2


Enter the total number of initial documents :- 1
-------------------------------------------------
Enter the path of file :- E:\t1


MiKTeX requires Windows 10 (or greater): https://miktex.org/announcement/legacy-windows-deprecation
1 E:\t1\f1_.txt document scanned succesfully
```

```
 Press 1 to add file for counting frequency,
 Press 2 to add a word which will not be counted for frequency
 Press 3 to get top k most frequently used words
 Press 4 to Exit : 1

 <---You selected option 1--->
 Enter the path of file :- E:\t1_
 E:\t1_\OK2.txt document scanned succesfully

 Press 1 to add file for counting frequency,
 Press 2 to add a word which will not be counted for frequency
 Press 3 to get top k most frequently used words
 Press 4 to Exit : 3

 <---You selected option 3--->
 Enter the value of K :- 8
 ------------------------------------------------------------
 |Frequency || Word                                         |
 ------------------------------------------------------------
 |189       || the                                          |
 |106       || to                                           |
 |95        || was                                          |
 |86        || and                                          |
 |85        || she                                          |
 |76        || it                                           |
 |75        || a                                            |
 |74        || that                                         |
 ------------------------------------------------------------
```

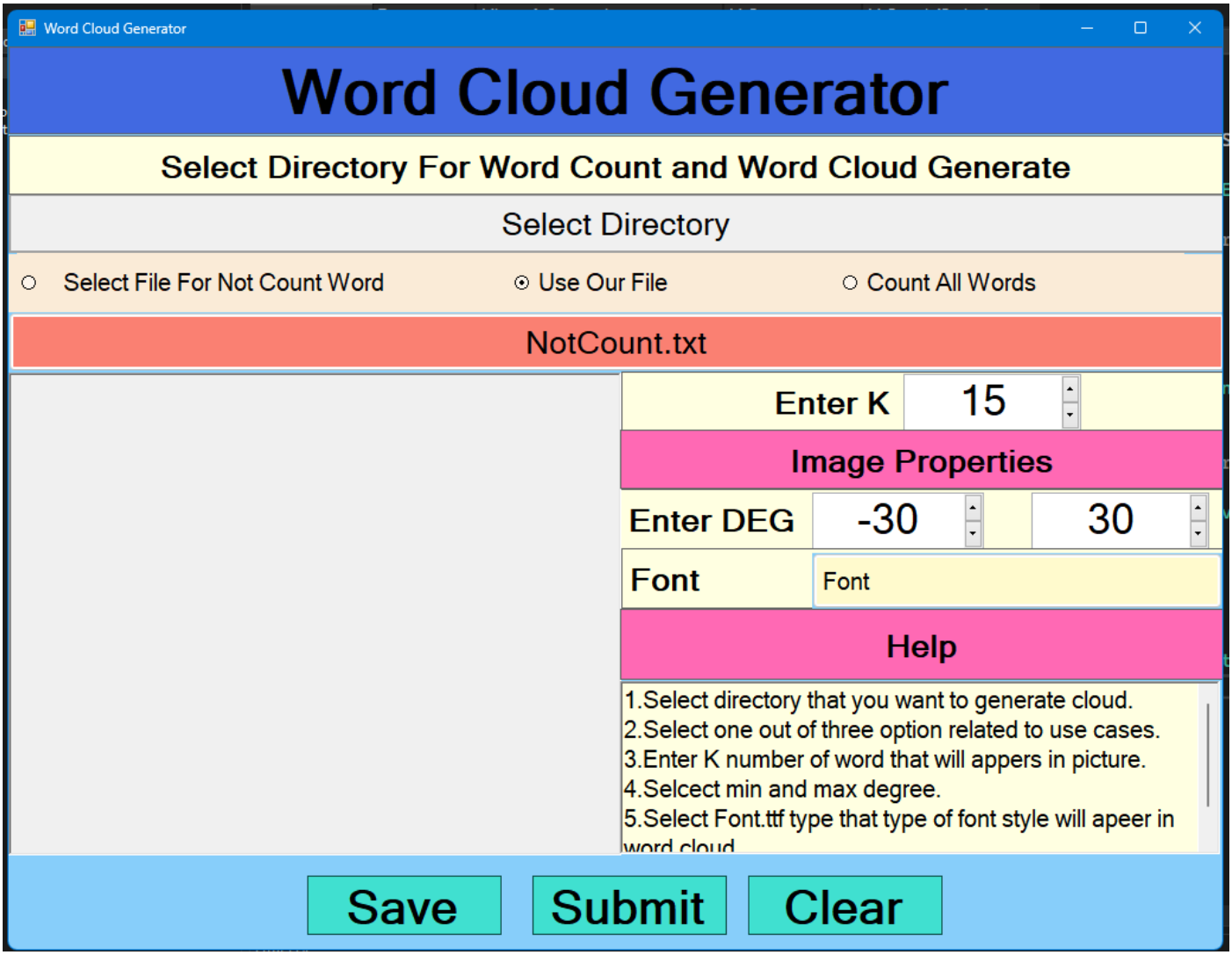
```
Press 1 to add file for counting frequency,
Press 2 to add a word which will not be counted for frequency
Press 3 to get top k most frequently used words
Press 4 to Exit : 1

<---You selected option 1--->
Enter the path of file :- E:\t2
E:\t2\sample-text-file (1).txt document scanned succesfully

Press 1 to add file for counting frequency,
Press 2 to add a word which will not be counted for frequency
Press 3 to get top k most frequently used words
Press 4 to Exit : 3

<---You selected option 3--->
Enter the value of K :- 10
------------------------------------------------------------
|Frequency || Word                                         |
------------------------------------------------------------
|189       || the                                          |
|106       || to                                           |
|95        || was                                          |
|86        || and                                          |
|85        || she                                          |
|76        || it                                           |
|75        || a                                            |
|74        || that                                         |
|63        || of                                           |
|63        || he                                           |
------------------------------------------------------------
```

This is Our GUI:



1.In GUI we have to select directory that will count frequency

2.For Not Count word or pre-defined words we have three options "Select File" "Use Our File" "Count All Words"

3.Select no of K Word that will show in textbox and generate word cloud using CPP SFML library.

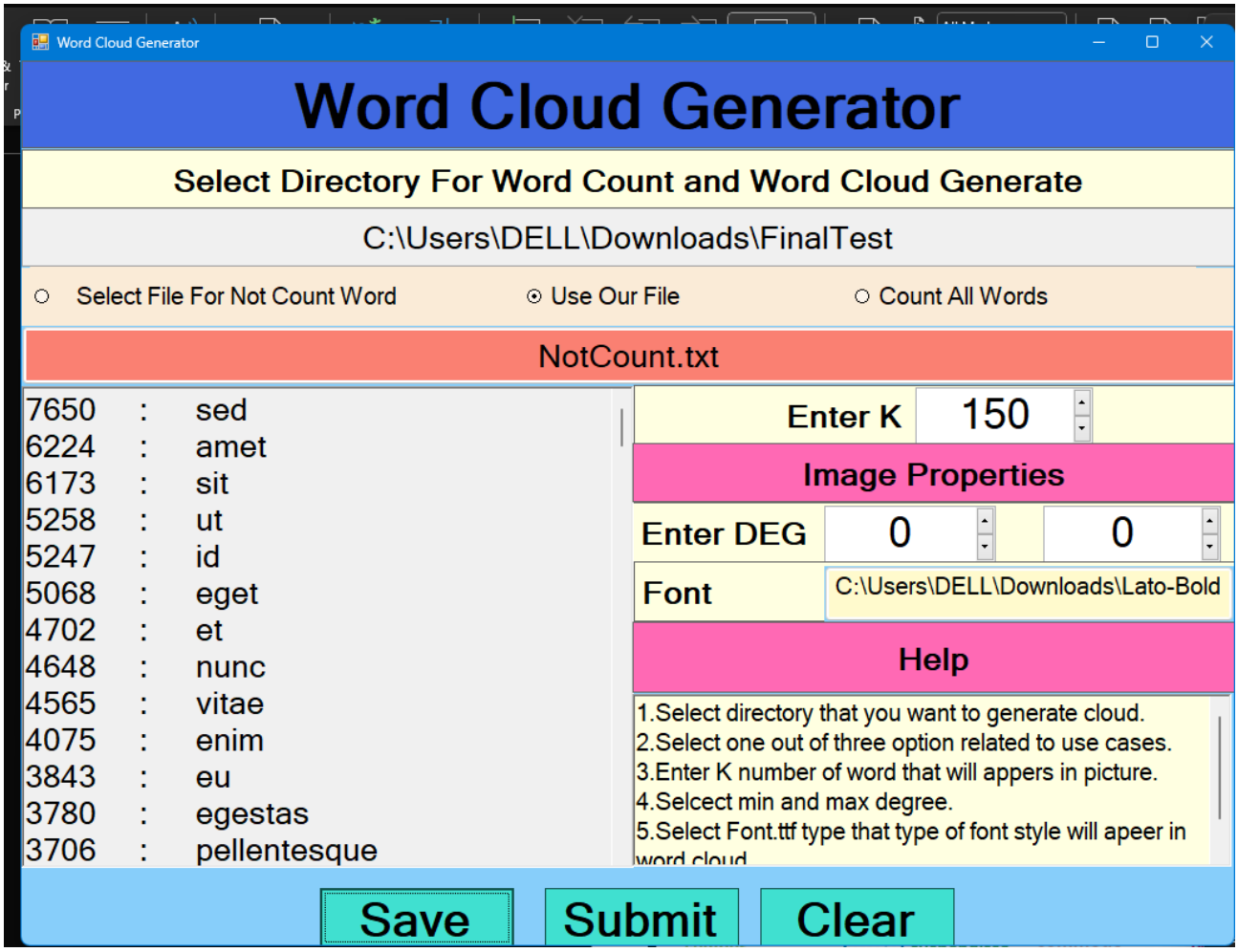
4.For word cloud generate we have 2 properties that can be user define degree of each text in image.we have to give min_max degree in NummericUpDown Box.

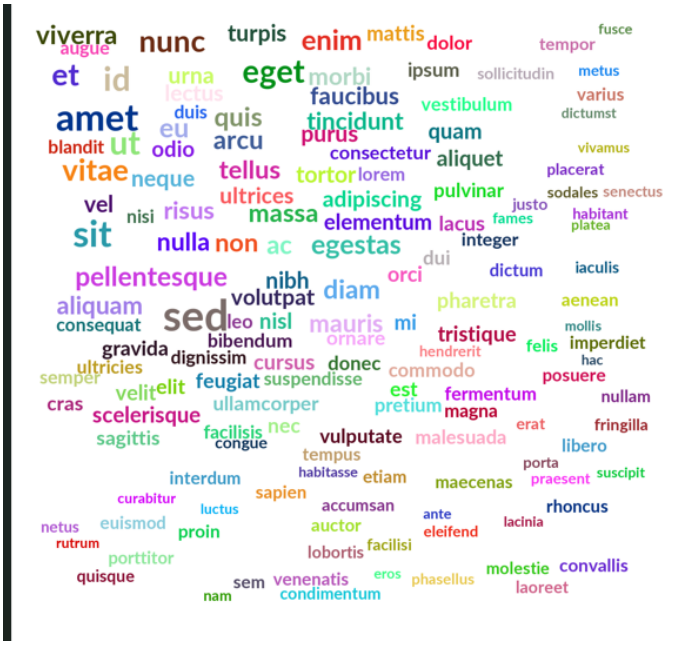
5.Submit when you press that image will open and top k word print in textbox.

6.Save For Save Output in .txt file.

➢ Output:

For Sample input we use file from this website:
https://www.learningcontainer.com/sample-text-file/





For 0 to 0 degree

For -45 to 45 degree

Different Fonts

Reference

- Text File reference link : https://filesamples.com/formats/txt
- ChatGPT 3.5
- Wikipedia: Simple and Fast Multimedia Library - Wikipedia
- Stack Overflow: Stack Overflow - Where Developers Learn, Share, & Build Careers