EXPERIMENT NO 4

A. CALCULATE WORD COUNT OF A GIVEN SPECIFIC DOCUMENT AND SHOW TOP 10 FREQUENT WORDS WITH THEIR FREQUENCY

IMPORTING LIBRARIES

In [1]:
```
import collections
import pandas as pd
import matplotlib.pyplot as plt
import nltk as ntk
%matplotlib inline
```

READING THE DOCUMENT FILE

In [2]:
```
# Read input file, note the encoding is specified here
# It may be different in your text file
file = open('/DOCUMENTS/COLLEGE/CLASSES/EXPERIMENT_NO_4/EXPERIMENT4.txt',
                    encoding="utf8")
a = file.read()
```

In [3]:
```
# Stopwords
stopwords = set(line.strip() for line in ntk.corpus.stopwords.words('english'))
```

CALCULATE THE WORD COUNTING

In [4]:
```
# Instantiate a dictionary, and for every word in the file,
# Add to the dictionary if it doesn't exist. If it does, increase the count.
wordcount = {}
# To eliminate duplicates, splitting by punctuation, and use case demiliters.
for word in a.lower().split():
    word = word.replace(".", "")
    word = word.replace(",", "")
    word = word.replace("[", "")
    word = word.replace("]", "")
    word = word.replace("!", "")
    word = word.replace("*", "")
    if word not in stopwords:
        if word not in wordcount:
            wordcount[word] = 1
        else:
            wordcount[word] += 1
```

TOP 'N' frequently used words

In [5]:
```
# Print most common word
n_print = int(input("How many most common words to print: "))
print("\nOK. The {} most common words are as follows\n".format(n_print))
word_counter = collections.Counter(wordcount)
for word, count in word_counter.most_common(n_print):
    print(word, ": ", count)
# Close the file
file.close()
# Create a data frame of the most common words
lst = word_counter.most_common(n_print)
df = pd.DataFrame(lst, columns=['Word', 'Count'])
df.head()
```

```
How many most common words to print: 10

OK. The 10 most common words are as follows

sed :  7650
amet :  6224
sit :  6173
ut :  5258
id :  5247
eget :  5068
et :  4702
nunc :  4648
vitae :  4565
enim :  4075
```

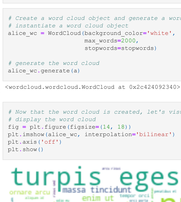
Out[5]:

	Word	Count
0	sed	7650
1	amet	6224
2	sit	6173
3	ut	5258
4	id	5247

VISUALIZING THE DATASET

In [6]:
```
# Draw a bar chart
df.plot.bar(x='Word', y='Count')
```

Out[6]:
```
<AxesSubplot:xlabel='Word'>
```



B. CREATING A WORD CLOUD

INSTALLING THE WORD CLOUD

In [7]:
```
# install wordcloud
from wordcloud import WordCloud, STOPWORDS
from PIL import Image
!pip install wordcloud

# import package and its set of stopwords

print('Wordcloud is installed and imported!')
```

```
Requirement already satisfied: wordcloud in c:\users\hp\anaconda3\lib\site-packages (1.8.1)
Requirement already satisfied: numpy>=1.6.1 in c:\users\hp\anaconda3\lib\site-packages (from wordcloud) (1.20.3)
Requirement already satisfied: matplotlib in c:\users\hp\anaconda3\lib\site-packages (from wordcloud) (3.4.3)
Requirement already satisfied: pillow in c:\users\hp\anaconda3\lib\site-packages (from wordcloud) (8.4.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib->wordcloud) (1.3.1)
Requirement already satisfied: cycler>=0.10 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib->wordcloud) (0.10.0)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib->wordcloud) (2.8.2)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib->wordcloud) (3.0.4)
Requirement already satisfied: six in c:\users\hp\anaconda3\lib\site-packages (from cycler>=0.10->matplotlib->wordcloud) (1.16.0)
Wordcloud is installed and imported!
```

READING THE DOCUMENT FILE

In [8]:
```
# Read input file, note the encoding is specified here
# It may be different in your text file
file = open('/DOCUMENTS/COLLEGE/CLASSES/EXPERIMENT_NO_4/EXPERIMENT4.txt',
            encoding="utf8")
a = file.read()
```

CREATING THE WORD CLOUD

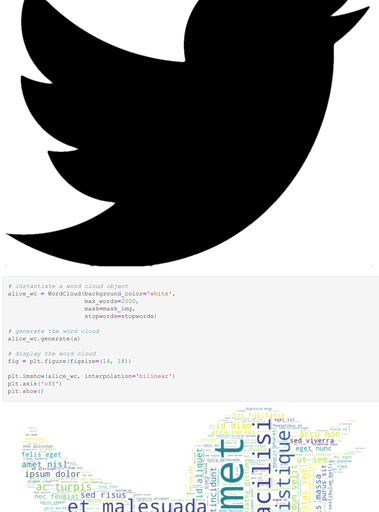
In [9]:
```
# use the function set to remove any redundant stopwords.
stopwords = set(STOPWORDS)
```

In [10]:
```
# Create a word cloud object and generate a word cloud using only the first 2000 words.
# instantiate a word cloud object
alice_wc = WordCloud(background_color='white',
                     max_words=2000,
                     stopwords=stopwords)

# generate the word cloud
alice_wc.generate(a)
```

Out[10]:
```
<wordcloud.wordcloud.WordCloud at 0x2c424092340>
```

In [11]:
```
# Now that the word cloud is created, let's visualize it.
# display the word cloud
fig = plt.figure(figsize=(14, 18))
plt.imshow(alice_wc, interpolation='bilinear')
plt.axis('off')
plt.show()
```



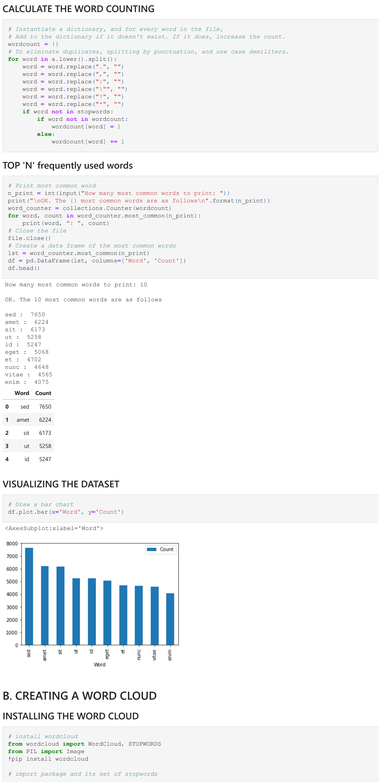
CREATING WORD CLOUD OVER THE MASK

In [12]:
```
import urllib
```

In [13]:
```
# masked image used
mask_img = np.array(
    Image.open('/DOCUMENTS/COLLEGE/CLASSES/EXPERIMENT_NO_4/twitter_mask.png'))

fig = plt.figure(figsize=(14, 18))

plt.imshow(mask_img, cmap=plt.cm.gray, interpolation='bilinear')
plt.axis('off')
plt.show()
```

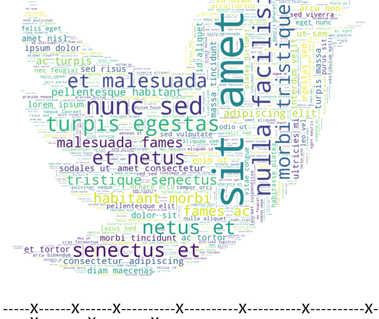


In [14]:
```
# instantiate a word cloud object
alice_wc = WordCloud(background_color='white',
                     max_words=2000,
                     mask=mask_img,
                     stopwords=stopwords)

# generate the word cloud
alice_wc.generate(a)

# display the word cloud
fig = plt.figure(figsize=(14, 18))

plt.imshow(alice_wc, interpolation='bilinear')
plt.axis('off')
plt.show()
```



-----X-----X-----X----------X----------X----------X----------X----------X----------X----------X-----