Assignment 8

1. WAP to scrape any web-site. Print the top 5 repeated words and count them and plot the graph of it.

->

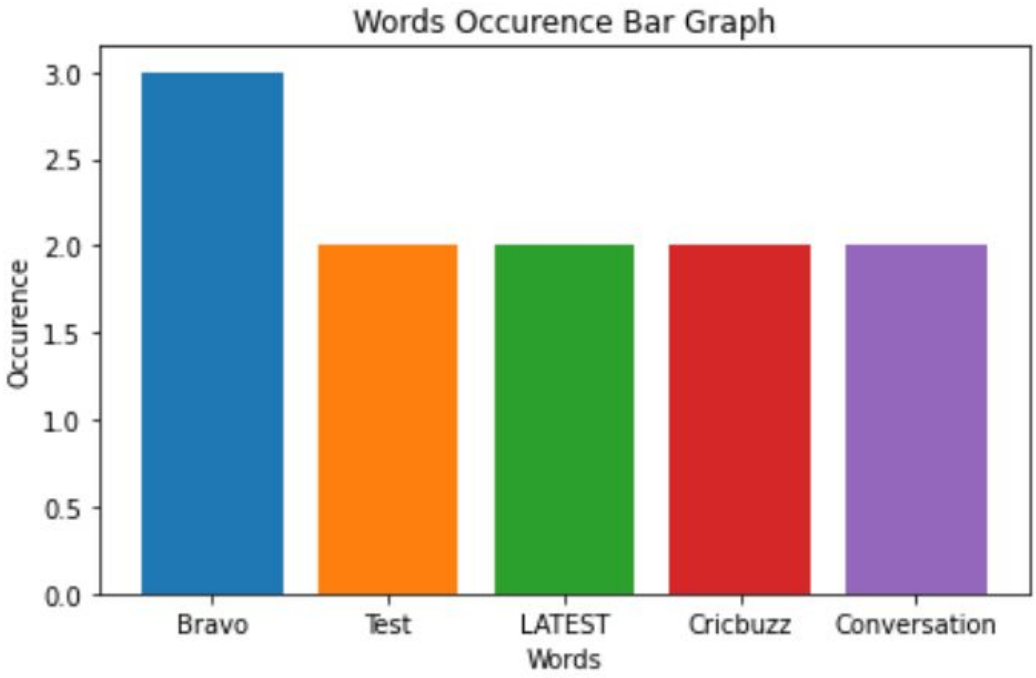
Code:

```
import requests
from bs4 import BeautifulSoup
from collections import Counter
import matplotlib.pyplot as plt

page = requests.get('https://www.cricbuzz.com/')
page = page.text

soup = BeautifulSoup(page, 'html.parser')
words = ''
words += soup.title.text + ' '

h4 = soup.find_all('h4')
h3 = soup.find_all('h3')
h2 = soup.find_all('h2')
for i, j, k in zip(h4, h3, h2):
  print(i.text, j.text, k.text)
  words += i.text + ' ' + j.text + ' ' + k.text + ' '

list1 =
['to','-',':','and','&','1','2','3','All','tags','In','With','This','The',
'his','her','or','What','do','was','I','It','as','a','the','that','but','w
ho','by','when','An','an','can','with','is','was','for','on','of','upto','
from','in','And','A','','lot','`','!','@','#','$','%','^','*','(',')','my'
,'|']

word_list = words.split(' ')
for i in word_list:
  if i in list1:
    word_list.remove(i)
```

```
coun = Counter(word_list)

for word, count in coun.most_common(5):
  print(f'{word} : {count}')
  plt.bar(word, count)
  plt.xlabel('Words')
  plt.ylabel('Occurence')
  plt.title('Words Occurence Bar Graph')
  plt.tight_layout()
```

Output:

```
Bravo : 3
Test : 2
LATEST : 2
Cricbuzz : 2
Conversation : 2
```



2. WAP to do a sentiment analysis of any word entered by the user in voice command.

->

Code:

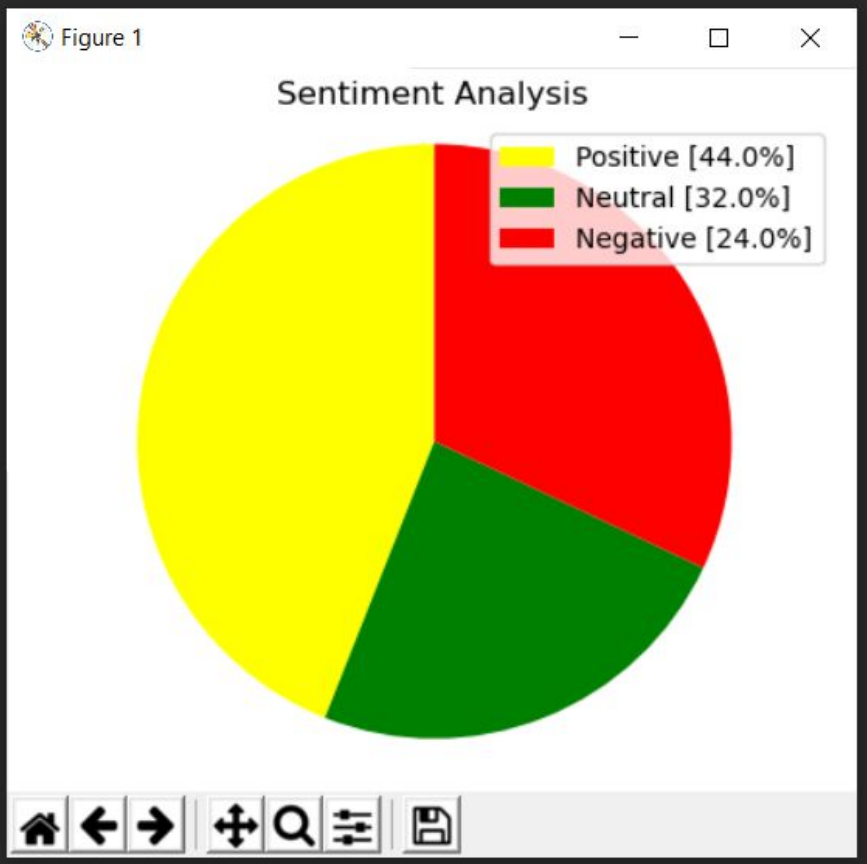
```
import speech_recognition as s_r
import tweepy
from textblob import TextBlob
import time
import matplotlib.pyplot as plt

recognise_voice = s_r.Recognizer()
mic = s_r.Microphone(device_index=1)
with mic as source:
    print("Speak now 🔊 to perform the Sentiment Analysis on Twitter!!!!")
    print('Tell us the Keywords/Tags/Person 🔊')
    audio = recognise_voice.listen(source)
    time.sleep(2)
    print('Tell us the number of tweets for Sentiment Analysing 🔊')
    audio2 = recognise_voice.listen(source)

word = recognise_voice.recognize_google(audio)
total = recognise_voice.recognize_google(audio2)
print(f'{total} tweets will be use for Sentiment Analysis of {word}')

def percentage(part, whole):
    return round(100 * float(part)/float(whole), 2)

consumer_key = 'COfg63NIGgNFeVZDIEA8imE5C'
consumer_secret = 'Dx96nwyY6Wy8l3adPfvFRjS81Ie672eNAjUXsBaQCowz4NLV4h'
access_token = '717706611821654017-0aAzSitKPXu6fNrC6Y8S7qYjP2eTMDi'
access_token_secret = 'x9cT3aYsz4OGgFOJzmN0oI9Q7EE4T5bQ7oSOD0FvNMHT8'

auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
api = tweepy.API(auth)
public_tweets = tweepy.Cursor(api.search, q = word).items(int(total))
```

```
polarity = 0
positive = 0
negative = 0
neutral = 0

for tweet in public_tweets:
    #print(tweet.text)
    analysis = TextBlob(tweet.text)
    #print(analysis.sentiment)
    polarity += analysis.sentiment.polarity

    if analysis.sentiment.polarity == 0:
        neutral += 1
    elif analysis.sentiment.polarity < 0.00:
        negative += 1
    elif analysis.sentiment.polarity > 0.00:
        positive += 1

positive = percentage(positive, int(total))
negative = percentage(negative, int(total))
neutral = percentage(neutral, int(total))
polarity = percentage(polarity, int(total))

check_max = [positive, negative, neutral]
maxi_index = check_max.index(max(check_max))
print(f'After analyzing {total} tweets, Reaction of peoples about {word}
is: ')
if maxi_index == 0:
    print('Positive')
elif maxi_index == 1:
    print('Negative')
else:
    print('Neutral')

labels = ['Positive ['+str(positive)+'%]', 'Neutral ['+str(neutral)+'%]',
'Negative ['+str(negative)+'%]']
sizes = [positive, negative, neutral]
```

```
colors = ['yellow', 'green', 'red']
patches, texts = plt.pie(sizes, colors=colors, startangle = 90)
plt.legend(patches, labels, loc='best')
plt.title('Sentiment Analysis')
plt.axis('equal')
plt.tight_layout()
plt.show()
```

Output:

```
Speak now 🔊 to perform the Sentiment Analysis on Twitter!!!!
Tell us the Keywords/Tags/Person 🔊
Tell us the number of tweets for Sentiment Analysing 🔊
100 tweets will be use for Sentiment Analysis of Virat Kohli
After analyzing 100 tweets, Reaction of peoples about Virat Kohli is:
Positive
```



3. WAP to pin different locations of Fruit Farms on the map and plot a pie chart according to states.

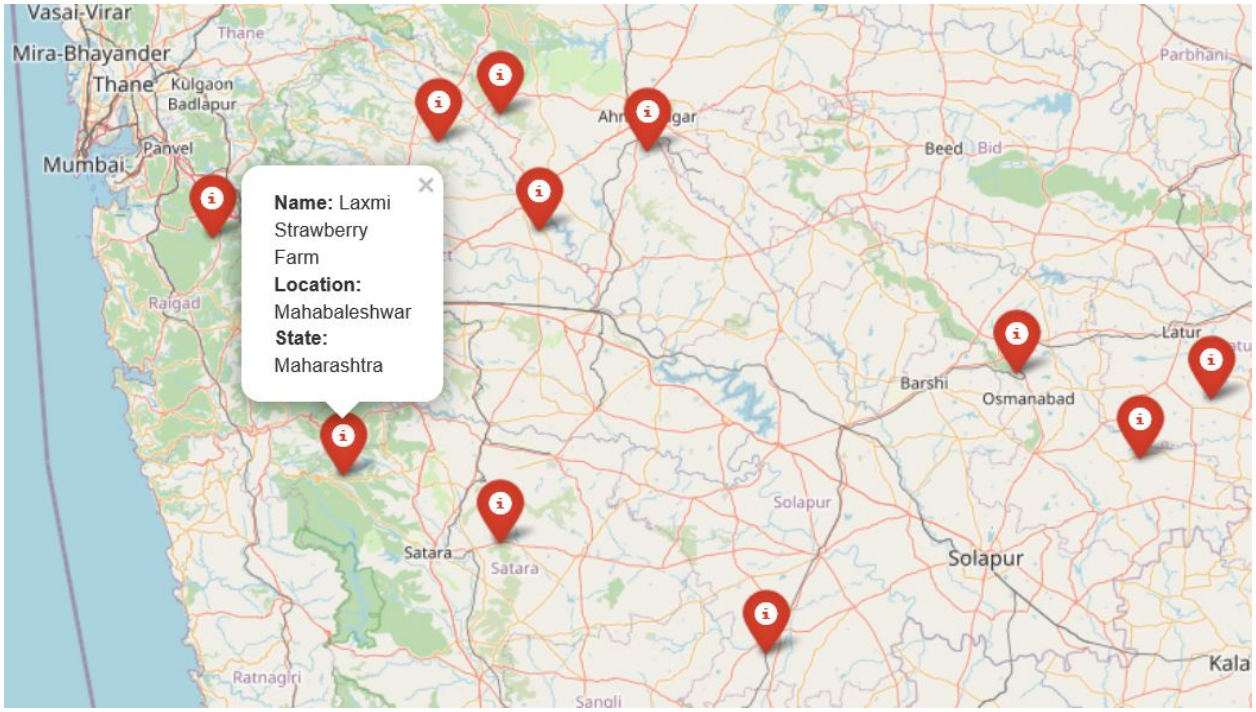
->

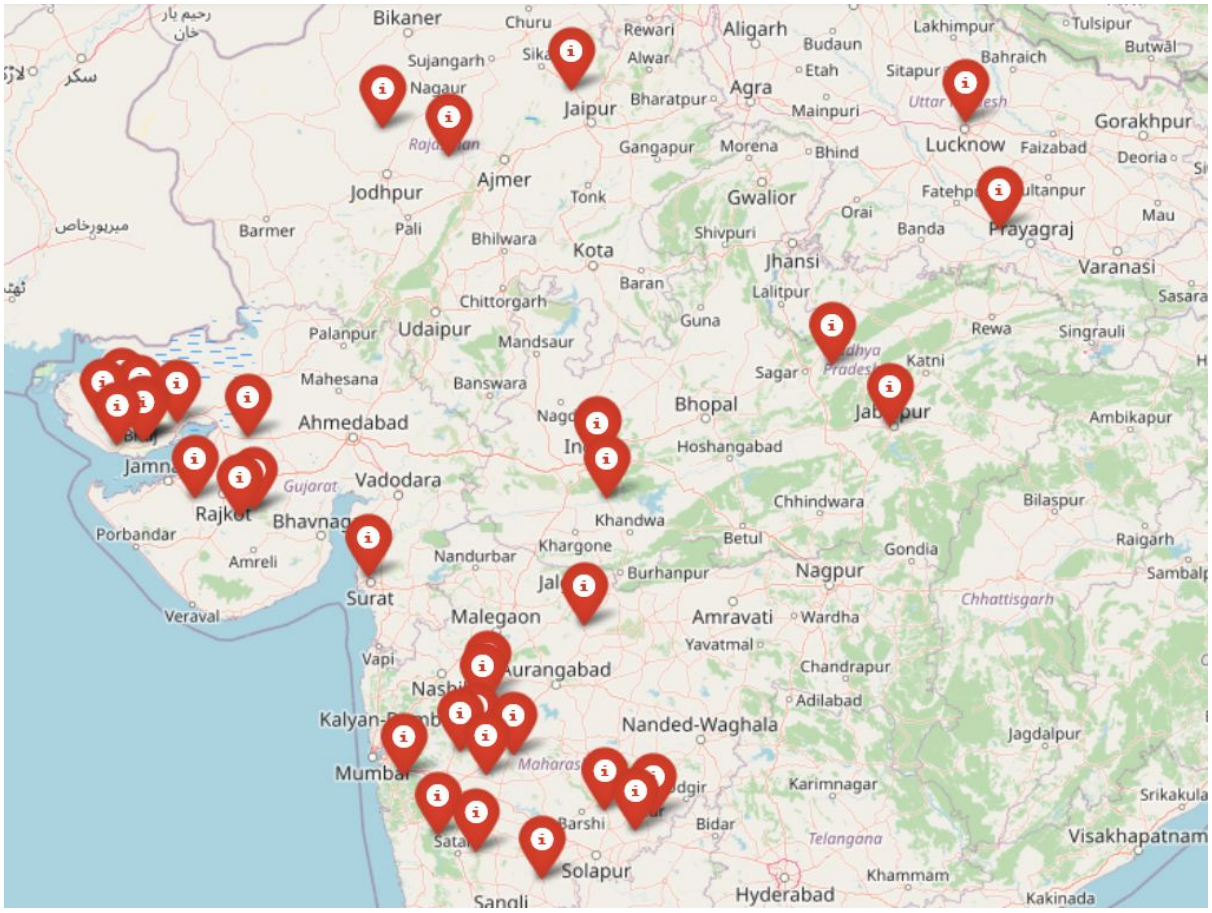
Code:

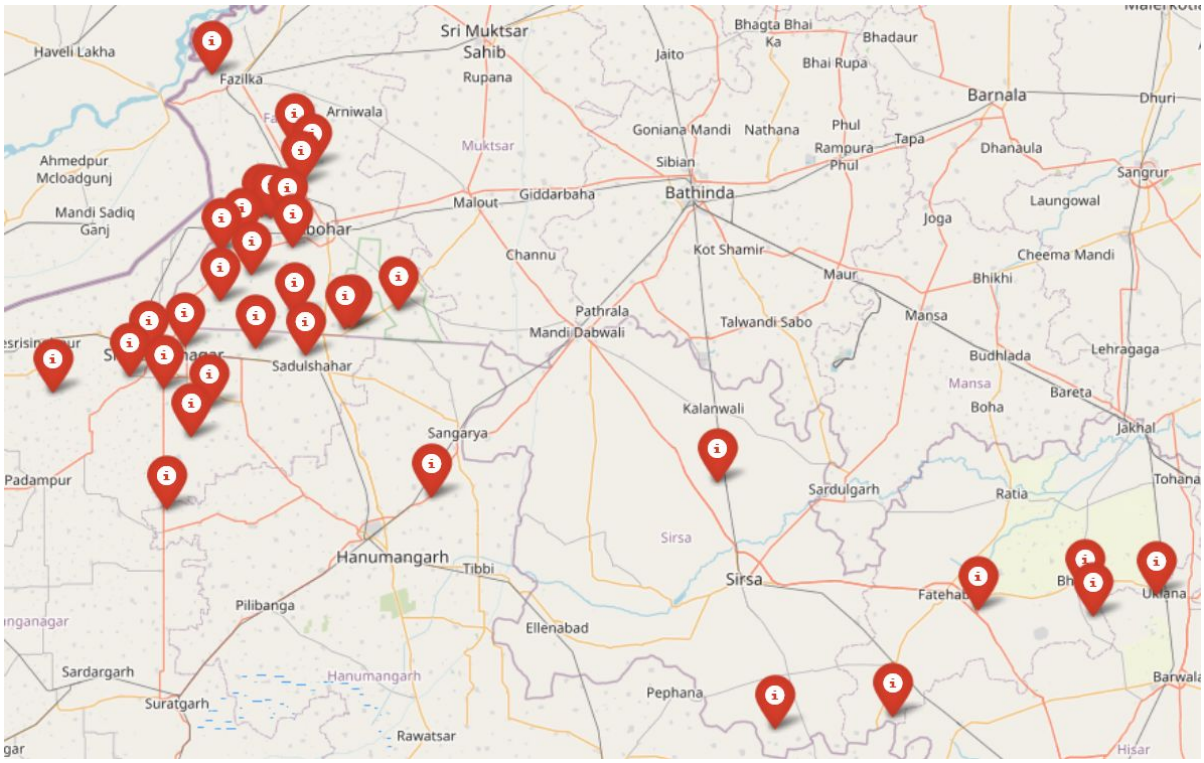
```
import folium
import pandas as pd

india_map = folium.Map(location = [20.593683, 78.962883], zoom_start = 5)

df = pd.read_csv('/content/Fruit_Farming_Dataset.csv')
lat = list(df['Latitude'])
lon = list(df['Longitude'])
name = list(df['Name'])
state = list(df['State'])
loca = list(df['Location'])

for lt, ln, nm, st, lo in zip(lat, lon, name, state, loca):
  folium.Marker(location=[lt, ln], icon =
folium.Icon(color='red'),popup='<b>Name: </b>' + nm + '<br> <b>Location:
</b>' + lo + '<br> <b>State: </b>'+ st).add_to(india_map)

india_map

def per(part, whole):
  return round(100 * float(part) / float(whole), 2)

state_coun = Counter(state)
labels = []
size = []
label2 = []
state_list = ['Gujarat', 'Punjab', 'Rajasthan', 'Maharashtra', 'Haryana',
'Madhya-Pradesh', 'Uttar-Pradesh']

for state in state_list:
  res = per(state_coun[state], len(name))
  size.append(res)
```

```
  labels.append(state)
  label2.extend([state + ' : ' + str(res) + '%'])

print('State-wise Fruit Farms in India are as follows: ')
for i in label2:
  print(i)
print()

color = ['yellow', 'olive', 'orangered', 'palegreen', 'grey', 'cyan',
'pink']
patches, texts = plt.pie(size, startangle = 90, colors=color)
plt.pie(size, labels=labels, colors=color, radius=1.4, autopct='%1.1f%%',
pctdistance=0.8,startangle=90, textprops={'fontsize':12})
plt.axis('equal')
plt.legend(patches, labels, loc='upper left')
plt.title('Fruit Farms in India', y=1.08, fontsize=20)
plt.tight_layout()
plt.show()
```

Output:




Bikaner
Churu
Rewari
Aligarh
Lakhimpur
Tulsipur
Sujangarh
Alwar
Budaun
Bahraich
Nagaur
Jaipur
Agra
Etah
Sitapur
Bharatpur
Mainpuri
Gorakhpur
Lucknow
Faizabad
Gangapur
Morena
Bhind
Deoria
Jodhpur
Ajmer
Tonk
Gwalior
Fatehpur
Barmer
Pali
Bhilwara
Kota
Orai
Banda
Prayagraj
Mau
Shivpuri
Jhansi
Varanasi
Baran
Lalitpur
Chittorgarh
Guna
Sasaram
Udaipur
Palanpur
Mandsor
Rewa
Singrauli
Sagar
Katni
Mahesana
Banswara
Bhopal
Ambikapur
Ahmedabad
Hoshangabad
Vadodara
Gujarat
Chhindwara
Bilaspur
Rajkot
Khandwa
Porbandar
Bhavnagar
Betul
Raigarh
Nandurbar
Khargone
Gondia
Amreli
Burhanpur
Nagpur
Surat
Chhattisgarh
Veraval
Malegaon
Amravati
Wardha
Yavatmal
Vapi
Aurangabad
Chandrapur
Adilabad
Nanded-Waghala
Jagdalpur
Mumbai
Karimnagar
Srikakulam
Barshi
Bidar
Telangana
Visakhapatnam
Solapur
Khammam
Hyderabad
Kakinada


Haveli Lakha
Sri Muktsar Sahib
Bhagta Bhai Ka
Bhadaur
Jaito
Bhai Rupa
Fazilka
Rupana
Barnala
Dhuri
Arniwala
Goniana Mandi
Nathana
Phul
Tapa
Muktsar
Rampura Phul
Dhanaula
Ahmedpur Mcloadgunj
Sibian
Sangrur
Giddarbaha
Bathinda
Laungowal
Mandi Sadiq Ganj
Malout
Joga
Channu
Kot Shamir
Cheema Mandi
Maur
Bhikhi
Pathrala
Talwandi Sabo
Mansa
Mandi Dabwali
Budhlada
Lehragaga
Sadulshahar
Mansa
Bareta
Kalanwali
Boha
Jakhal
Sangarya
Padampur
Sardulgarh
Ratia
Tohana
Sirsa
Hanumangarh
Tibbi
Sirsa
Pilibanga
Fatehabad
Ellenabad
Sardargarh
Hanumangarh
Pephana
Barwala
Suratgarh
Rawatsar
Hisar

```
State-wise Fruit Farms in India are as follows:
Gujarat : 17.81%
Punjab : 21.92%
Rajasthan : 20.55%
Maharashtra : 19.18%
Haryana : 10.96%
Madhya-Pradesh : 6.85%
Uttar-Pradesh : 2.74%
```