```
import nltk
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('vader_lexicon')

from django.shortcuts import render
from django.http import HttpResponse
import string
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from nltk.tokenize import word_tokenize
from collections import Counter
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import plotly.graph_objects as go

#Introduction function
def intro(request):
   return render(request,'sentiment/intro.html')

#Analyse function
def analyse(request):
   return render(request, "sentiment/analyse.html")

#Display the questions.
def ans(request):
      answer_questions = request.GET.get('answer_questions', 'off')
      if answer_questions == 'on':
          qq = {'e': 'How are you feeling these days ?', 'a': 'Confident.', 'b': 'Really awful.', 'c': 'Very excited.',
              'd': 'Too Lazy.', 'f': 'How is your lockdown going on ?', 'g': 'It is too frustating.',
              'h': 'Very peaceful.', 'i': 'Motivated and focused.', 'j': 'Bored of staying at home.',
              'k': 'How do you spend your free time ?', 'l': 'Having fun with my freinds.', 'm': 'Enjoy reading books.',
              'n': 'Like watching movies and series.', 'o': 'Playing sports.',
              'p': 'Do you have trouble sleeping at night ?',
              'q': "I often don't sleep at night.", 'r': ' Very rare.', 's': 'Not at all', 't': '2-3 days in a week.',
              'u': 'If you are assigned a project, how will you deal with it ?', 'v': 'I focus too much on the details.',
              'w': 'I have trouble asking for help.', 'x': 'I have trouble saying No.',
              'y': 'I become impatient when projects run beyond the deadline.'}
          return render(request, "sentiment/answer_questions.html", qq)
      else :
         return HttpResponse('<h1> Toggle button should be on. </h1>')

#Analysing input function
def final_output(request):
   user_input= request.GET.get('user_input', 'off')


   if user_input== 'on' :
      text = open('sentiment/output1.txt', 'w')
      print("Enter your comment")
      inp = request.GET.get('text', 'default')
      print('*** The sentiment of your input ***\n');
      text.write(inp)
      text.close()
```

```
text = open('sentiment/output1.txt', 'r').read()
lower_case = text.lower()

cleaned_text = lower_case.translate(str.maketrans(" ", " ", string.punctuation))

tokenized_words = word_tokenize(cleaned_text, "english")

final_words = []
for word in tokenized_words:
    if word not in stopwords.words('english'):
        final_words.append(word)
    lemma_words = []

for word in final_words:
    word = WordNetLemmatizer().lemmatize(word)
    lemma_words.append(word)
emotion_list = []

file = open("sentiment/emotions.txt", "r")
for line in file:
    clear_line = line.replace("\n", '').replace(",", '').replace("'", '').strip()
    word, emotion = clear_line.split(':')
    if word in lemma_words:
        emotion_list.append(emotion)
file.close()

w = Counter(emotion_list)

def sentiment_analyse(sentiment_text):
    score = SentimentIntensityAnalyzer().polarity_scores(sentiment_text)
    if score['neg'] > score['pos']:
        mes="No matter what you are going through , there's always a  light at the end of the tunnel."
        au=" -DEMI LOVATO "
        return("NEGATIVE  SENTIMENT", mes, au)
    elif score['neg'] < score['pos']:

        mes="Never bend your head. Always hold it high. Look the world straight in the eye."
        au = "- HELEN KELLER "
        return("POSITIVE  SENTIMENT", mes,au)

    else:
       mes=" Life has got all those TWIST and TURNS. You've got to hold on tight and off you go."
       au=" -NICOLE KIDMAN"
       return("NEUTRAL  SENTIMENT",mes,au)


p,m,a= sentiment_analyse(cleaned_text)
ll=list(w.keys())
vv=list(w.values())
marker = dict(color="red")
data = [go.Bar(x=ll, y=vv, name="POLARITY SCORE")]
layout = go.Layout(title="POLARITY SCORE")
fig = go.Figure(data=data, layout=layout)
pp=fig.write_image("sentiment/static/sentiment/image1.jpeg")
```

```
        d={'value': p , 'message': m ,'author':a}

        return render(request, "sentiment/final_output.html", d)
    else :
        return HttpResponse('<h1> Toggle button should be on. </h1>')


#Analysis of quiz
def get_ans(request):
    aa = request.GET.get('text1', 'default')
    aa.replace('+'," ")
    ff, mm, ii, kk, gg = aa.split()
    dic = {'a': 'Confident', 'b': 'Really awful', 'c': 'Very excited.', 'd': 'Too Lazy'}
    ee = dic[ff]
    dic1 = {'a': 'It is too frustating.', 'b': 'Very peaceful.', 'c': 'Motivated and focused.',
            'd': 'Bored of staying at home.'}
    hh = dic1[gg]
    dic2 = {'a': 'Having fun with my freinds.', 'b': 'Enjoy reading books.',
            'c': 'Like watching movies and series.', 'd': 'Playing sports.'}
    jj = dic2[ii]
    dic3 = {'a': 'I often don,t sleep at night.', 'b': 'Very rare.', 'c': 'Not at all', 'd': '2-3 days in a week.'}
    ll = dic3[kk]
    dic4 = {'a': 'I focus too much on the details.', 'b': 'I have trouble asking for help.',
            'c': 'I have trouble saying No.', 'd': 'I become impatient when projects run beyond the deadline.'}
    nn = dic4[mm]

    inp = ee + " " + nn + " " + jj + " " + ll + " " + hh
    print(inp)
    lower_case = inp.lower()
    print(lower_case)
    cleaned_text = lower_case.translate(str.maketrans(" ", " ", string.punctuation))
    print(cleaned_text)
    tokenized_words = word_tokenize(cleaned_text, "english")

    final_words = []
    for word in tokenized_words:
        if word not in stopwords.words('english'):
            final_words.append(word)
        lemma_words = []

    for word in final_words:
        word = WordNetLemmatizer().lemmatize(word)
        lemma_words.append(word)
    emotion_list = []

    file = open("sentiment/emotions.txt", "r")
    for line in file:
        clear_line = line.replace("\n", '').replace(",", '').replace("'", '').strip()
        word, emotion = clear_line.split(':')
        if word in lemma_words:
            emotion_list.append(emotion)
    file.close()

    w = Counter(emotion_list)
```

```
    def sentiment_analyse(sentiment_text):
        score = SentimentIntensityAnalyzer().polarity_scores(sentiment_text)
        print(score)
        if score['neg'] > score['pos']:
            mes = "No matter what you are going through , there's always a  light at the end of the tunnel."
            au = " -DEMI LOVATO "
            return ("NEGATIVE  SENTIMENT",mes,au)
        elif score['neg'] < score['pos']:
            mes = "Never bend your head. Always hold it high. Look the world straight in the eye."
            au = "- HELEN KELLER "
            return ("POSITIVE SENTIMENT",mes,au)
        elif score['neg']==score['pos']:
            mes = " Life has got all those TWIST and TURNS. You've got to hold on tight and off you go."
            au = " -NICOLE KIDMAN"
            return ("NEUTRAL  SENTIMENT",mes,au)

    aaa,mmm,aaaa = sentiment_analyse(cleaned_text)
    lll = list(w.keys())
    vvv = list(w.values())
    data = [go.Bar(x=lll, y=vvv, name="GRAPH")]
    layout = go.Layout(title="GRAPH")
    fig = go.Figure(data=data, layout=layout)
    ppp = fig.write_image("sentiment/static/sentiment/image1.jpeg")
    qq = {'value': aaa, 'message': mmm, 'author': aaaa}

    return render(request, "sentiment/final_output.html", qq)


def form(request):
    return render (request, 'sentiment/analyse.html', {})

def upload(request):

        for count, x in enumerate(request.FILES.getlist("files")):
            d={}
            def process(f):
                with open('C:/Users/HP/PycharmProjects/SENTIMENT___ANALYSIS/SA/media/' + f.name, 'wb+') as
destination:
                    for chunk in f.chunks():
                        destination.write(chunk)
                        print(f.name)
                    str = f.name
                    text = open('media/' + str, 'r').read()
                    print("*** The sentiment of the given file ***\n")
                    lower_case = text.lower()

                    cleaned_text = lower_case.translate(str.maketrans(" ", " ", string.punctuation))

                    tokenized_words = word_tokenize(cleaned_text, "english")

                    final_words = []
                    for word in tokenized_words:
                        if word not in stopwords.words('english'):
```

```
            final_words.append(word)
        lemma_words = []

    for word in final_words:
        word = WordNetLemmatizer().lemmatize(word)
        lemma_words.append(word)
    emotion_list = []

    file = open("sentiment/emotions.txt", "r")
    for line in file:
        clear_line = line.replace("\n", '').replace(",", '').replace("'", '').strip()
        word, emotion = clear_line.split(':')
        if word in lemma_words:
            emotion_list.append(emotion)
    file.close()

    w = Counter(emotion_list)

    def sentiment_analyse(sentiment_text):
        score = SentimentIntensityAnalyzer().polarity_scores(sentiment_text)
        if score['neg'] > score['pos']:
            mes = "No matter what you are going through , there's always a  light at the end of the tunnel."
            au = " -DEMI LOVATO "
            return ("NEGATIVE  SENTIMENT", mes, au)
        elif score['neg'] < score['pos']:
            mes = "Never bend your head. Always hold it high. Look the world straight in the eye."
            au = "- HELEN KELLER "
            return ("POSITIVE SENTIMENT", mes, au)
        else:
            mes = " Life has got all those TWIST and TURNS. You've got to hold on tight and off you go."
            au = " -NICOLE KIDMAN"
            return ("NEUTRAL  SENTIMENT", mes, au)

    p, m, a = sentiment_analyse(cleaned_text)
    ll = list(w.keys())
    vv = list(w.values())
    data = [go.Bar(x=ll, y=vv, name="GRAPH")]
    layout = go.Layout(title="GRAPH")
    fig = go.Figure(data=data, layout=layout)
    pp = fig.write_image("sentiment/static/sentiment/image1.jpeg")
    d = {'value': p, 'message': m, 'author': a}
    return(d)


ijk=process(x)
return render(request, "sentiment/final_output.html", ijk)
```