**Python Project Report**

**Sentiment Analysis**

Project Details :

The Project is based on calculating average polarity of the sentiments by counting the number of positive and negative sentiments and then getting their average. We have done sentiment analysis using twitter tweets, courses listed on coursetalk.com and also by manually uploading .csv files containing sentiments column. We have also put in the Visualization part for the sentiments that are analyzed.

1. Twitter Sentiment Analysis:

In this module, the user has to enter any keyword for which he/she wants to do analysis of tweets, and then he has to select the total number of tweets that he wants to analyze for getting average polarity of the sentiments for that subject.

1. Sentiments Analysis of a User Defined Dataset:

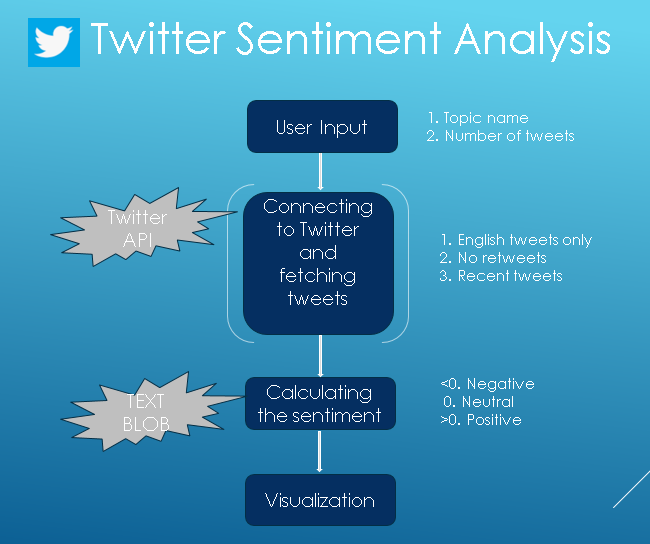
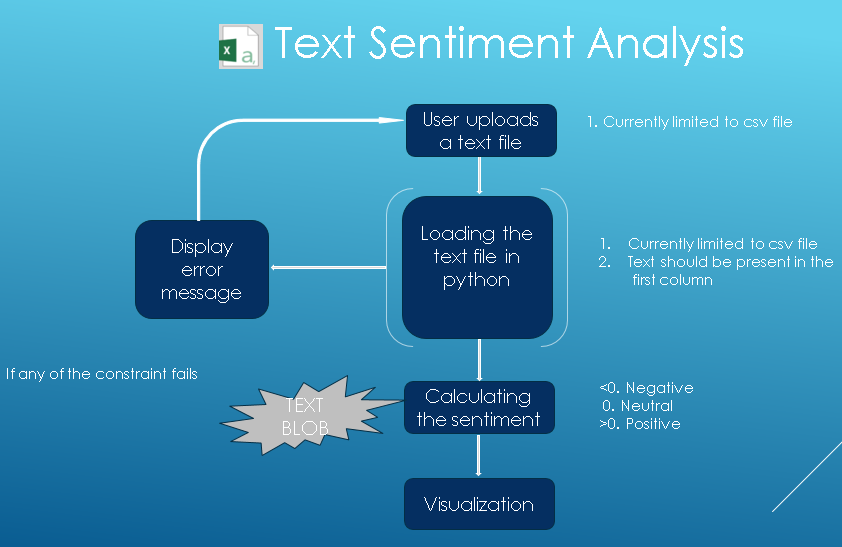
In this module, the webpage will ask the user the user to upload a csv file for his/her choice that he wants to do Sentiment Analysis for. The dataset that is uploaded by the user must have the first column as Sentiments.

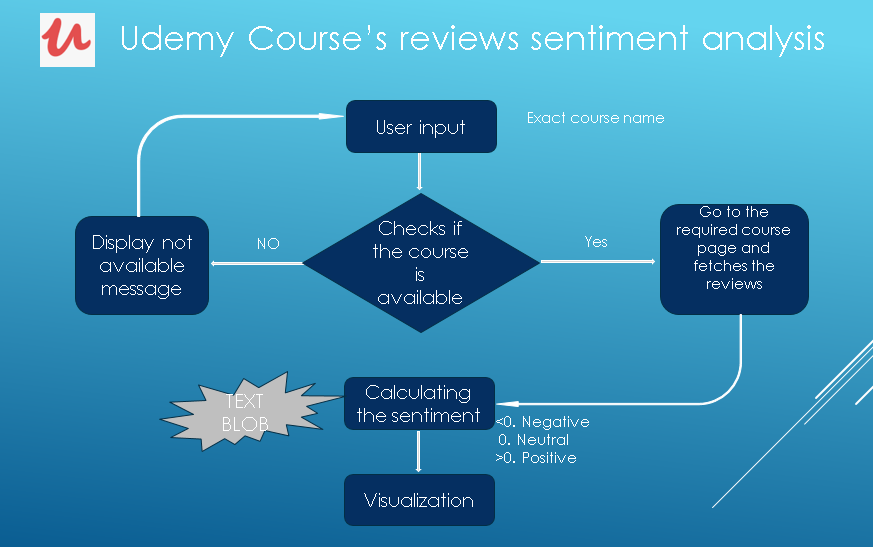
1. Sentiment Analysis for Courses listed on Coursetalk.com:

In this module, the webpage will ask the user to enter a course of his/her choice, the module will then analyze the sentiments associated with the following course and does Sentiment Analysis for it, and returns Average Polarity of the Sentiments for the course.

**II. Work Flow of the Project:**

**1. Twitter Sentiment Analysis. 2. Text File Sentiment Analysis.**

**3.Course Review Sentiment Analysis**.

1. **Twitter Sentiment Analysis:**

For the following section , the details are as follows:

1. The Webpage consists of two user input fields,

Enter Topic Name: Here the user needs to enter the subject for which he wants to analyze polarity of tweets.

Enter number of tweets: Here the user needs to enter the number of tweets that he wants to analyze.

(The maximum number of tweets that can be analyzed for a subject is set to 100)

1. The webpage then connects to the twitter API hose, using the API keys (consumer\_key, consumer\_secret, access\_key, access\_token\_secret).
2. The program then searches for the recent number of tweets as specified by the user for the subject that the user as entered.
3. We are only collecting the tweets that satisfy the following conditions: Tweets that are in English Language, Retweets are not considered, Maximum tweets collected are user specified or not greater than hundred, Only recent tweets are collected.
4. All the tweets collected from the API are stored in a variable named **results**:
5. Using Regular Expressions function i.text( ), only the text part is extracted from all the tweets in the results variable and appended in a list named **tweets.**
6. By using the TextBlob package, for all the tweets stored in the tweets list polarity of the tweet is analyzed using .sentiment and .polarity functions.
7. The polarity calculated is then appended to a list named **polarity\_list**.
8. If the polarity comes out to be greater than 0,then it is considered as positive tweet , if polarity is less than 0 then it is considered as negative tweet ,neutral if polarity is 0.
9. Count of the respective variable is increased by 1 as the tweets are encountered.
10. Total\_text variable returns the value of total tweets analyzed by the program.
11. After calculating the polarity of all the tweets in the tweets list,average polarity is calculated using the formula :**sum(polarity\_list)/len(polarity\_list)**
12. In the Visualization part,a scatter plot will be displayed on the output page that is constructed using numbers\_list and polarity\_list.
13. **Sentiments Analysis of a User Defined Dataset:**

For the following section , the details are as follows:

1. The user will be asked to upload a csv datatset of his/her choice that he wants to do sentiment analysis for.
2. The dataset uploaded by the user should mandatorily contain the reviews/comments text in the first column of the dataset and also the dataset should be in csv formst only.
3. After the dataset has been uploaded by the user it is loaded into the python program, the first column containing review text is extracted from the dataset and stored in a list named **df1.**
4. By using the TextBlob package, for all the text stored in the **df1** list polarity of the text is analyzed using .sentiment and .polarity functions.
5. The polarity calculated is then appended to a list named **polarity\_list**.
6. If the polarity comes out to be greater than 0,then it is considered as positive tweet , if polarity is less than 0 then it is considered as negative tweet ,neutral if polarity is 0.
7. Count of the respective variable is increased by 1 as the tweets are encountered.
8. Total\_text variable returns the value of total tweets analyzed by the program.
9. After calculating the polarity of all the tweets in the tweets list,average polarity is calculated using the formula :**sum(polarity\_list)/len(polarity\_list)**
10. In the Visualization part,a scatter plot will be displayed on the output page that is constructed using numbers\_list and polarity\_list.
11. **Sentiment Analysis for Courses listed on Coursetalk.com:**

For the following section , the details are as follows:

1. The user will be asked to enter exact name of the course that he/she wants to do sentiment analysis for.
2. The user should enter the exact course name if he/she fails to do so or if the course is not available in the course list then a message of **Course not available** will be displayed
3. After the appropriate course name is entered and after clicking **Analyze** ,the course name is searched in the excel file which contains course names and their corresponding URL’s
4. The Excel file output.xlsx contains 4 columns:

* **course\_name**: Contains name of the courses
* **course\_description**: Short introduction about what is the course all about
* **review\_nos**: No. of reviews a course has
* **course\_url**: URL of the course

1. The course name which the user entered at input is checked whether if it exists in the excel file
2. If it exists, then course name is taken into new variable **course\_in\_name**, and its corresponding URL and no. of reviews is searched with the help of **.loc** function
3. Using Beautiful Soup all the reviews are extracted for first and then for following next pages and all the reviews are appended in **review\_lst** list.
4. Now for each review in **review\_lst** is then polarity of the text is analyzed using **TextBlob** and depending on intensity of the text it assigns a no. between -1 and 1 and all these conversions are stored in **analysis**.
5. If the **polarity >0**, it is considered as positive tweet and corresponding count of positive tweets is  **incremented by 1** and if the **polarity <0**,it is considered as negative tweetand corresponding count of negative tweets is **incremented by 1** and the residual texts are considered as neutral (**polarity=0)**.
6. After calculation and assignment of tweets in categories they are **average**  is calculated and is printed along with positive,negative,neutral reviews.
7. In order to summarize what analysis the algorithm creates and for better understanding of the analysis a pie chart is created .

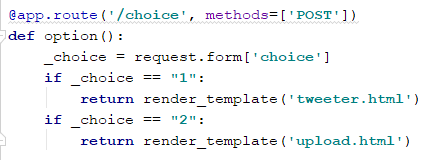
**III. Detailed Working of the Project:**

**A. Working of Twitter Sentiment Analysis.**

**1.Asking User to choose type of sentiment analysis:**



Working Code for the above screenshot.



Explanation:

We have created a function using switch case,which performs according to the choice of the user:

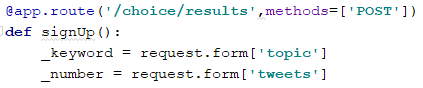
If user chooses 1st option, the webpage will redirect him to the tweeter.html page.

If user chooses 2nd option, the webpage will redirect him to the upload.html page.

If user chooses 3rd option ,the webpage will redirect him to the course\_analysis.html page.

2. Home Page of tweeter.html.

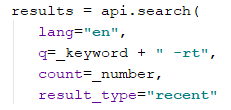
Backend Code for the above screenshot:



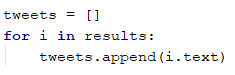
Explanation:

The user needs to enter a topic name and also the number of tweets he wants to analyze. The above code accepts the input for the topic name and number of tweets from input given to the form on the webpage.

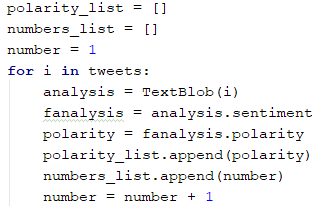
3.Code for Processing the tweets and generating the sentiment.



In the above code the raw tweets that are collected from the twitter api is processed using the api,search function after which only the tweets that are in English language are stored in the results list. Also,the retweets are not taken into consideration using the **-rt** parameter.All the tweets stored in the results list are the most recent tweets associated to the topic.



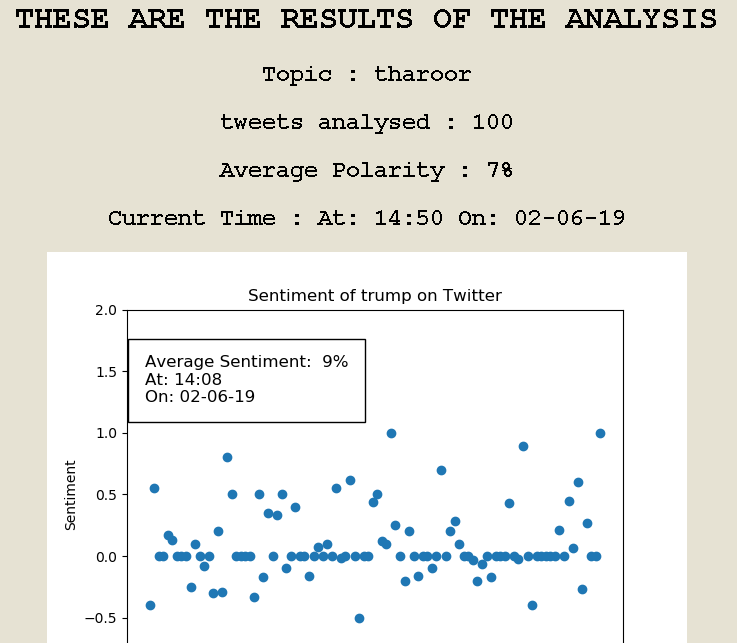
In the above code,only the text part from the tweets are extracted from the results list and appended to the tweets list.



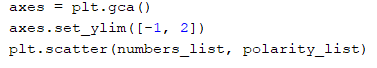
For each tweet in the tweets list,the TextBlob function takes care of the tokenization and lemmatization part required for sentiment analysis,and is applied to each tweet which gives the polarity of the tweet.i.e Positive, if polarity>0

Neutral, if polarity=0

Negative, if polarity<0

1. Output Page of tweeter.html.

Code for the above Scatter Plot.



Code for Average Polarity .



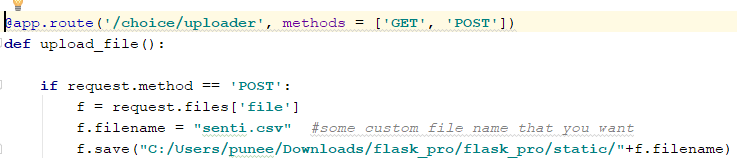
Explanation:

The output screen displays the topic name, no.of tweets analyzed, average polarity of the topic, current date and time and a scatter plot displaying the most recent tweets according to its polarity as classified by the TextBlob function.

B. **Working of Text File Sentiment Analysis.**

1.WebPage for Uploading File:

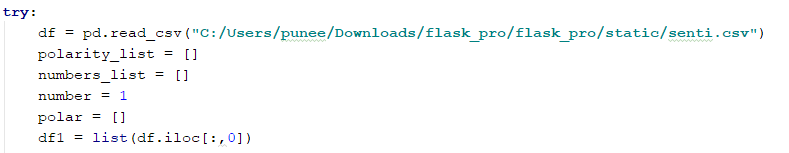


Backend Code for above Webpage:

Explanation:

We have created a function named upload\_file, which will accept the csv file given as input to the form by the user using request.files function and saved to a pre-decided location.

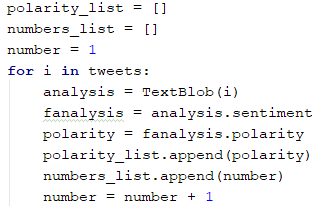
2.BackEnd Code for processing the Dataset:

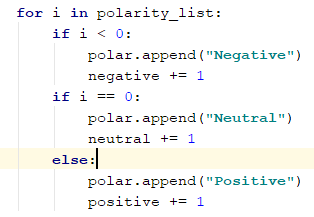


Explanation:

The dataset is read from the location and assigned to a variable. After that only the first column of the dataset which conatains text for sentiment analysis is extracted and assigned to a variable which is then converted to a list.

3.Backend Code For Performing Sentiment Analysis on the dataset:

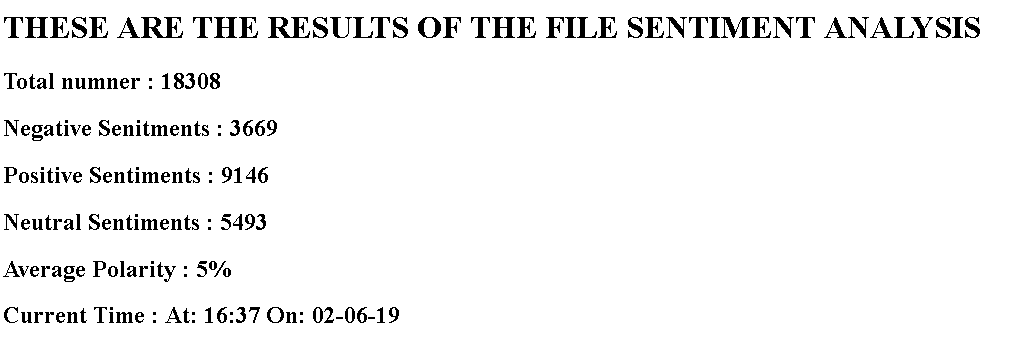




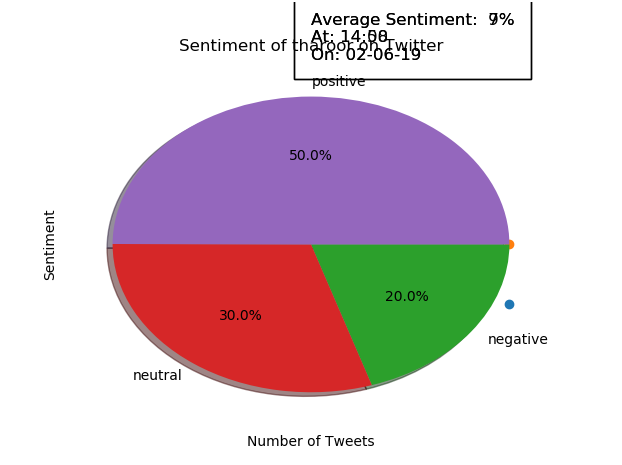
For each text in the df1 list,the TextBlob function takes care of the tokenization and lemmatization part required for sentiment analysis,and is applied to each text which gives the polarity of the text.i.e Positive, if polarity>0

Neutral, if polarity=0

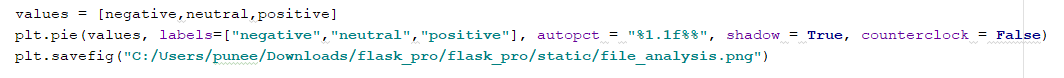
Negative, if polarity<0

4.Output Page for Text File Sentiment Analysis:

5.Visualization Output for Text File Sentiment Analysis.

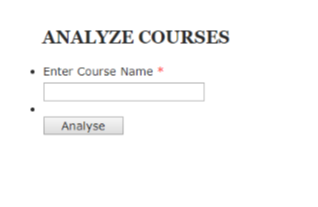


Code for the above Pie Chart.

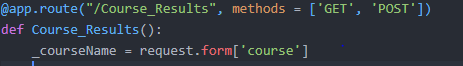


C. Working of Course Review Sentiment Analysis.

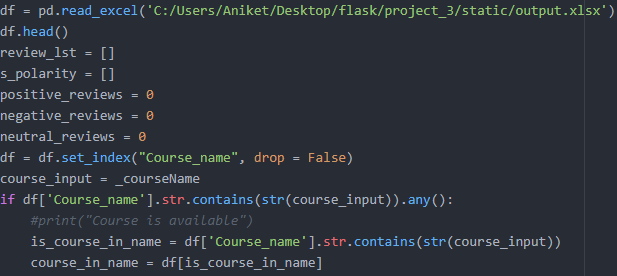
1. Home Page for Course Analysis:



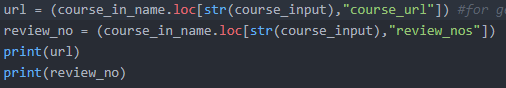
2. The user needs to enter exact course name for successful retrieval of desired output.



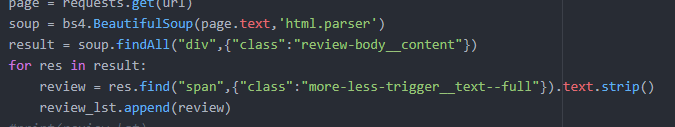
3. As soon as user enters course name in the front end ,Course\_Results() gets activated and the function gets the course name via ‘request.form’ .



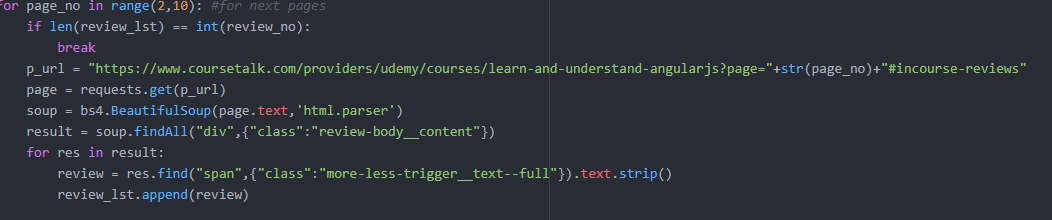
4.The algorithm searches in the excel file and after setting index as Course name if the excel consists the desired course name.



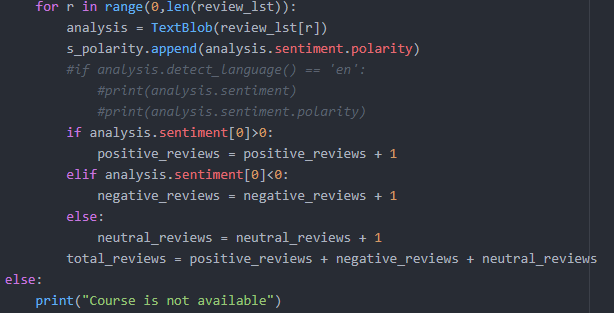
5.To retrieve URL and review no. for corresponding course name, it searches for URL and stores it in variable url,review\_no.



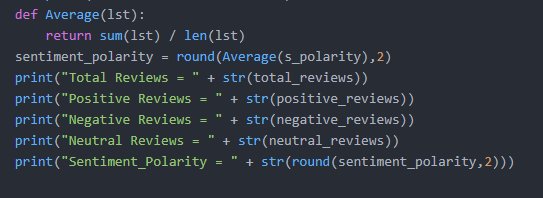
6.To extract reviews for the 1st page we use Beautiful-Soup and we find all similar tags and class and store it in ‘result’. And after parsing through result we obtained reviews and that we have appended it in review



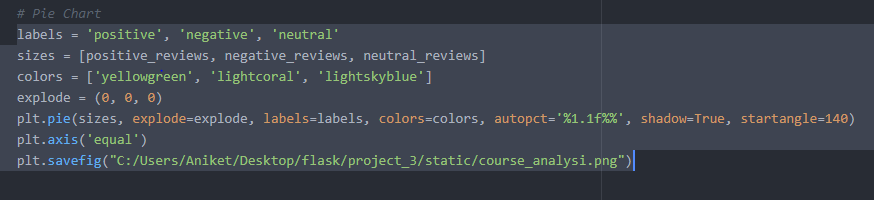
7.Similarly for retrieving multiple pages, same procedure is used and all the reviews for the desired course name is present in ‘review\_lst’



8.Using the ‘TextBlob’ function on the ‘review\_lst’, based on condition reviews are classified as positive, negative and neutral and total count of each category is stored in ‘total\_reviews’. If the course name is not present on the website or if the user has not given exact course name in input then in such cases it will display “Course is not available.”



9.We defined a function ‘Average ‘ to which we pass list containing sentiment polarity which contains polarity(positive/negative/neutral) and degree of subjectivity and all types of reviews are printed



8.And finally for better understanding and visual representation of the analysis a pie chart is created so that it becomes convenient to show what the algorithm is trying to convey via the analysis.

IV. **DEPLOYMENT OF THE PROJECT.**

Steps for deploying the project:

1. Start Command Prompt on the System
2. Create Virtual Environment.

Reason for Creating Virtual Environment: If you try to run the project without creating a virtual environment,and you are using a python version for example 3.6 and a new version of python, then the system will automatically update your older version of python to the latest version of python which will cause the project program to break,hence by running the project program by creating a virtual environment is a safe method to keep the python program intact.

1. Commands for Creating Virtual Environment.
   1. **mkvirtualenv** (name of your virtual environment directory you want to keep).
2. Setting your current virtual environment directory to the one you have created above and move into its root folder.
   1. **workon** (name of your virtual environment folder)
3. Installing all the packages that are required for the project to run.
   1. **pip install flask**
   2. **pip install tweepy**
   3. **pip install datetime**
   4. **pip install textblob**
   5. **pip install matplotlib.pyplot**
   6. **pip install os**
   7. **pip install pandas**
   8. **pip install numpy**

6.Copy your Project Folder that contains the source code, webpage code and all the details that are required for the product to run into your Virtual Environment`s root folder.

7. Final Step, in the command prompt call the python source code file using command.

**python app.py**