PROJECT REPORT

On

**“Antonyms, Synonyms and Description”**

***Submitted By***

SAKSHI SAWATE

***Guided By: -***

Mr. Ratnesh K. Choudhary



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

**S. B. JAIN INSTITUTE OF TECHNOLOGY MANAGEMENT AND RESEARCH, NAGPUR.**

(An Autonomous Institute, Affiliated to RTMNU, Nagpur)

**2021-2022**

**© S.B.J.I.T.M.R Nagpur 2022**

**S.B. JAIN INSTITUTE OF TECHNOLOGY MANAGEMENT AND RESEARCH, NAGPUR**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

SESSION 2021-2022

#### CERTIFICATE

This is to certify that the Project titled **“Antonyms, Synonyms and Description”** is a bonafide work of **Sakshi Sawate** carried out for the partial fulfillment of the requirement for the award of Degree of Bachelor of Engineering in **Computer Science & Engineering.**

|  |  |  |
| --- | --- | --- |
| **Mr. Ratnesh K. Choudhary**  Assistant Professor |  | **Mr. Animesh Tayal**  Head of Department |

**INDEX**

|  |  |  |
| --- | --- | --- |
| CERTIFICATE | | i |
| INDEX | | ii |
| LIST OF FIGURES | | iii |
|  |  |  |
| CHAPTER 1 | INTRODUCTION | 1 |
| CHAPTER 2 | METHODOLOGY | 2 |
| CHAPTER 3 | TOOLS/PLATFORMS | 3-5 |
| CHAPTER 4 | DESIGN & IMPLEMENTATION | 6-9 |
|  | 4.1 ALGORITHM |  |
|  | 4.2 FLOWCHART |  |
|  | 4.3 SOURCE CODE |  |
| CHAPTER 5 | RESULT & DISCUSSION | 10-12 |
|  | 5.1 OUTPUT |  |
|  | 5.2 DISCUSSION |  |
|  | 5.3 APPLICATION |  |
| CHAPTER 6 | CONCLUSION | 13 |
|  | REFERENCES | 14 |
|  |  |  |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIG. NO.** | **TITLE OF FIGURE** | **PAGE NO.** |
| 4.2.1 | Flowchart | 7 |
| 5.1.1 | Installation of pydictionary package | 10 |
| 5.1.2 | Installation of nltk package | 10 |
| 5.1.3 | Download of nltk package | 10 |
| 5.1.4 | Final output with case 1 | 11 |
| 5.1.4 | Final output with case 2 | 11 |
| 5.1.4 | Final output with case 3 | 11 |

**CHAPTER 1**

**INTRODUCTION**

The WordNet is a part of Python’s Natural Language Toolkit. It is a large collection of words and vocabulary from the English language that are related to each other and are grouped in some way. A collection of similar words is called lemmas. Also, It’s a combination of dictionary and thesaurus. It is used for automatic text analysis and artificial intelligence applications. It supports many other languages in its collection.

Nouns, verbs, adjectives, and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations.

PyDictionary is an open-source python library that is used to find the meaning of the words, translation of words and sentences to different languages.

**CHAPTER 2**

**METHODOLOGY**

First, we must install and import python packages NLTK, Wordnet and PyDictionary as these packages gives some perfect methods for the aim of our program.

After that, declare required variables and lists and take required input in a variable.

Check for the availability of given input in Wordnet package using wordnet.synset() method. If input is available then check for availability of synonyms. If synonyms are found then append it in the list for synonyms. Then check for antonyms and append it the list for antonyms.

Then, use PyDictionary(input).printMeanings() method from PyDictionary to find and print the description of given input.

Finally, print all the stored values of respective variables and list as per requirements.

**CHAPTER 3**

**TOOLS/PLATFORMS**

**a. Software:** Python3

**b. Libraries:** nltk, wordnet, Pydictionary

**c. Operating System:** Windows 10

**d. IDE:** Google Colab

**Python 3:** Python 3.0 was released in 2008. Although this version is supposed to be backward incompatibles, later on many of its important features have been backported to be compatible with version 2.7.

Following are important characteristics of python −

* It supports functional and structured programming methods as well as OOP.
* It can be used as a scripting language or can be compiled to byte-code for building large applications.
* It provides very high-level dynamic data types and supports dynamic type checking.
* It supports automatic garbage collection.
* It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.
* Some of the key advantages of learning Python:
* **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
* **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
* **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

**IDE:**

**Google Colab:**

Colaboratory, or “Colab” for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. More technically, Colab is a hosted Jupyter notebook service that requires no setup to use, while providing access free of charge to computing resources including GPUs.

* **Libraries used in this project are as follows:**

1. **NLTK Package:**

NLTK, or Natural Language Toolkit, is a Python package that you can use for NLP. A lot of the data that you could be analyzing is unstructured data and contains human-readable text. Before you can analyze that data programmatically, you first need to preprocess it.

1. **Wordnet Package:**

The WordNet is a part of Python's Natural Language Toolkit. It is a large word database of English Nouns, Adjectives, Adverbs and Verbs. These are grouped into some set of cognitive synonyms, which are called synsets.

To use the Wordnet, at first, we have to install the NLTK module, then download the WordNet package.

1. **PyDictionary Package:**

PyDictionary is a Python Module that helps to get meaning translations, antonyms and synonyms of words. It uses WordNet for getting meanings, Google for translations, and synonym.com for getting synonyms and antonyms. PyDictionary uses Beautiful Soup, Requests module as the dependencies.

* **Operating System:**

**Windows 10:** Windows 10 is a major release of Microsoft's Windows NT operating system. It is the direct successor to Windows 8.1, which was released nearly two years earlier. It was released to manufacturing on July 15, 2015, and later to retail on July 29, 2015. Windows 10 was made available for download via MSDN and TechNet, as a free upgrade for retail copies of Windows 8 and Windows 8.1 users via the Windows Store, and to Windows 7 users via Windows Update. Windows 10 receives new builds on an ongoing basis, which are available at no additional cost to users, in addition to additional test builds of Windows 10, which are available to Windows Insiders. Devices in enterprise environments can receive these updates at a slower pace, or use long-term support milestones that only receive critical updates, such as security patches, over their ten-year lifespan of extended support. In June 2021, Microsoft announced that support for Windows 10 editions which are not in the Long-Term Servicing Channel (LTSC) will end on October 14, 2025.

**CHAPTER 4**

**DESIGN & IMPLEMENTATION**

**4.1 ALGORITHM**

**Step 1 :** Start.

**Step 2 :** Declare two array variables “syn”, “ant” for storing synonym and antonym respectively.

**Step 3 :** Display “Enter any word” and take the input in variable “word”.

**Step 4 :** Check for availability of the value of variable “word” in respective python package.

**Step 5 :** Repeat steps 5.1 to 5.2 if the word is found.

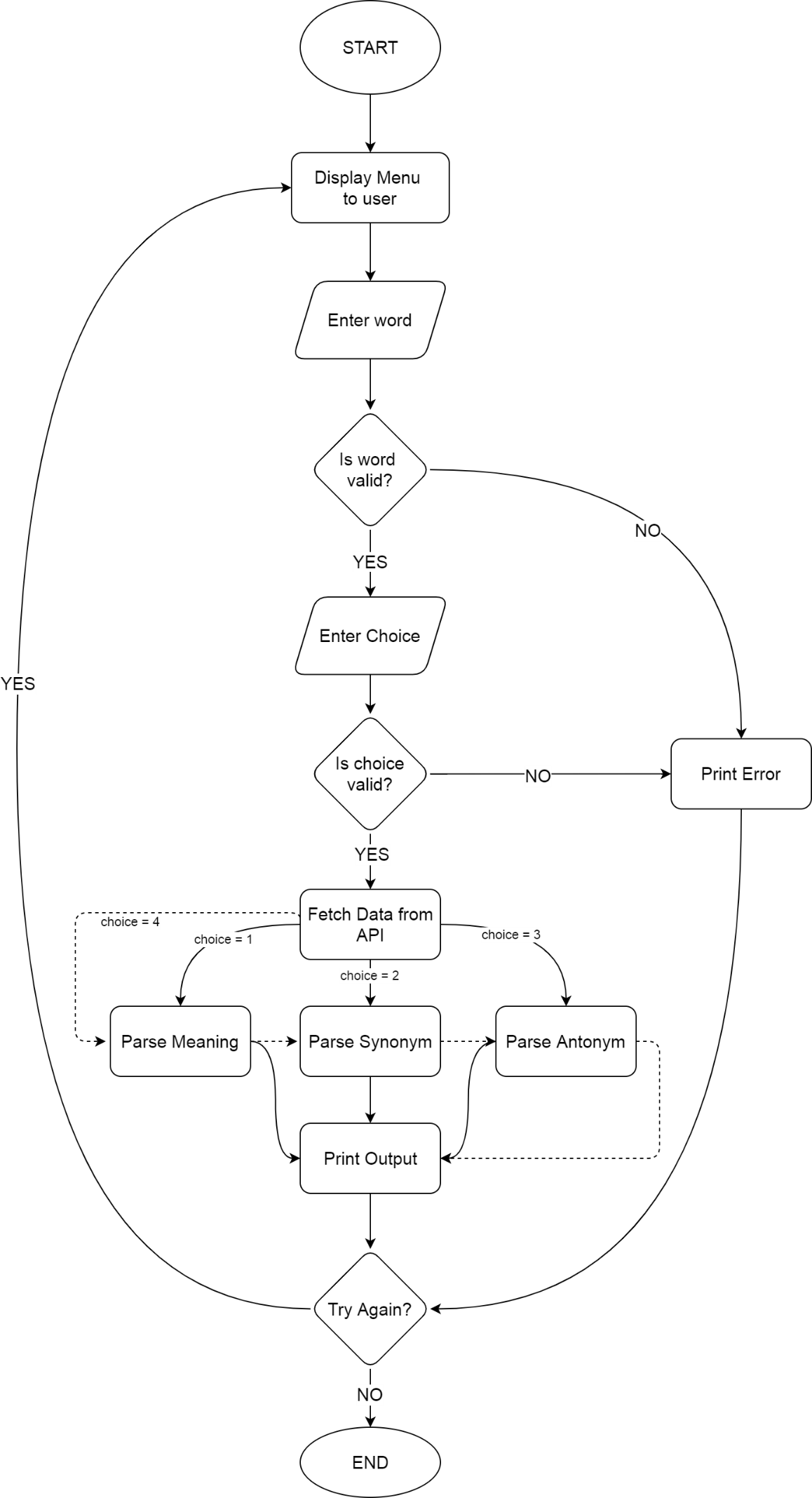
**Step 5.1 :** Check and add synonyms in array “syn”.

**Step 5.2 :** Check for antonyms and add them in array “ant” if found.

**Step 6 :** Declare variable “dictionary” and assign thevalue of function PyDictionary(word) to it.

**Step 7 :** Print values of variables “syn”, “ant” and “dictionary.printMeaning()”.

**Step 8 :** Exit.

**4.2 FLOWCHART**

**Figure (4.2.1):** Flowchart

**4.3 SOURCE CODE**

* **To install nltk package:**

!pip install -q wordcloud

import wordcloud

import nltk

nltk.download('stopwords')

nltk.download('wordnet')

nltk.download('punkt')

nltk.download('averaged\_perceptron\_tagger')

import pandas as pd

import matplotlib.pyplot as plt

import io

import unicodedata

import numpy as np

import re

import string

import nltk

nltk.download('omw-1.4')

* **To install Pydictionary package:**

!pip install PyDictionary

import PyDictionary

* **Code:**

import nltk

from nltk.corpus import wordnet #Import wordnet from the NLTK

syn = list()

ant = list()

word=input("Enter any word: ")

for synset in wordnet.synsets(word):

for lemma in synset.lemmas():

syn.append(lemma.name()) #add the synonyms

if lemma.antonyms(): #When antonyms are available, add them into the list

ant.append(lemma.antonyms()[0].name())

from PyDictionary import PyDictionary

dictionary=PyDictionary(word)

print('DESCRIPTION of',word)

print(dictionary.printMeanings())

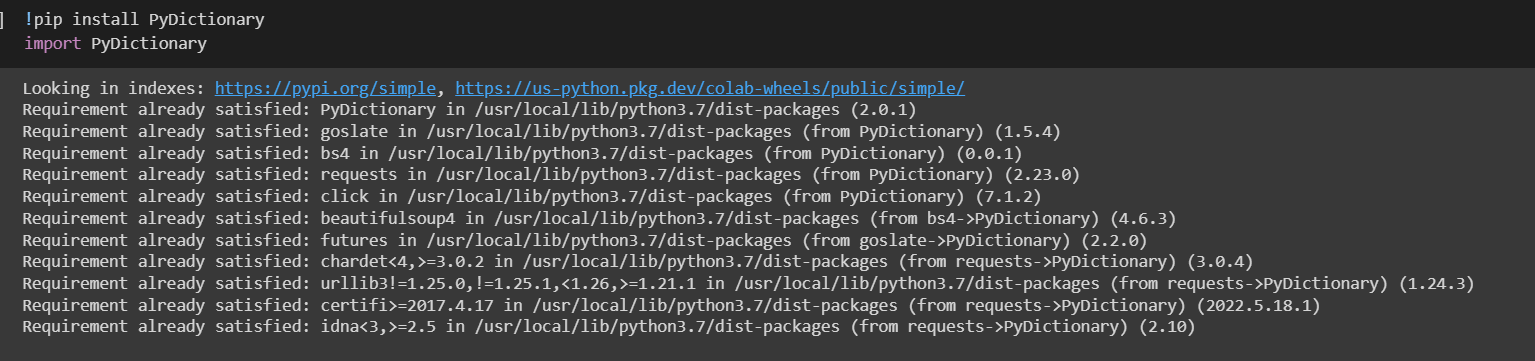
print('SYNONYMS: ' + str(syn))

print('ANTONYMS: ' + str(ant))

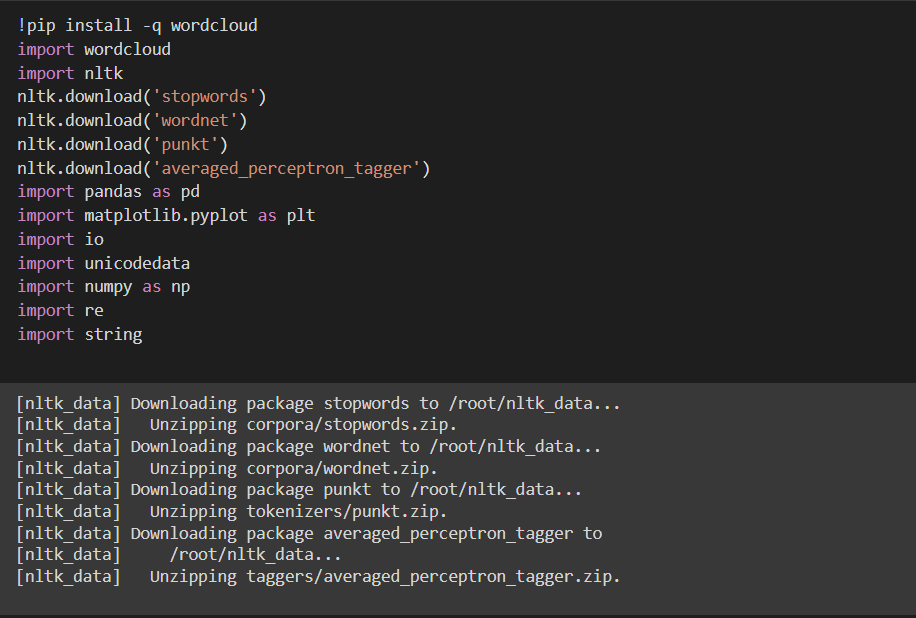
**CHAPTER 5**

**RESULT & DISCUSSION**

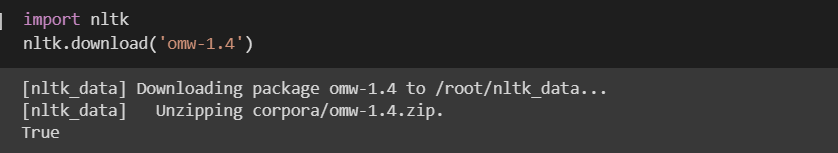
**5.1 OUTPUT**

****

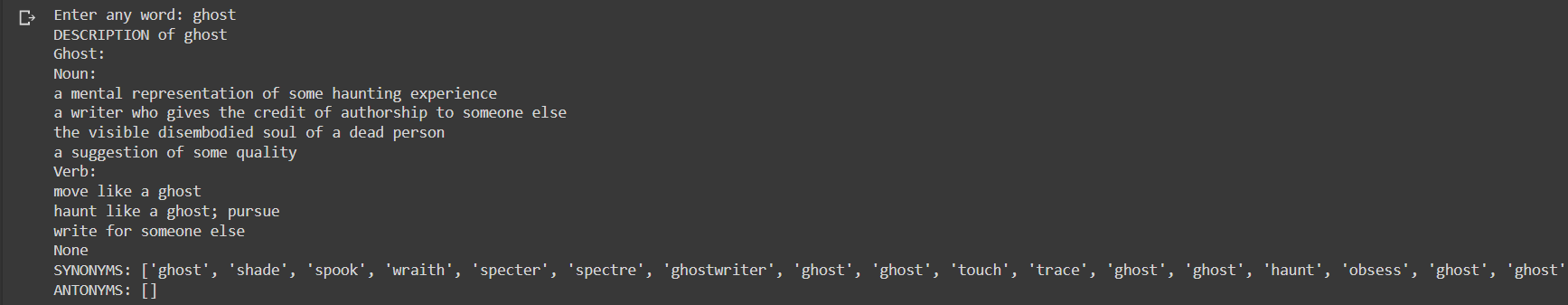
**Figure (5.1.1):** Installation of pydictionary package

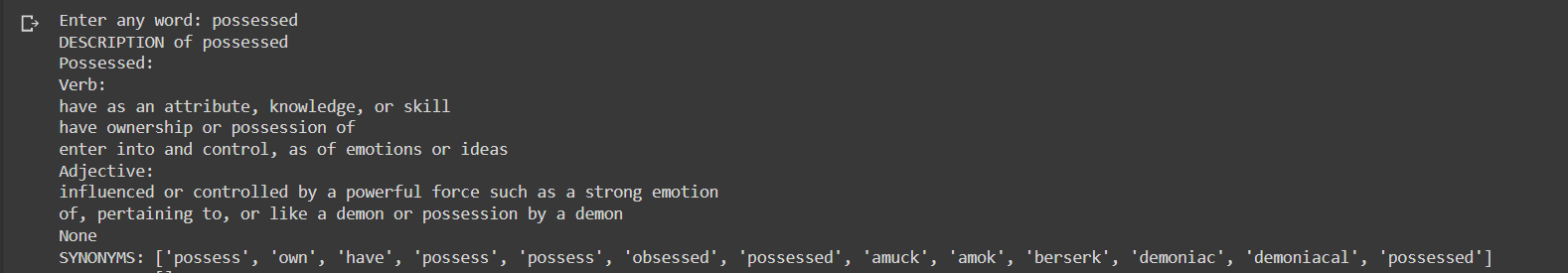
****

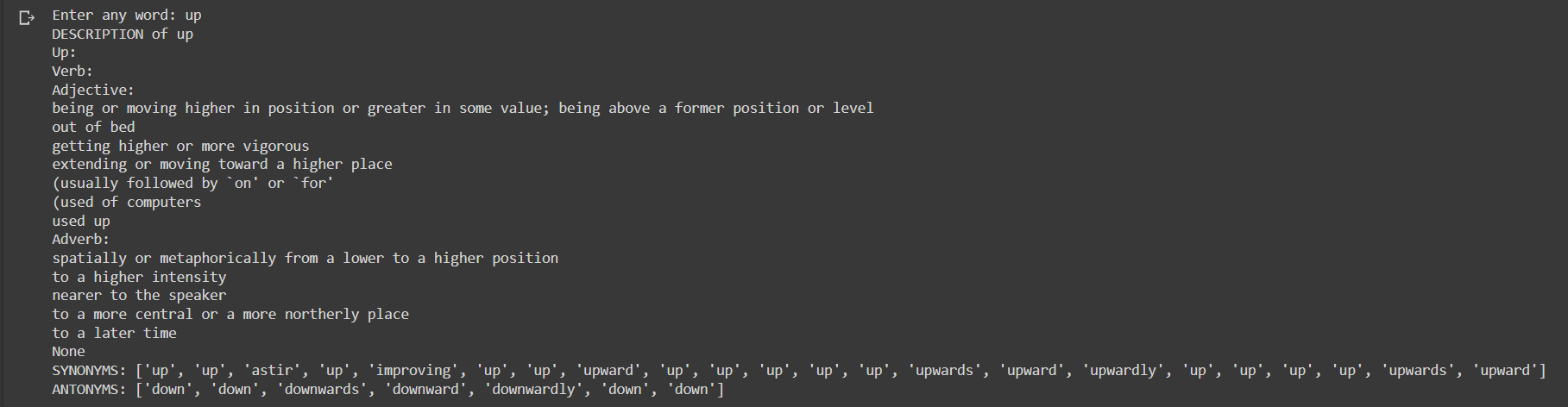
**Figure (5.1.2):** Installation of nltk package

****

**Figure (5.1.3):** Download of nltk package

****

**Figure (5.1.4):** Final output with case 1

**Figure (5.1.4):** Final outputs with case 2

**Figure (5.1.4):** Final outputs with case 3

**5.2 DISCUSSION**

In this project, we used different types of new python packages like NLTK, Wordnet and PyDictionary in order to perform the given task. Generally, we use these packages for character-based data types’ projects. At the end of this project we are able to use these packages with ease and we ran many test cases and we got almost all of the desired outcomes. The package PyDictionary gives almost all types of available descriptions for given input, like if we give an input “possessed” it will give the noun as well as the verbal description for the word “possessed”.

**5.3 APPLICATION**

Using this project, we can find similar words, opposite words and meaning of the given word.

This project can be used as Dictionary to the user which contains antonyms, synonyms and meaning of the word.

Dictionary in Python is an unordered collection of data values, used to store data values like a map, which unlike other Data Types that hold only single value as an element, Dictionary holds key: value pair. Key value is provided in the dictionary to make it more optimized. Each key-value pair in a Dictionary is separated by a ‘colon’, whereas each key is separated by a ‘comma’.

A Dictionary in Python works similar to the Dictionary in a real world. Keys of a Dictionary must be unique and of immutable data type such as Strings, Integers, and tuples, but the key-values can be repeated and be of any type.

**CHAPTER 6**

**CONCLUSION**

I have designed and developed a code to find Antonyms, Synonyms and Meaning of a word. This can be used as a Dictionary. It is easier to use. It provides detail information of the entered word. It can improve the vocabulary of the user.

**REFERENCE**

* Wikipedia
* Geeks for Geeks
* Tutorials point

**Notes:**

**1. Chapter Title –** Front size = 16, Front Type = Times New Roman, Alignment = Centered,

Bold.

(Example – INTRODUCTION)

**2. Heading –** Front size = 14, Front Type = Times New Roman, Alignment = Left, Bold.

(Example – 4.1 ALGORITHM)

**3. Sub Heading -** Front size = 12, Front Type = Times New Roman, Alignment = Left, Bold.

(Example – Module)

**4. Content -** Front size = 12, Front Type = Times New Roman, Alignment = Left.

(Example – Write down proper information about you project)

**5. Figure –** Give proper name to your figure and screenshot.

Front size = 12, Front Type = Times New Roman, Alignment = Centered.

Fig. Chapter no. name of figure/screenshot

(Example – Fig. 4 Flowchart of Tic-Tac-Toe)