**TRF Task 2 -**

**Web Scrapping**

## 

## -Ritvik Patil

## -Soham Bhure



# **INDEX**

Group Number:

Project Title: Web Scrapping (Dictionary and NEWS Articles)

**Member 1** **Member 2**

Name: Ritvik Patil Name: Soham Bhure

Div. & Roll Number: E-40 Div. & Roll Number: E-59

GR Number:11810490 Gr Number: 11810736

E-mail ID: [ritvik.patil18@vit.edu](mailto:ritvik.patil18@vit.edu) E-mail ID: [soham.bhure18@vit.edu](mailto:soham.bhure18@vit.edu)

# **INTRODUCTION:**

## **Web Scrapping –**

Web Scrapping is the collection of information from the Internet (generally from some particular site). Generally, this is done with software that simulates human Web surfing to collect specified bits of information from different websites.

Web scrapped data can be used to sell to users or for promotional purposes for some website. Web scraping is also called Web data extraction, screen scraping or Web harvesting.

It is a part of data mining where data is mined from sites and used for various purposes. Few techniques of web scrapping are illegal as data can be stolen from a site and used elsewhere.

# **WORKING METHODOLOGY:**

Main Libraries used for Web Scrapping:

1. **Beautiful Soup:**

Beautiful Soup is a Python library for pulling data out of HTML and XML files. It is an useful software for data mining and web scrapping that helps you clean up and parse the documents you have pulled down from the web.

1. **Scrappy:**

Scrapy is a free and open-source web-crawling framework written in Python. It was originally designed for web scraping, but it can also be used to extract data using APIs or as a general-purpose web crawler.

1. **Selenium:**

Selenium is an open-source web-based automation tool. Selenium can send the standard python commands to different browsers, despite variation in their browser's design.

## **TASK 2:**

## **PART 1: Dictionary using Web Scrapping**

## **PART 2: Displaying News Articles using Web Scrapping**

## **Libraries used:**

## bs4:

The library bs4 contains all features of Beautiful Soap 4. Beautiful Soup is a library that makes it easy to scrape information from web pages. It sits atop an HTML or XML parser, providing Pythonic idioms for iterating, searching, and modifying the parse tree.

## Requests:

The requests library is the standard for making HTTP requests in Python. It abstracts the complexities of making requests behind a beautiful, simple API so that you can focus on interacting with services and consuming data in your application.

## **PART 1: Dictionary using Web Scrapping**

### Algorithm:

* Import libraries.
* Store a site (which is to be used to search the word) as a string in a variable.
* Request to get site using the variable as parameter.
* Get the back end of the site using BeautifulSoup() and passing requested site and ‘html.parser’
* Inspect the site which is to be scrapped.
* Find the correct class which is to be scrapped from the site and print out the scrapped data.

### Working:

Basically, we are trying to find the correct text that is to be printed on the console. This is done by inspecting the elements from a particular site and finding the right class.

After finding the correct class we have to scrape the data under it by using functions and methods under Beautiful Soup.

Lastly, we have to output the data in good readable format. For that methods like get\_text() and counters are used.

## **PART 2: NEWS Articles using Web Scrapping**

### **Algorithm:**

* Import libraries.
* Store a site (which is to be used to search the word) as a string in a variable.
* Request to get site using the variable as parameter.
* Get the back end of the site using BeautifulSoup() and passing requested site and ‘html.parser’
* Inspect the site which is to be scrapped.
* Find the correct class which is to be scrapped from the site and print out the scrapped data.

# **INSIGHTS:**

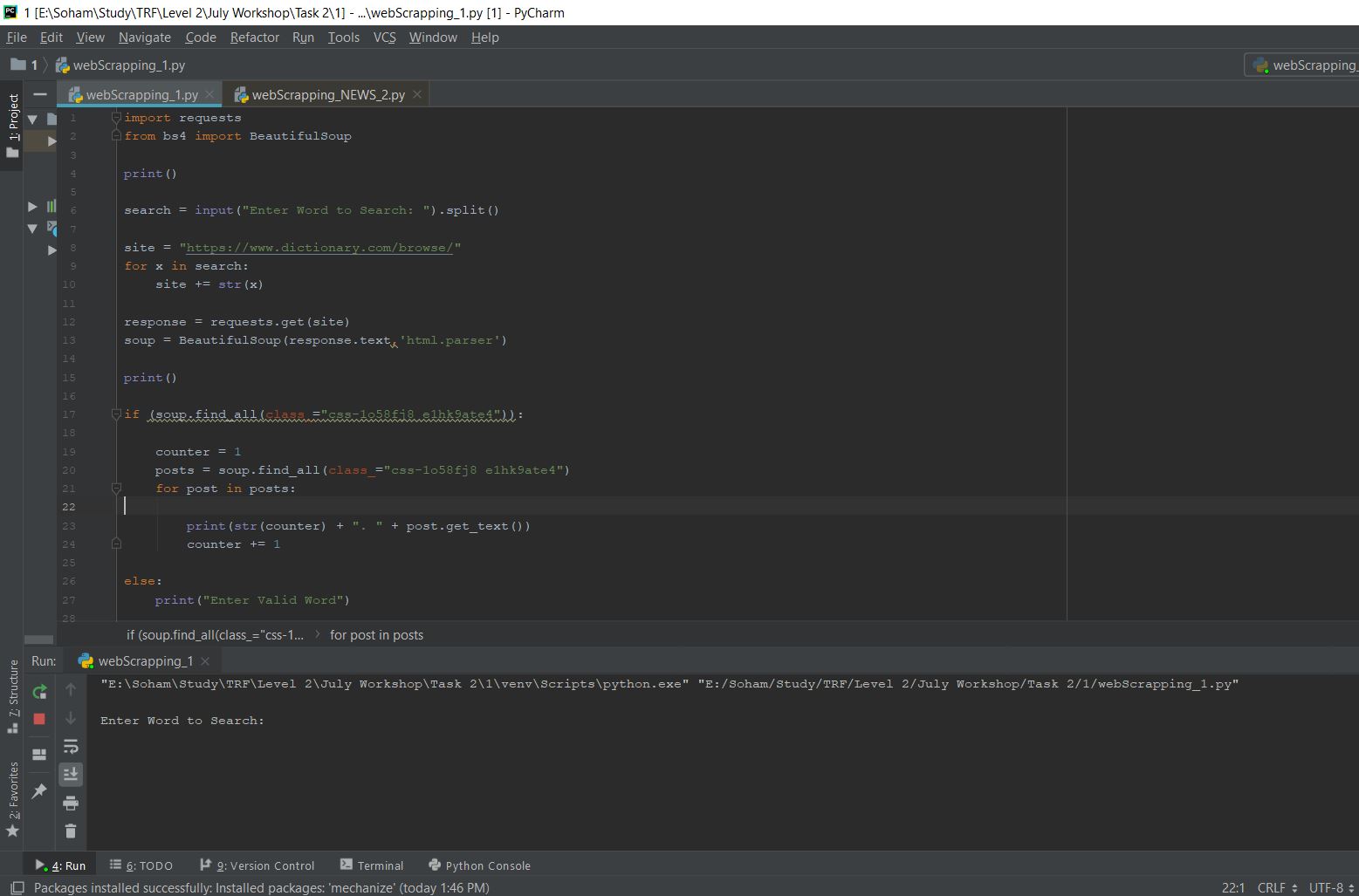
**Problems Faced:**

* Creating the correct url for the words entered by the user.
* To find the exact tag and class of the required information on website by inspecting the website.
* To print the collected data from website in a good readable format.

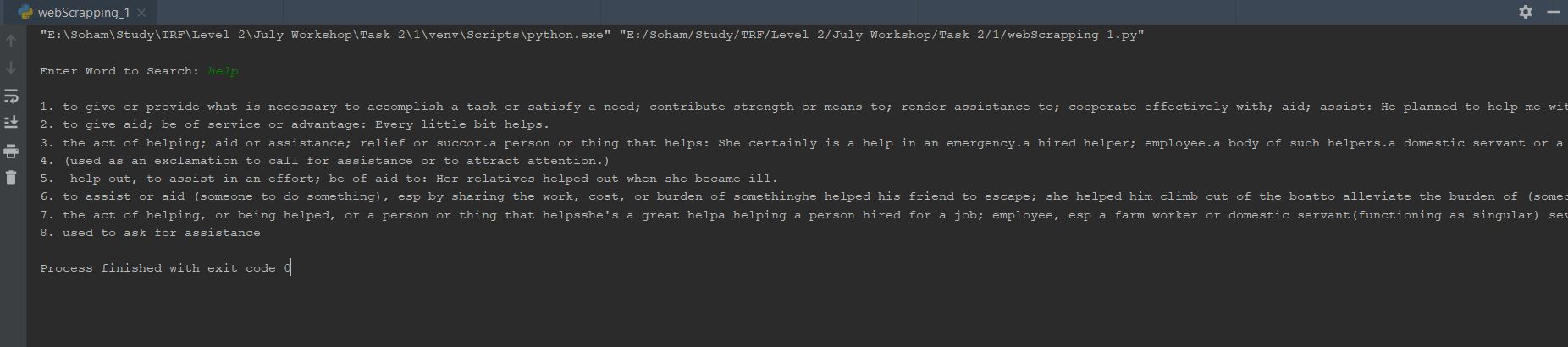
# **RESULTS:**

## **PART 1:**

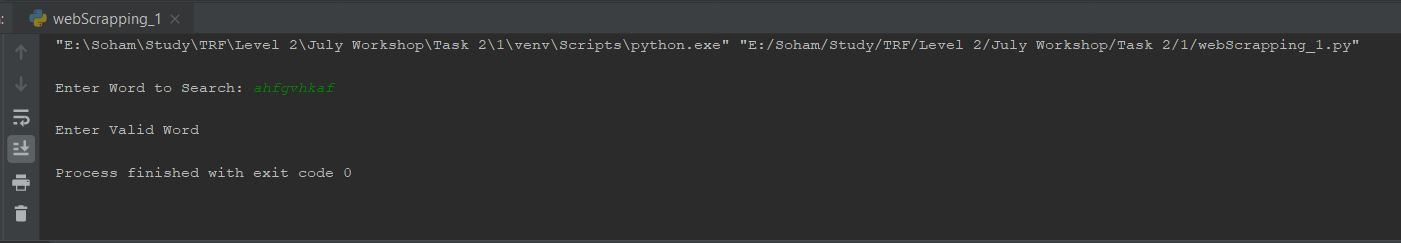
### **CODE:**



### **OUTPUT:**

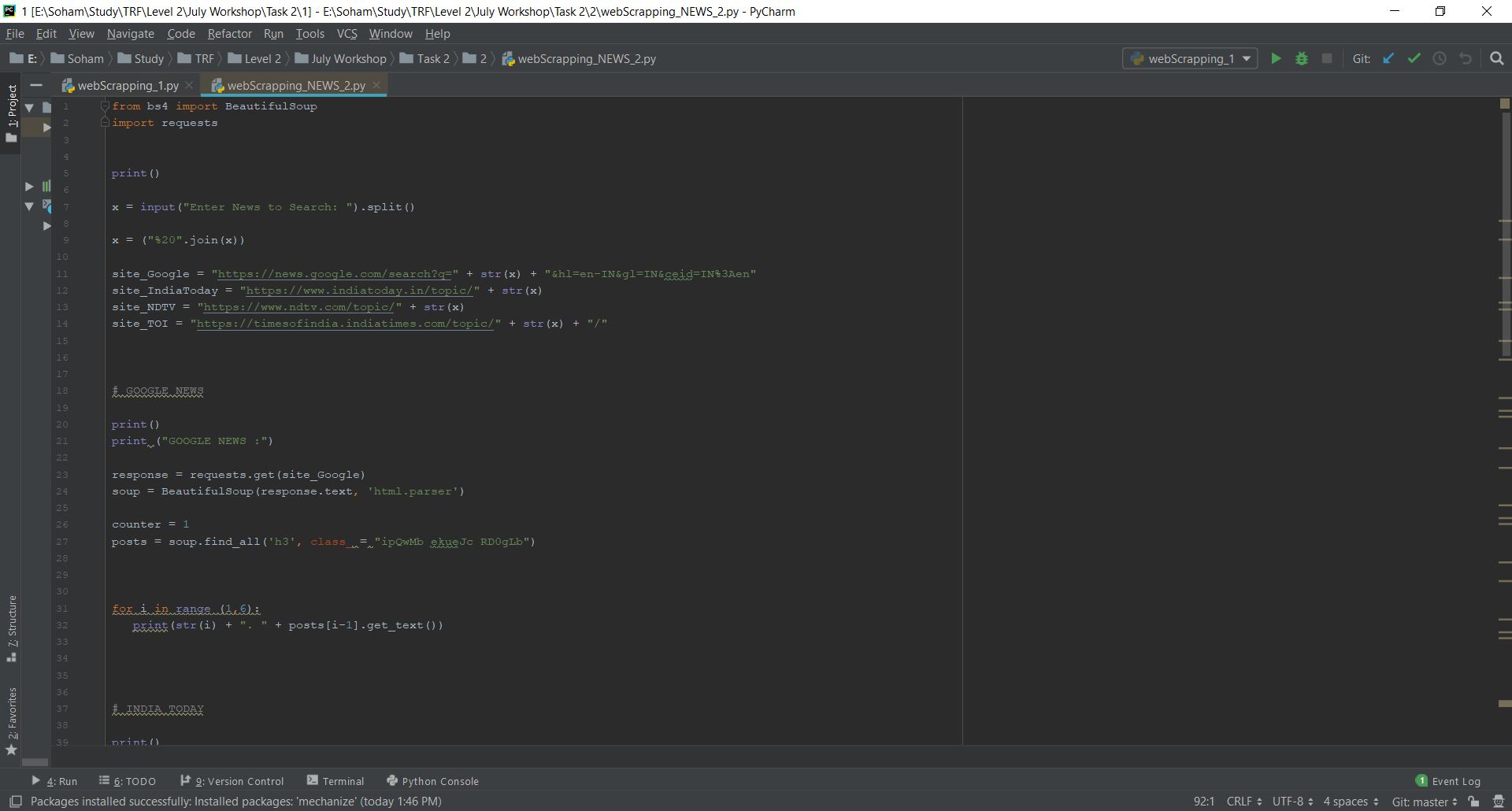


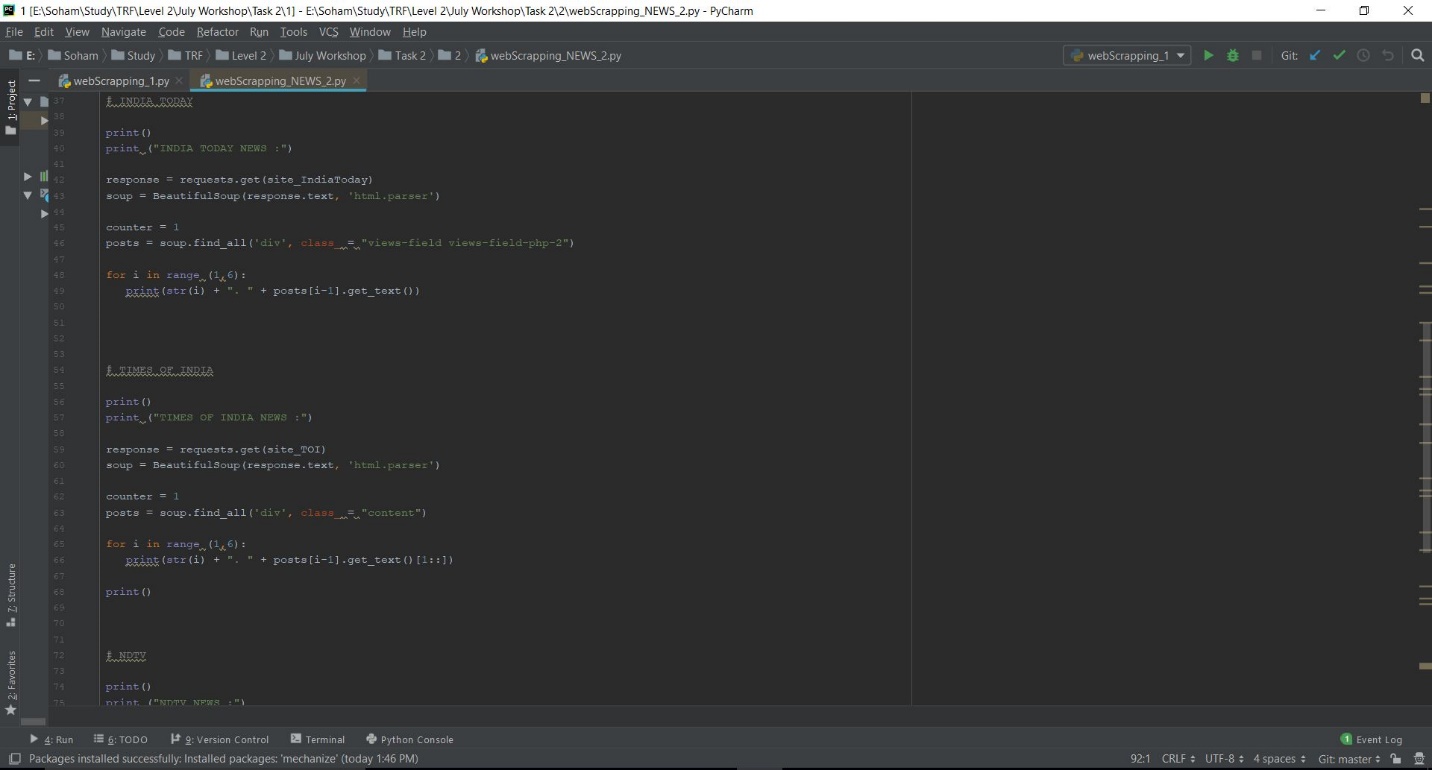
**FOR INVALID INPUT :-**

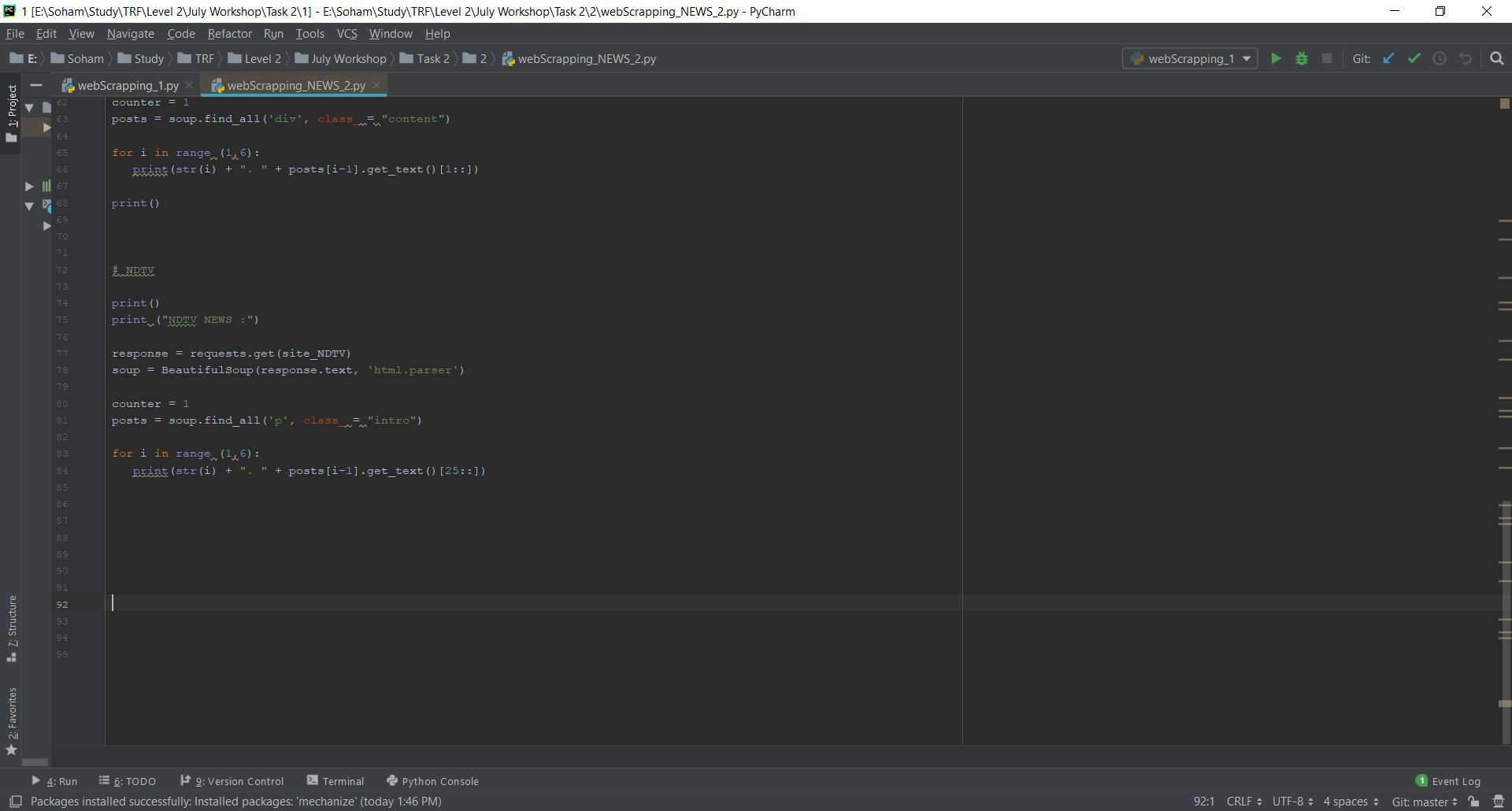


## **PART 2:**

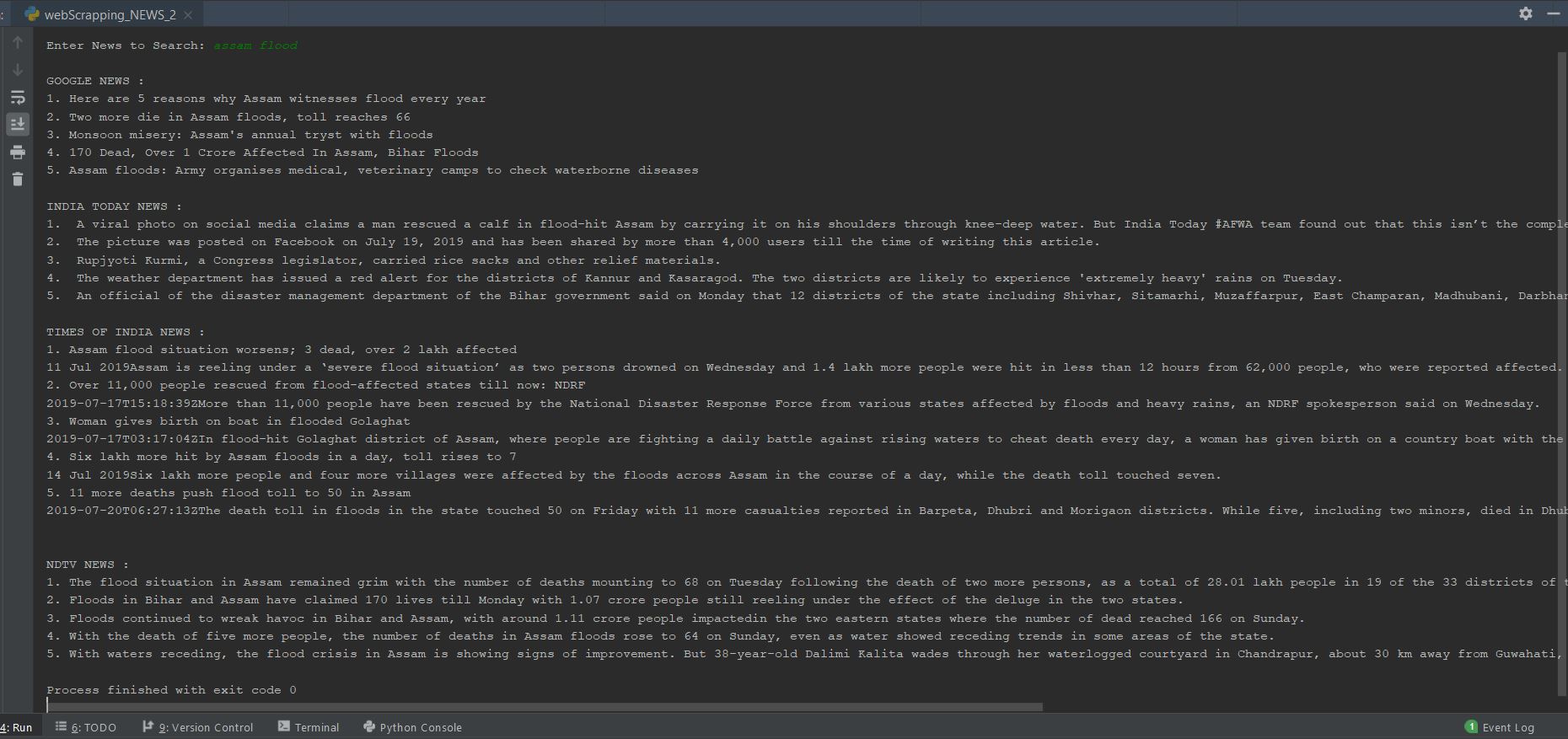
### **CODE:**







### **OUTPUT:**



# **REFERENCES:**

* [www.google.com](http://www.google.com)
* [www.stackoverflow.com](http://www.stackoverflow.com)
* [www.crummy.com](http://www.crummy.com)
* [www.techopedia.com](http://www.techopedia.com)
* [www.news.google.com](http://www.news.google.com)
* [www.timesofindia.in](http://www.timesofindia.in)
* [www.indiatoday.in](http://www.indiatoday.in)
* [www.indianexpress.com](http://www.indianexpress.com)
* [www.hindustantimes.com](http://www.hindustantimes.com)
* [www.ndtv.com](http://www.ndtv.com)
* [www.youtube.com](http://www.youtube.com)
* [www.github.com](http://www.github.com)
* [www.programminghistorian.org](http://www.programminghistorian.org)