# Project Report

###### Of

**Web Scraping**

**BACHELOR OF TECHNOLOGY**

**In**

**COMPUTER SCIENCE AND ENGINEERING**

***By***

**VAIBHAV KUMAR**

**Registration number: 11811376**

**Roll no:A-23 Section: K18RD**



**School of Computer Science and Engineering**

Lovely Professional University

Introduction:

What is web scraping:

In simple word web scraping is extraction of data from web.

Why to use web scraping:

we can say that instead of manually saving the data from websites, the web scraping software will automatically load and extract data from multiple websites as per our requirement.

Aims of this project:

Uses of web scraping:

1) A lot of digital businesses that rely on data harvesting such as search engine bots that crawl a site, analyze its content and then rank it.

2) Scrape product details (price, images, rating, reviews etc.) from retailer/manufacturer/e-Commerce websites.(Ex: Amazon, eBay, AliExpress, Alibaba etc.) to show on own websites, to provide price comparisons, to perform a price watch on competing sellers etc.

3) Scrape reviews of products and places/hotels/restaurants

4) Market research companies using scrapers to extract data from forums and social media.

5) Web Scraping helps you to gather data for testing / training your Machine Learning models and much more.

**Because of its uses I decided to build an understanding of the topics of web crawling techniques and web scrapers--**

In this project, I have done web scraping as well as web crawling:

So, first of let me tell you what is difference between web scraping and web crawling:

Web crawling is what search engine do, it is all about viewing multiple web pages or web sites as a whole and indexing it. when a bot crawls a website, it goes through every pages and every link until the last line or page of the website.

While web scraping mainly focusing on content(data) of web pages.

**Finally, aim of my project is to implement its uses as much I can--**

## Python Modules for Web Scraping:

## Requests

It is a simple python web scraping library. It is an efficient HTTP library used for accessing web pages. With the help of **Requests**, but its API is better than Urllib3

## Urllib3

It is another Python library that can be used for retrieving data from URLs similar to the **requests** library.

## Selenium

It is an open source automated testing suite for web applications across different browsers and platforms. It is not a single tool but a suite of software. We have selenium bindings for Python, Java, C#, Ruby and JavaScript. Here we are going to perform web scraping by using selenium and its Python bindings. Selenium Python bindings provide a convenient API to access Selenium WebDrivers like Firefox, IE, Chrome, Remote etc.

## Scrapy

Scrapy is a fast, open-source web crawling framework written in Python, used to extract the data from the web page with the help of selectors based on XPath.

Libraries used:

1. Beautiful soup:

I have used for getting the html and xml files(source code of web pages).

1. Requests:

I have used send HTTP requests

1. Pandas:

I have used to make data in structure and then for data analysis.

1. Scrapy [not library , it is framework]:

All of the above libraries use for web scraping but scrapy can be use for both web scraping and web crawling.

1. Selenium:

I have used this package to examine every aspect of a website i.e. for doing dynamic web scrapping.

Many websites like fare of flights and share marketing value changes their content of web-pages within a sec.

So to scrap such websites we need selenium package.

1. Time:

To examine every aspect of a website related to time, i need time related function to keep refreshing it every second.

1. tkinter:

To add a photo-Image object to a tkinter widget and for making button, frame and label. In my project.

Working of Beautiful soup:

1) Install the required third-party libraries.

2) Accessing the HTML content from webpage.

a) Specify the URL of the webpage you want to scrape.

b) Send the HTTP request to the specified URL and save the response from server in a response object.

c) Parse the HTML content by the help of lxml or html.parser provided by Beautiful Soup.

d) Searching and navigating through the parse tree.[by find() function].This is very important part of web scraping , we have to understand the structure of HTML code completely.

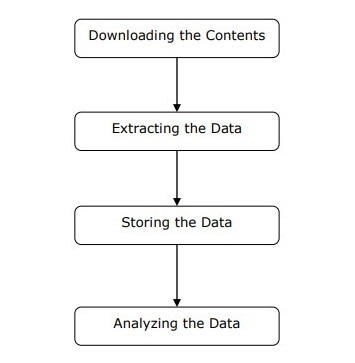
It can be done either by CSS selector or by X-path.

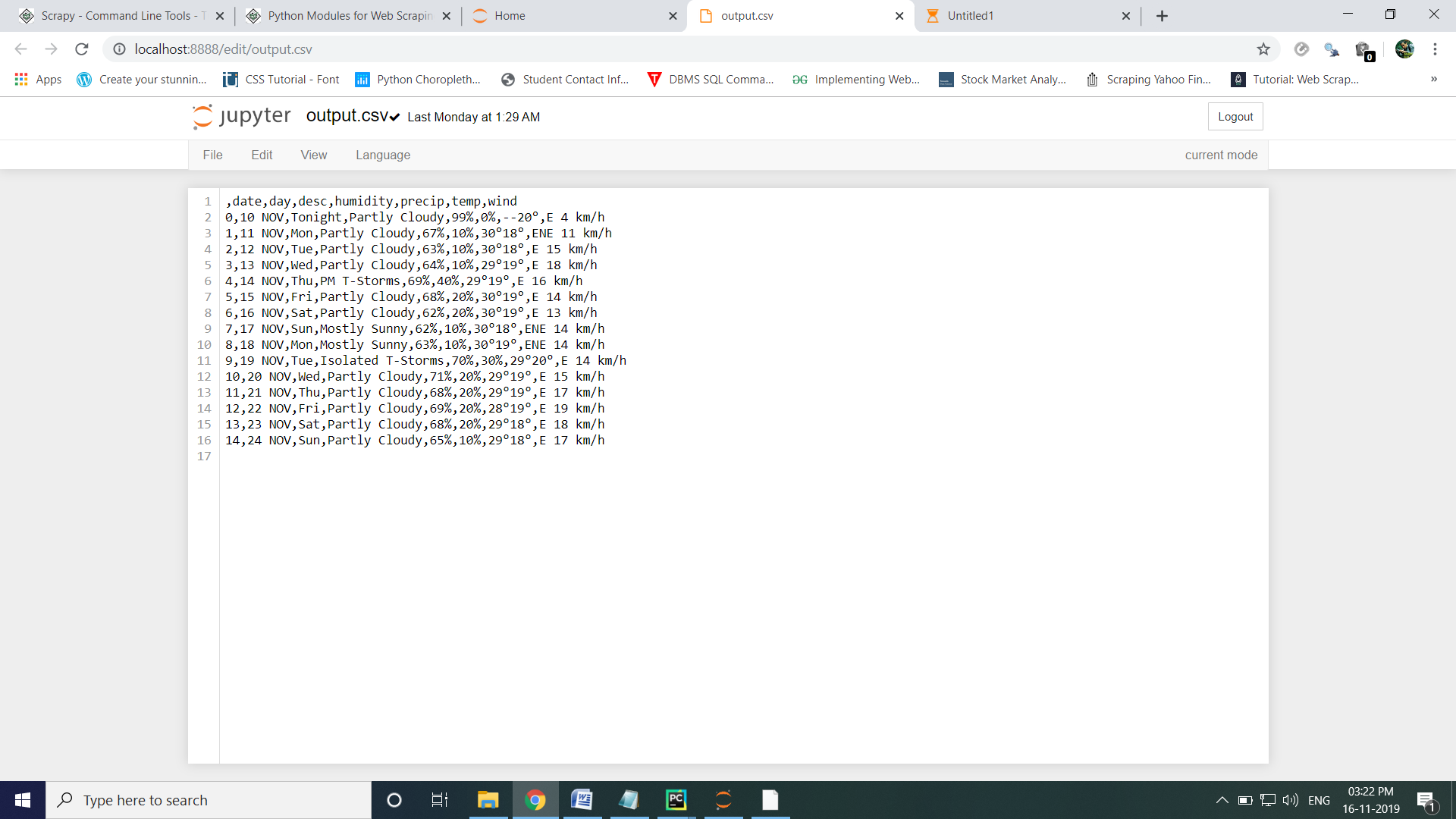
e) Finally, I saved data outcome as CSV file.

i) Create list to store data.

ii) Then data-frames these list using pandas.

iii) Now, import these table in CSV or JSON etc. files.



**Output by selenium of dynamic weather forecasting:** ****

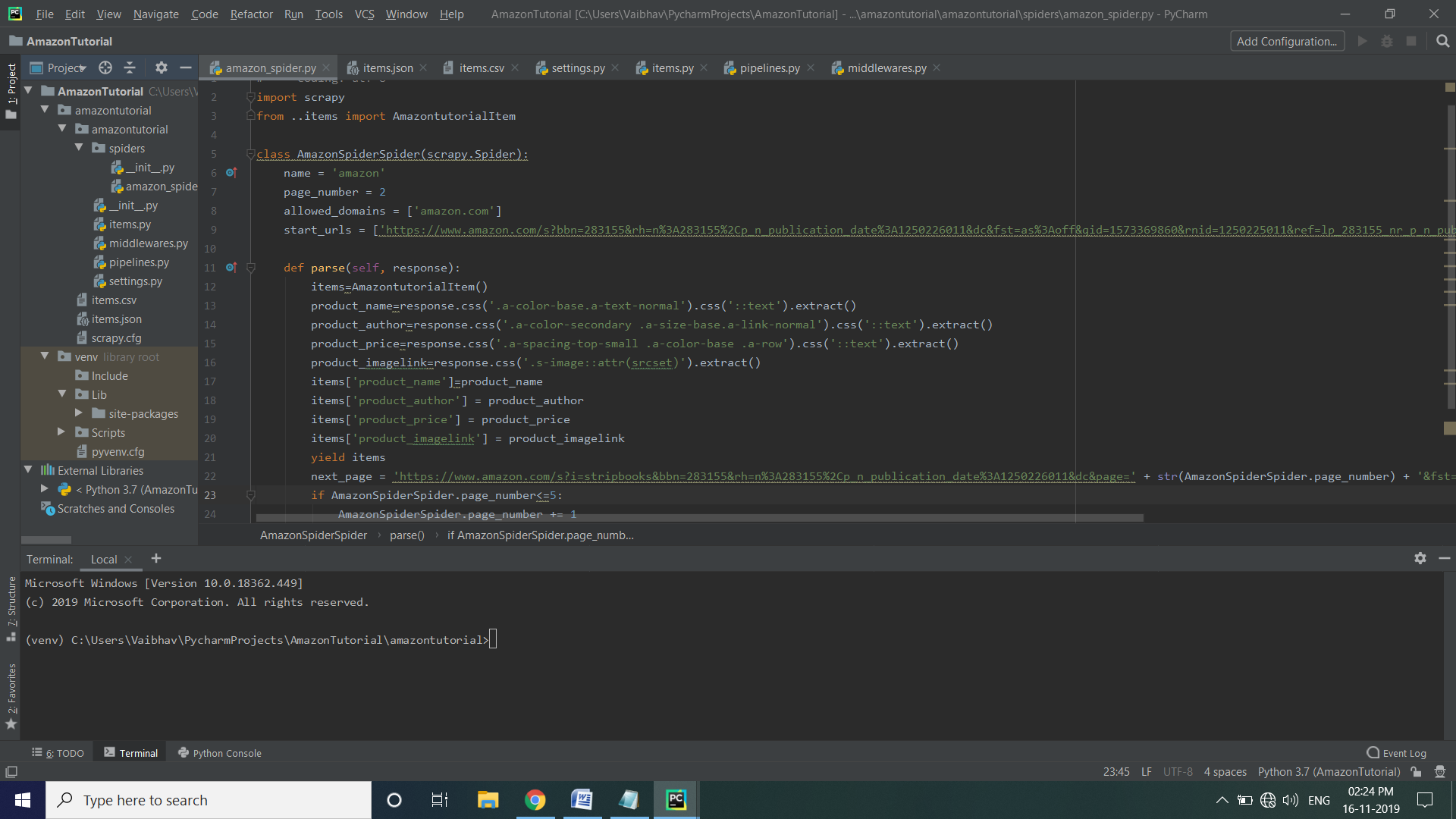
Working of scrapy:

1) **First of all we have create virtual environment so that we can work on our project without affecting the other project.**

This mean that each project have its own dependencies.

2) **Now write scrapy startproject <new project name>**

**This is structure of my project [ Created automatically by Py-Charm] :**

****

3) **Define your data structure.**

a) In items.py

Class AmazontutorialItem(scrapy.Item):

# define the fields for your item here like:

# name = scrapy.Field()

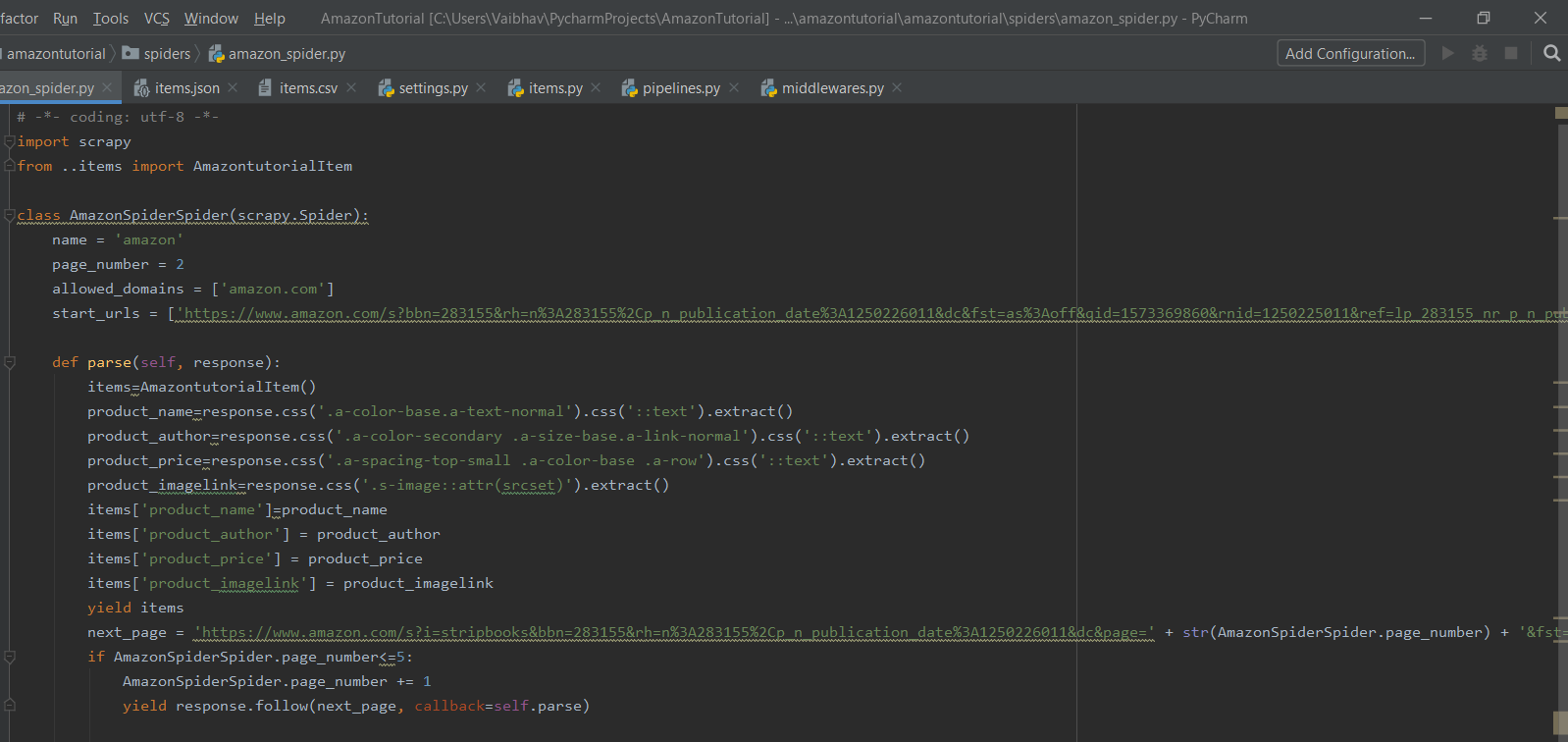
product\_name = scrapy.Field()

product\_author = scrapy.Field()

product\_price = scrapy.Field()

product\_imagelink = scrapy.Field()

3) **Create a spider:**



Now in code:

**Name**: Identifies the spider . It must be unique.

**start\_url** : it is list of URLS where the spider will begin to crawl form.

**Parse** : it is method of the spider, which will be called with the downloaded response object of each start URL.

4) **Output:**

To store the information of our spider we have to execute the following command:

scrapy runspider <name> \_o items.json(.csv or .xml) .

5) **In pipelines.py:**

In this section :

1. We import a class of your own data structure.
2. Extend process\_item() function.
3. Add to ITEM\_PIPELINES at settings.py.
4. To store the scraped item in a database.

**Command:**

**Creating a Project:**

scrapy startproject project\_name

cd project\_name

### Controlling Projects:

scrapy genspider <name>

**Run:**

scrapy runspider <name>

Proxy:

Some websites doesn't allow to scrap from their content from any pages like Amazon because of certain privacy.

So , to overcome over this problem we can use rotation user agents and second method is proxy.[more efficient]

However it is illegal to do web scrapping via these methods.

1) **Users agent:**

Giving the random users----

DOWNLOADER\_MIDDLEWARES = {

'scrapy.downloadermiddlewares.useragent.UserAgentMiddleware': None,

'scrapy\_user\_agents.middlewares.RandomUserAgentMiddleware': 400,

}

In this method we are trying to access the request on secure web-pages by giving different users agent who got permission to scrap data from secure web-pages.

Like Google and many other companies, who have API for those websites.

2) **Proxy pool method:**

First we have to enable this middleware-PROXY\_POOL\_ENABLE=True in the setting.py

Then add rotating\_proxies middleware to your DOWNLOADER\_MIDDLEWARES:

DOWNLOADER\_MIDDLEWARES = {

# ...

'scrapy\_proxy\_pool.middlewares.ProxyPoolMiddleware': 610,

'scrapy\_proxy\_pool.middlewares.BanDetectionMiddleware': 620,

# ...

}

**After this all request will be proxied using proxies.**

Difficulties (which I have faced):

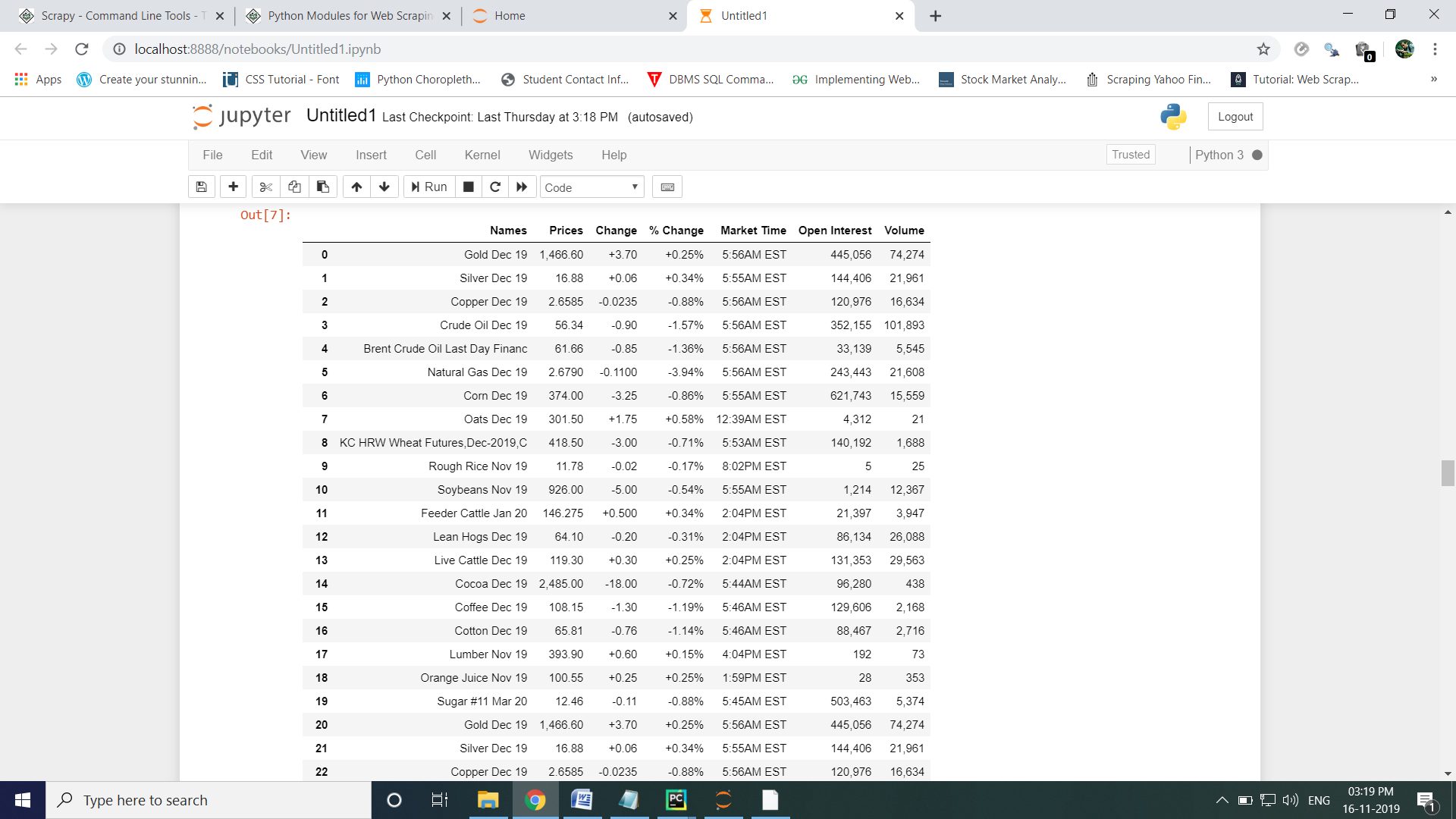
1. The websites uses dynamic element like javascipt/AJAX code, a scraper tool like Beautiful Soup not able to fetch data, but I solve this problem with the help of selenium package.
2. The web scraping problem of web scraping, from a single IP for a long time

when you scrap data. It will recognized and locked, However I solve this problem by using proxy method.

**Advantages:**

1. Imagine how much you spend if you had copy and paste each piece of information but scraping software automates most of the process.
2. It giving accuracy of data extracted.
3. We can you use spreadsheet and database to manage the data easily.

**Output in tabular form:**



Disadvantages:

1. Webmasters tend to change their websites frequently in order to improve their functionality, which can easily break the logic of web scraping software.
2. It is very hard to scrap some websites which are dynamics.

References:

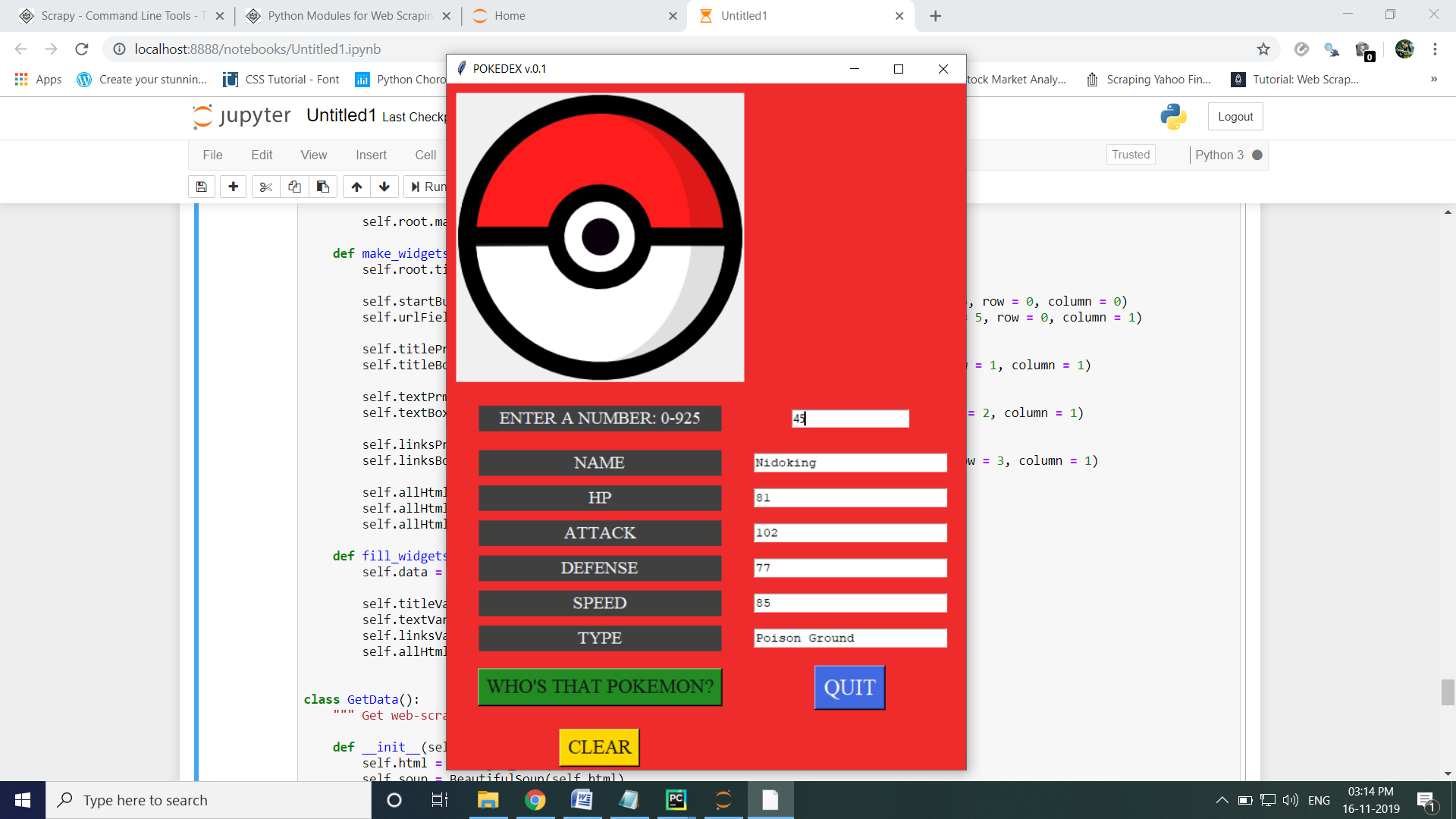
1. **https://towardsdatascience.com/an-introduction-to-web-scraping-with-python-a2601e8619e5**
2. **https://www.datacamp.com/community/tutorials/web-scraping-using-python**
3. **https://hackernoon.com/building-a-web-scraper-from-start-to-finish-bb6b95388184**
4. **https://pypi.python.org/pypi/Scrapy**
5. **http://scrapy.org/**
6. **http://doc.scrapy.org/**
7. **https://www.analyticsvidhya.com/blog/2017/07/web-scraping-in-python-using-scrapy/**
8. **https://medium.com/@asishraz/scraping-data-from-imdb-top-35-movies-using-python-48d1986dc6c9**
9. **https://www.tutorialspoint.com/python\_web\_scraping/index.htm**
10. **https://youtu.be/0\_VZ7NpVw1Y**
11. **https://www.youtube.com/watch?v=ve\_0h4Y8nuI&list=PLhTjy8cBISEqkN-5Ku\_kXG4QW33sxQo0t**

Conclusion:

1. My main goal of doing this project to know how to use web scraping techniques to gather data and display or use it in a meaningful way.
2. I scrapped many different-different websites either it allowing to scrap or not, while scrapping different websites, I faced lot of problems but after understanding the structure of HTML code of websites it is easy to scrap then.
3. I have scrapped Amazon product information which didn't allow scrapping, share marketing stock value websites which is dynamic every second value of stock is changing , here the package selenium help me lot, quotes scraping, the websites of LPU.
4. After getting suggestion from my teacher to make GUI based web scraping.

I have made GUI based web scrapping projects.

1. First project- scrapping the data of Pokemon character by giving input of pokemon token number[from 0 to 995], We will get all detail about that character of pokemon as a output.



b)Second project- scrapping from any URL as a input the links of interconnecting page (href) and of images.

