International Journal of Humanities Education
VOL -13 NO 1 2025 ISSN:2327-0063 | E-ISSN:2327-2457

DYNAMIC WEB SCRAPING THROUGH SELENIUM WEBDRIVER IN PYTHON

Aneesha Gudavalli
M.Tech Scholar, Department of Information Technology, V R Siddhartha Engineering College, Vijayawada, Andhra Pradesh, India, gudavallianeesha@gmail.com

G. JayaLakshmi
Assistant Professor, Department of Information Technology, V R Siddhartha Engineering College, Vijayawada, Andhra Pradesh, India,
,jaya1123@vrsiddahrtha.ac.in

Abstract:
Machine learning is used to power today's technological marvels, such as self-driving cars, picture and speech recognition, and space transportation. To build a stable & dependable machine learning model for such business challenges, Data Science experts would require a large amount of data. The first step in the data science life cycle is data mining or data collection. Depending on the business needs, data might be collected through SAP servers, logs, databases, APIs, online repositories, or the web. Web scraping, often known as "crawling" or "spidering," is a method of extracting information from an online source, most typically a webpage. Crawling the web is one of the methods for extracting large volumes of data in a short amount of time. This paper proposes a way of scraping data from numerous dynamic websites or web pages via the internet using Selenium WebDriver and Python. Selenium is capable of scraping large amounts of data in a short amount of time, such as text and images.
Keywords: data scraping, web scraping, selenium, selenium WebDriver, dynamic webpage scraping.

I. INTRODUCTION

On the internet, there is a significant amount of data that can be used to meet business objectives. To get this information from the web, one requires a tool or technique. This is where the concept of web-scraping comes into play. Web scraping [7] can help us collect large amounts of data about customers, products, people, stock markets, and other topics. Customer buying trends, employee churn, customer attitudes, and other factors are all evaluated, data acquired from websites like as e-commerce portals, job portals, and social media channels can be used. Sites are a huge expanse of information that everybody can get to. Innovation's recent fad has constrained us to adjust our organization rehearses. Data crawling, also known as online scraping or data harvesting, has existed since the beginning of the internet. Although it is now synonymous with web content extraction, it wasn't originally used for that. Selenium is a web application testing framework that is portable. It's free software that runs on Windows, Linux, and Mac OS X and is licensed under the Apache License 2.0. Selenium is also used as a web scraping tool, in addition to its primary purpose. We won't go over all of Selenium's components; instead, we'll concentrate on WebDriver shown in fig1, a single component essential for web scraping. Selenium WebDriver allows us to develop and execute test cases by controlling a web browser through a programming interface. When websites display dynamic content, Selenium comes very helpful. Despite the fact that Scrapy is a sophisticated online scraping technology, it is rendered useless when dealing with dynamic websites.

Fig1: Selenium Webdriver

To get rendered HTML content from dynamic pages, Selenium web drivers are used. Selenium is a Python library that allows you to run a web driver from a remote computer. To browse to the given data source, the selenium package's driver.get method is used. The drivers (such as ChromeDriver and FirefoxDriver) web browsers are directed to deliver commands in question and receive the response. To receive reliable information, user input is sometimes required, such as clicking buttons and picking relevant alternatives from dropdown menus, which Selenium supports effectively.

SELENIUM SUITE AND WEBDRIVER ARCHITECTURE

Selenium is an open source program automation tool that is commonly used for web application testing. It automates cyclic tasks by computerizing the control of an internet browser. Selenium is a collection of testing tools that may be used to test a variety of programs, frameworks, and languages such as C#, Java, Ruby, and Python. Figure 2 depicts the four-part selenium system. Selenium IDE, for example, is a Firefox add-on that enables you record and replay tests. Selenium Webdriver, the third component, offers APIs in a range of languages, giving you greater control and allowing you to follow best practices in software development.

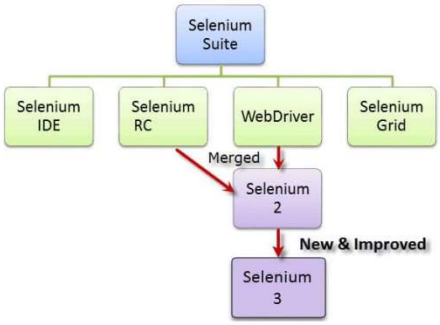


Fig2: Selenium Suite

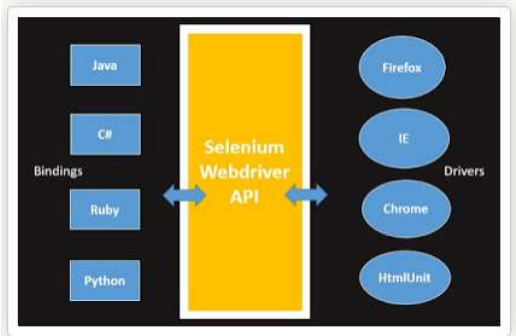


Fig3: selenium WebDriver architecture

WebDriver was designed by Simon Stewart in 2006, at a time when JavaScript tools like Selenium Core were becoming increasingly powerful and restrictive in browsers and web apps. It was the first browser-controlling cross-platform testing framework. To build a programming interface that is more intuitive and concise. It supports [2] dynamic web pages, which allow elements of a page to change without requiring the entire page to be refreshed. The Web Driver (shown in fig.3) .

II. LITERATURE SURVEY

Web scraping is a technique for obtaining large volumes of data from a variety of websites. This information can then be saved in a file or database. Web scraping is the process of extracting a web page's HTML code. Selenium [1] is utilized to achieve this goal. Selenium is an automated testing tool that is commonly used for web browser automation. It can, however, be used to scrape the web as well. The URLs of websites connected to the search query are acquired using the Selenium web driver.

Following that, each URL is automatically opened, and the data on each webpage is scraped one by one.

Selenium WebDriver settles on direct decisions to the program, exploiting every program's intrinsic robotization usefulness. Since there are such countless programs and programming dialects, WebDriver API[3] gives a standard determination. The expression "far off Webdriver" alludes to an API that should be carried out by every program. The language ties will provide orders to the normal driver API, and on the opposite end, a driver will tune in for those orders and execute them in the program utilizing far off WebDriver, returning the outcome/reaction by means of API to the code/Binding.

An inquiry demand is made to the Twitter Search[4] endpoint, which then, at that point makes a URL, utilizing any legitimate arrangement of HTTP-HEADERS, a variety of words (otherwise called looking through terms), a variety of dates (date range), and discretionary contentions.

A web bug measures a delivered HTTP reaction, and the HTML payload is sent to a download layer. At long last, natural information including tweets is provided into the Scrapy motor, which uses label selectors to extricate hypertext labels; each tweet is dealt with as an individual design comprised of plain content, the date it was made, and geographic information (if exists). Reactions are scratched utilizing a greatest position class quality from a div > HTML tag; this data is a pagination identifier from the most as of late appended tweets.

Web scratching of plans and fixings to acquire plans dependent on a specific fixing, which helps the culinary specialist in managing an element of interest, picking dishes dependent on fixings, or adhering to an eating routine arrangement dependent on the components utilized in plans. The data assembled is saved in the MongoDB data set. The interaction of web scratching [5] is utilized to move web information into a predefined storable arrangement. The culinary specialist can plan an eating regimen plan for the customers dependent on the fixing data or prescribe whether to utilize a specific thing in the formula dependent on the clients' inclinations or wellbeing concerns.

A toolset using cloud-based web scratching [6] to remove, refine, unite, and store COVID-19 cases data at different scales for each and every available country all through the planet normally. Changing the format crawler unit to meet the specific prerequisites of every information source. Testing and guaranteeing that simply the information expected from the objective information source is gathered.

The python-based web crawler [8] scrapy may likewise help us in recovering the ideal outcome, as we dissect the cycle utilizing explicit code and give the required url to the emphasis to scratch the information from the source url. The undertaking's essential web slithering content, which shows the information scratched and put away in the data set of items from an interpersonal interaction webpage, for this situation.

The initial step in this module is to scrape [9] Google results for a given string. Selenium, a portable web application testing framework, is used to achieve this goal. On the basis of a list compiled by us, stratifying the connections is in charge of identifying the links as reliable or unreliable. Extracting the links, on the other hand, entails scraping the content of each reliable link. We do it by utilizing the Python-based Scrappy framework.

Web scraping technology was utilized to obtain weather data [10] from websites in the South Sumatera area. Online scraping is a technique for retrieving individual elements from a web page. The information acquired through this web scraping technique will be saved in a database or data warehouse, where it can be used for future research into weather forecast data mining in South Sumatra, as well as the creation of weather-based decision-making applications.

II. PROPOSED SYSTEM

The proposed a method which can scrap information from dynamic website pages utilizing selenium WebDriver in python. During the scratching interaction, any client activity on a program window can interfere with the stream and can cause a sudden conduct. Thus, for scratching applications, it is essential to stay away from any outer reliance while making applications, for example, browser. To overcome such issues Headless programs is used, which can work without showing any graphical UI and furthermore permits applications to be a solitary wellspring of cooperation for clients and gives a smooth client experience.

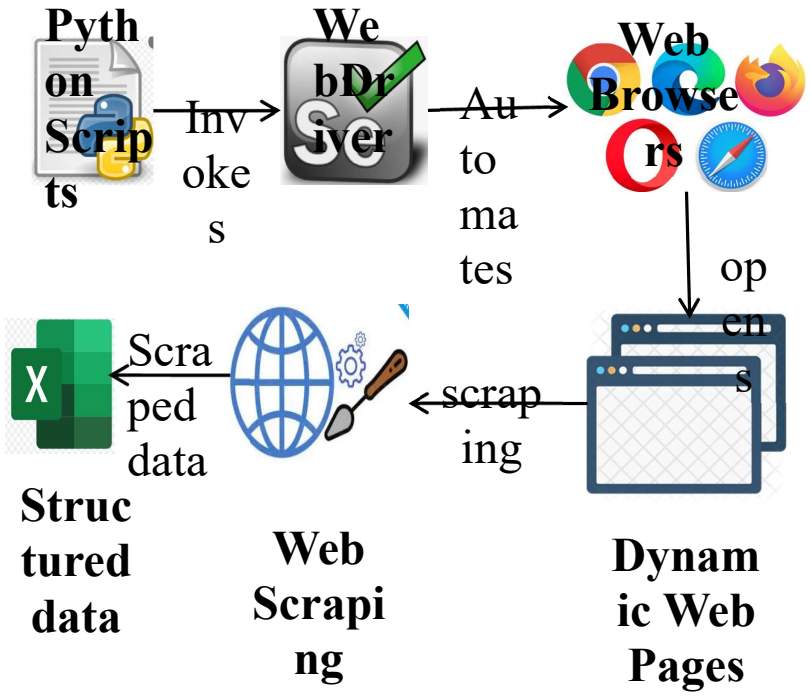


Fig4: Architecture diagram

As shown in the fig4, The webdriver will be executed by the scripts. The webdriver will automate the web browsers and open dynamic web pages that are directed to open, then scrape the data from those web pages and save it in an excel sheet or database. Selenium WebDriver is a popular technology for automating web user interfaces. It enables the automatic execution of online browser window tasks such as travelling to a website, filling out forms (including dealing with text boxes, radio buttons, and drop-downs), submitting the forms, browsing around web pages, and handling pop-ups, among others.

VOL -13 NO 1 2025 ISSN:2327-0063 | E-ISSN:2327-2457

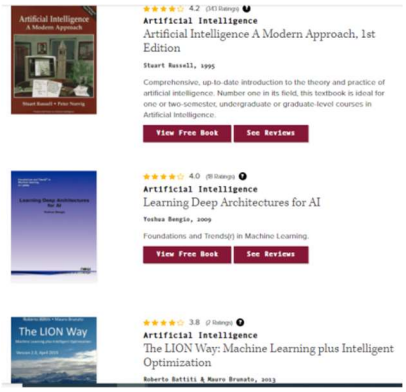


Fig5: Image of website that is been scraped

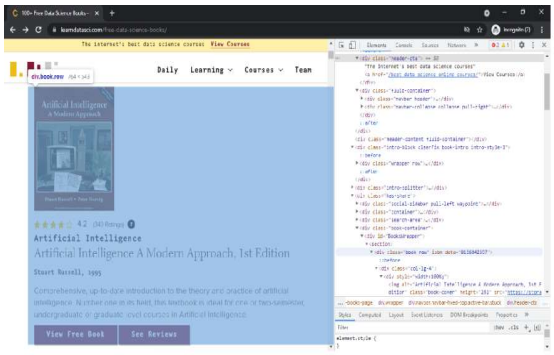


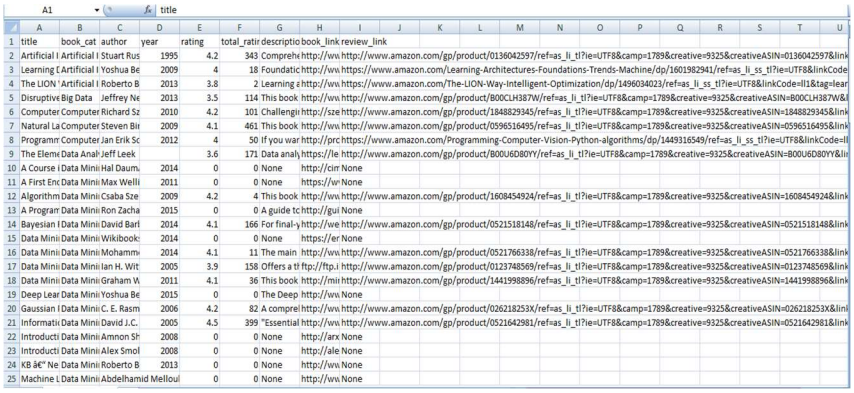
Fig6: Image of inspected code of one book.

Locating the Elements: The most crucial aspect of web scraping is finding the elements. A class, id, tag, name, or xpath can be used to identify elements on a web page. Classes are not unique, but ids are. This means that a class can identify several web elements, whereas an id can only identify one element. Any of the methods listed below can be used to identify an HTML element.**driver.find_element_by_id,driver.find_element_by_name,driver.find_element_by_xpath,driver.find_element_by_tag_name,driver.find_element_by_class_name**

Any of the following can be used to identify multiple HTML elements.**driver.find_elements_by_name,driver.find_elements_by_xpath,driver.find_elements_by_tag_name,driver.find_elements_by_class_name**

Scratch subtleties for each book on the page. Each page as 20 books. The subtleties of each book can be found by utilizing the URL on each card. Along these lines, to get the book subtleties we need this links. Scrape books in every single page. This implies that we will have a circle to scratch each book in the page and another to repeat through pages. Moving starting with one page then onto the next includes a change of the URL such that it is unimportant to anticipate a connection to any page.Fig6 shows the HTML code for the highlighted region after inspecting the site. To inspect the code right click on the particular element of the webpage. To access to href, we must go via the following hierarchy: class= “product prod” > h3 tag > a tag and the get value of the href property. In fact, all books on all pages are of the same product prod class and include the article tag.

III. Results and Conclusion



	title	book_cat	author	year	rating	total_ratir	descriptio	book_link	review_link
2	Artificial I	Artificial I	Stuart Rus	1995	4.2	343	Comprehe	http://ww	http://www.amazon.com/gp/product/0136042597/ref=as_li_tl?ie=UTF8&camp=1789&creative=9325&creativeASIN=0136042597&link
3	Learning D	Artificial I	Yoshua Be	2009	4	18	Foundatic	http://ww	https://www.amazon.com/Learning-Architectures-Foundations-Trends-Machine/dp/1601982941/ref=as_li_ss_tl?ie=UTF8&linkCode
4	The LION V	Artificial I	Roberto B	2013	3.8	2	Learning a	http://ww	https://www.amazon.com/The-LION-Way-Intelligent-Optimization/dp/1496034023/ref=as_li_ss_tl?ie=UTF8&linkCode=ll1&tag=lear
5	Disruptive	Big Data	Jeffrey Ne	2013	3.5	114	This book	http://ww	http://www.amazon.com/gp/product/B00CLH387W/ref=as_li_tl?ie=UTF8&camp=1789&creative=9325&creativeASIN=B00CLH387W&l
6	Computer	Computer	Richard Sz	2010	4.2	101	Challengi	http://sze	http://www.amazon.com/gp/product/1848829345/ref=as_li_tl?ie=UTF8&camp=1789&creative=9325&creativeASIN=1848829345&link
7	Natural La	Computer	Steven Bi	2009	4.1	461	This book	http://ww	http://www.amazon.com/gp/product/0596516495/ref=as_li_tl?ie=UTF8&camp=1789&creative=9325&creativeASIN=0596516495&link
8	Programm	Computer	Jan Erik So	2012	4	50	If you war	http://prc	https://www.amazon.com/Programming-Computer-Vision-Python-algorithms/dp/1449316549/ref=as_li_ss_tl?ie=UTF8&linkCode=ll
9	The Eleme	Data Anal	Jeff Leek		3.6	171	Data anal	https://le	http://www.amazon.com/gp/product/B00U6D80YY/ref=as_li_tl?ie=UTF8&camp=1789&creative=9325&creativeASIN=B00U6D80YY&li
10	A Course i	Data Minii	Hal Daum	2014	0	0	None	http://cir	None
11	A First Enc	Data Minii	Max Welli	2011	0	0	None	https://w	None
12	Algorithm	Data Minii	Csaba Sze	2009	4.2	4	This book	http://ww	http://www.amazon.com/gp/product/1608454924/ref=as_li_tl?ie=UTF8&camp=1789&creative=9325&creativeASIN=1608454924&link
13	A Program	Data Minii	Ron Zacha	2015	0	0	A guide tc	http://gui	None
14	Bayesian I	Data Minii	David Barl	2014	4.1	166	For final-y	http://we	http://www.amazon.com/gp/product/0521518148/ref=as_li_tl?ie=UTF8&camp=1789&creative=9325&creativeASIN=0521518148&link
15	Data Minii	Data Minii	Wikibooks	2014	0	0	None	https://er	None
16	Data Minii	Data Minii	Mohamm	2014	4.1	11	The main	http://ww	http://www.amazon.com/gp/product/0521766338/ref=as_li_tl?ie=UTF8&camp=1789&creative=9325&creativeASIN=0521766338&link
17	Data Minii	Data Minii	Ian H. Wit	2005	3.9	158	Offers a th	ftp://ftp.i	http://www.amazon.com/gp/product/0123748569/ref=as_li_tl?ie=UTF8&camp=1789&creative=9325&creativeASIN=0123748569&link
18	Data Minii	Data Minii	Graham W	2011	4.1	36	This book	http://mir	http://www.amazon.com/gp/product/1441998896/ref=as_li_tl?ie=UTF8&camp=1789&creative=9325&creativeASIN=1441998896&link
19	Deep Lear	Data Minii	Yoshua Be	2015	0	0	The Deep	http://ww	None
20	Gaussian I	Data Minii	C. E. Rasm	2006	4.2	82	A compre	http://ww	http://www.amazon.com/gp/product/026218253X/ref=as_li_tl?ie=UTF8&camp=1789&creative=9325&creativeASIN=026218253X&link
21	Informatio	Data Minii	David J.C.	2005	4.5	399	"Essential	http://ww	http://www.amazon.com/gp/product/0521642981/ref=as_li_tl?ie=UTF8&camp=1789&creative=9325&creativeASIN=0521642981&link
22	Introducti	Data Minii	Amnon Sh	2008	0	0	None	http://arx	None
23	Introducti	Data Minii	Alex Smol	2008	0	0	None	http://ale	None
24	KB â€" Ne	Data Minii	Roberto B	2013	0	0	None	http://ww	None
25	Machine L	Data Minii	Abdelhamid	Mellou	0	0	None	http://ww	None

Fig7: Scraped data saved in an excel sheet.

VOL -13 NO 1 2025 ISSN:2327-0063 | E-ISSN:2327-2457

Exploratory Data Analysis is the crucial process of using summary statistics and graphical representations to undertake preliminary investigations on data in order to uncover patterns, spot anomalies, test hypotheses, and verify assumptions. Results obtained after EDA are as follows

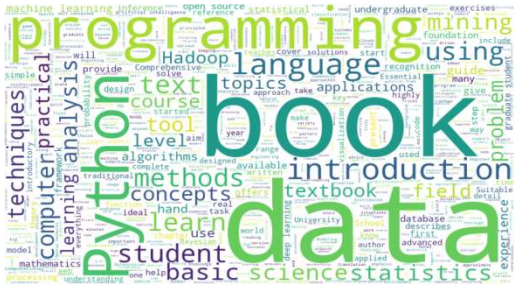


Fig8: Word cloud for book descriptions

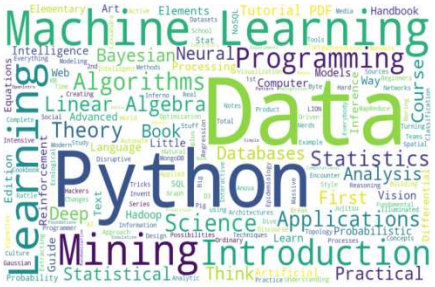


Fig9: Word cloud for book titles

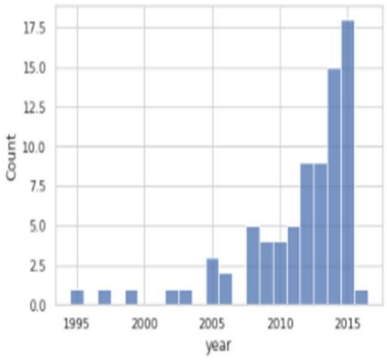


Fig10: Histogram of year.

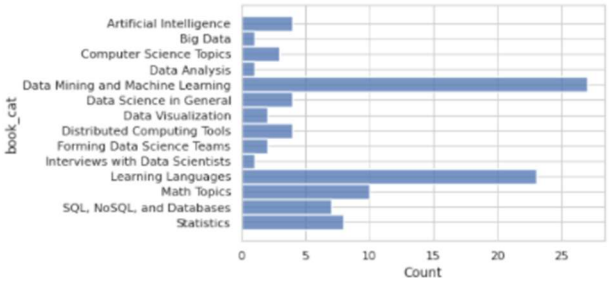


Fig11: Histogram for book category.

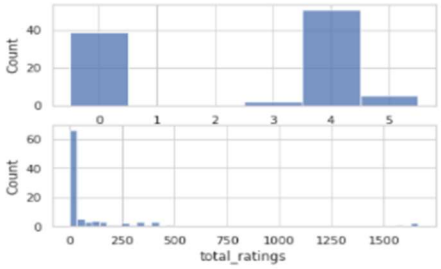


Fig12: plot for rating and total rating.

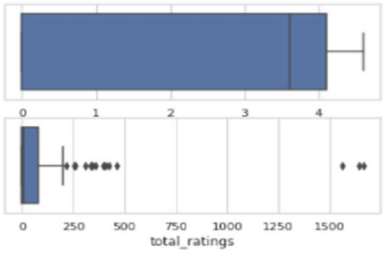


Fig13: Box plot for rating and total rating and outliers.

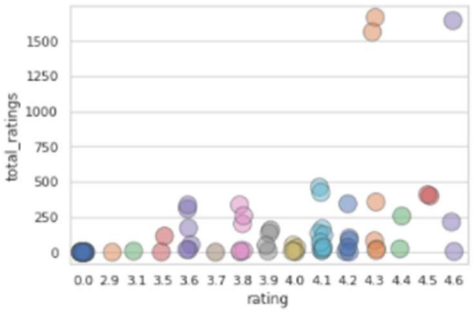


Fig14: strip plot of rating and total ratings

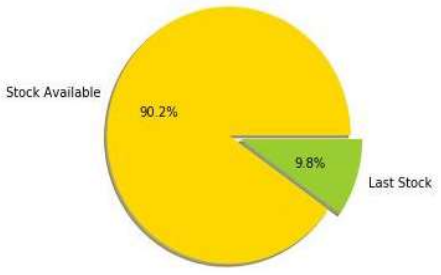


Fig15: Pie chart show stock availability.

Web scraping is a common method of gathering information from the internet. Because many websites have distinct layout, there isn't a single scraper that can be used on all of them. The most important

skill is a thorough understanding of online scraping, which includes knowing how to locate web items and recognizing and dealing with issues, This work has covered all those techniques to scrape the dynamic web pages easily and fastly.

REFERENCES

[1] K Usha Manjari, Syed Rousha, Dasi Sumanth, Dr. J Sirisha Devi," Extractive Text Summarization from Web pages using Selenium and TF-IDF algorithm", Proceedings of the Fourth International Conference on Trends in Electronics and Informatics (ICOEI 2020), IEEE Xplore Part Number: CFP20J32-ART; ISBN: 978-1-7281-5518-0.

[2] Revathi.K1 , Prof.V.Janani," SELENIUM TEST AUTOMATION FRAMEWORK IN ON-LINE BASED APPLICATION", International Journal of Advance Research In Science And Engineering, IJARSE, Vol. No.4, Special Issue (02), February 2015.

[3] Paruchuri Ramya, Vemuri Sindhura, P Vidya Sagar," Testing using Selenium Web Driver", IEEE,2017.

[4] A. Hernandez-Suarez,G. Sanchez-Perez, K. Toscano-Medina, V. Martinez-Hernandez, V. Sanchez and H. Perez-Meana ," A Web Scraping Methodology for Bypassing Twitter API Restrictions", arXiv:1803.09875v1 [cs.IR] 27 Mar 2018.

[5] Shilpa Chaudhari, Aparna R., Vinay G Tekkur, Pavan G L., and Shreekanth R Karki," Ingredient/Recipe Algorithm using Web Mining and Web Scraping for Smart Chef", 978-1-7281-6828-9/20/$31.00 ©2020 IEEE.

[6] HAI LAN, DEXUAN SHA,ANUSHA SRIRENGANATHAN MALARVIZHI,YI LIU, YUN LI, NADINE MEISTER, QIAN LIU , ZIFU WANG, JINGCHAO YANG, AND CHAOWEI PHIL YANG ," COVID-Scraper: An Open-Source Toolset for Automatically Scraping and Processing Global Multi-Scale Spatiotemporal COVID-19 Records", VOLUME 9,IEEE Access, 2021.

[7] Kasereka Henrys," Importance of web scraping in e-commerce and e-marketing", HKCorporation IT Official Journal January 2021.

[8]David Mathew Thomas, Sandeep Mathur," Data Analysis by Web Scraping using Python", Proceedings of the Third International Conference on Electronics Communication and Aerospace Technology [ICECA 2019] IEEE Conference Record # 45616; IEEE Xplore ISBN: 978-1-7281-0167-5.

[9] Dinesh Kumar Vishwakarma , Deepika Varshney, Ashima Yadav," Detection and veracity analysis of fake news via scrapping and authenticating the web search", 1389-0417/ 2019 Elsevier B.V.

[10] Fatmasari, Yesi Novaria Kunang, Susan Dian Purnamasari," Web Scraping Techniques to Collect Weather Data in South Sumatera",international conference on electrical engineering and computer science(ICECOS) 2018.