International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249 – 8958 (Online), Volume-9 Issue-5, June 2020

Web Scraping in Finance using Python

Siddhant Vinayak Chanda, Arivoli A

Abstract: ***The objective of this paper is to highlight different ways to extract financial data (Balance Sheet, Income Statement and Cash Flow) of different companies from Yahoo finance and present an elaborate model to provide an economical, reliable and, a time-efficient tool for this purpose. It aims at aiding business analysts who are not well versed with coding but need quantitative outputs to analyse, predict, and make market decisions, by automating the process of generation of financial data. A python model is used, which scrapes the required data from Yahoo finance and presents it in a precise and concise manner in the form of an Excel sheet. A web application is build using python with a minimalistic and simple User Interface to facilitate this process. This proposed method not only removes any chances of human error caused due to manual extraction of data but also improves the overall productivity of analysts by drastically reducing the time it takes to generate the data and thus saves a substantial amount of human hours for the consumer. We also discuss the importance of data mining and scraping technologies in the finance industry, different methods of scraping online data, and the legal aspect of web scraping which is highly dependent on generated data to analyse and make decisions.***

Keywords : Web Information extraction, Web Data Mining, Information Retrieval, Web Scraping, Financial Statements Analysis.

I. INTRODUCTION

Economic globalisation and advancement of information technology have in current times accounted for a massive increase in the volume of financial data being generated and accumulated at an unprecedented pace. Companies are becoming more data-driven and an increasing number of cases are coming up where they need to access and process data outside of their internal data sources – including publicly available data, data provided by data brokers, data stemming from APIs, etc. Efficient and effective extraction and utilisation of this huge amount of financial data available are of almost importance and by implementing automated data generating techniques, their analysis, and financial modeling, the companies can enhance their strategic planning, investment, risk management, and other critical decision-making processes. Web Scraping and crawling are integral processes for the automatic generation of relevant data. It is the process of extracting relevant data from the World Wide Web(the internet as we know it) and store it in a database for analysis or retrieval at a later stage. The information on the web is commonly scrapped by the Hypertext Transfer Protocol (HTTP) or through an internet browser, either manually by the user or automatically through bots or crawlers. As an extensive amount of heterogeneous information is continuously being generated on the internet, web scraping techniques are popularly recognised as an effective and powerful technique for storing this information Web-Crawling, on the other hand, is executed in a different manner for a different purpose.

Figure 1 illustrates both activities. As evident, we can notice that web crawling does not have a particularly defined target and processes any available data without aiming at specific information whereas web scraper extracts, processes, and parses the data from a specified source. Our discussion in this paper will be limited to web scraping.

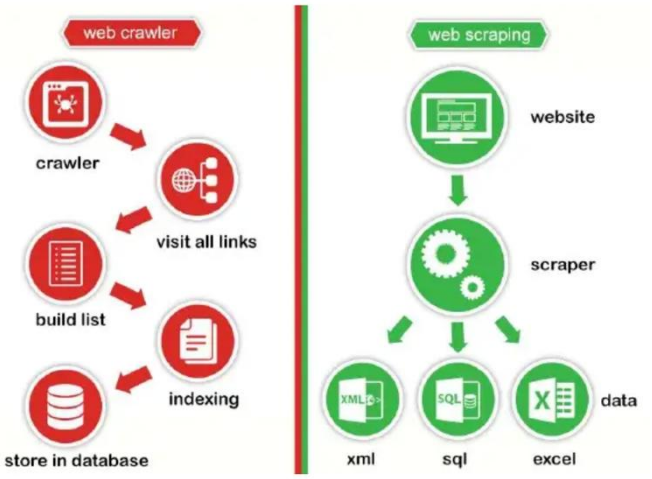

Figure 1. Web Crawler vs Scraper

Analysts working in accounting firms, investment banks, and other financial sectors rely heavily on the financial statements (*balance sheet, income statement and cash flow*) of companies as they contain noteworthy data regarding the financial health of the company, using which financial analysis of businesses are performed to analyse whether that business is stable and profitable enough to make a monetary investment. Even at an individualist level, these documents are important for making investment strategies, comparing different sectors, and learning more about the finances of the company. Such information is spread over the internet in various publicly available websites and thus manual extraction of the data is a tedious and time-consuming process. Our tool aims to solve this specific problem. In this paper, we propose an efficient and economical technique for automating the process of extraction of financial statements of any given number of companies from Yahoo finance in an articulate manner according to the user needs using python. The model aims to build an online

Revised Manuscript Received on May 15, 2020.
* Correspondence Author
Siddhant Vinayak Chanda*, School of Computer Science and Engineering, Vellore Institute of Technology, Vellore.
Arivoli A, School of Computer Science and Engineering, Vellore Institute of Technology, Vellore.

Web application using python with a minimalistic user interface and provides the user with several functionalities. In the future, we aim to host it on a web server that will provide access to anyone who wants to use it free of charge and make it open source so that others can add their own functionality to it.

There are a wide range of applications to integrate data extraction techniques in the finance sector. Our main motivation behind this project is:

1. Time: Manual extraction of data from websites is a tedious and time-consuming process. It can take several hours to a few days to collect all the information that we need. Manual extraction requires a user to search the name of the company on Yahoo finance, navigate the page to find the financial documents, and copy all the data points for each of the three documents to store it. This process needs to be repeated for each company. In our proposed model, we reduce this time to a few minutes. The user needs to upload the name of the companies for which the data needs to be generated and the tool automatically scrapes all the data and provides it articulately in an Excel sheet.
2. Human Error: There is a high probability of coping down the wrong data when extracting it manually. These kinds of potential human errors are removed in our model of automated extraction
3. Enhanced Productivity: Our proposed model not only removes the chances of any human error but also drastically reduces the time it takes to generate the data thereby enhancing the productivity of the user.

II. LITERATURE SURVEY

The internet is a repository of a massive amount of data. This data is organized on numerous websites and can be found in the basic structure of the HTML page. Gathering this data for relevant information can be a tedious task, but is necessary for further analyses. In this paper, we discuss the various processes that go into scraping the web for information and all the different data structures that make it easier for us to navigate through a web page. [1] The internet is an extensive, data-rich, and easily accessible source of information to the world, with its userbase increasing every day. We use search engines to navigate through all this information available and to return the relevant information as per the user's needs. These search engines cannot traverse through all the information available on the web at once and hence they make use of web crawlers to make this process more efficient[2] This paper discusses the benefits and functionality of a data mining powered search engine to revolutionize the education industry. Gathering necessary information about different schools and colleges from the web is a huge task and a lot of data can be missed due to the lack of a good search engine optimizer. Here they build a web based service for making the process of admission and decision making easier by crawling the web and analysing the data generated.[3] Data mining is the process of traversing the web for information. Scraping the web is one of the significant processes involved in this. The majority of the studies conducted on scraping have been related to the process of automating the extraction. In this paper, we discuss and study how Document Object Model parsing technique functions and its efficiency and effectiveness[4] Web usage has expanded a lot lately and the users can navigate and discover different data by utilizing various hypertext links. This process of navigating the web has given rise to the concept of web crawlers, which act like search engines and make this navigation easier. Different uses and applications of web crawlers are discussed in this paper. [5] This paper is about the few types of crawlers utilized in web mining and their processes. Focused crawlers, an important crawler for navigating and finding specific data from the web is discussed along with other strategies and processes to remove the irrelevant information and ways to increase the efficiency and speed up the process [6] In this paper, the author proposes a data logger application to manage the information collected from an online water monitoring system. We look into the functional requirements of the app and its dependencies. Python is used as the primary language for development purposes and web scraping techniques are incorporated[7] This paper discusses the coding behind extraction of information from the Securities and Exchange Commission(SEC) files. The author has used python for development purpose and the program crawls the web to return different URLs. A series of functions are performed on the data including text analysis and word count analysis.[8] The paper presents a technique to scrap web information from specific links, extract the desired data from them, and return it in a CSV file. Scrapy, an open-source framework utilising python, is used for the development of this model. Extraction of data from famous E-Commerce websites are performed to extract the ratings, comments, and price of various products and analyse them.[9] This paper extracts the data contained in the financial statements of various companies available on the internet and utilises them to perform a series of functions and build a working model focusing on analyzing all the data generated. It also discusses the existing models for financial analysis and the applications of data mining in the finance sector[10] Information is extensively retrieved from the internet for research and academic purposes. In this entire process, a factor that is often overshadowed is the legal aspect pertaining to web scraping. In this paper, the author discusses the various areas where scraping can be deemed illegal and the legal systems in place to keep web scraping in check. The ethical viewpoint that should be followed while mining the web is explained along with privacy policies and the various challenges. [11]

III. DIFFERENT WAYS OF SCRAPING

Over time with the advancement of technology and digitisation, web scraping techniques have also evolved. IDC insights forecast that by the year 2025, the global data sphere will grow by 61% to 175 zettabytes(1 Zettabyte = 1 Trillion Terra-byte), which is a staggering 10 times the data generated in 2016. With this exponential growth of unorganised data, more sophisticated and efficient scraping software's and techniques keep coming up in the market. In this paper, we will discuss the most common techniques and the easily available software for this purpose.

1. Manual Extraction: Probably the simplest technique used by the majority of the people. Manual scraping is helpful when we need to scrape minimal data with no repetition in the process.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Usually, it is preferred when the cost and time of setting up automated extraction are more than that required to achieve it manually. It is also helpful in cases where websites do not allow automatic extraction of data.

2. HTML Parsing: Most websites are primarily built on HTML. The HTML pages that contain all the relevant information are shown in the server as a result of a query by the user. Here, the HTML page code is analysed to extract all the relevant information necessary, from the title, heading, paragraphs, etc. Since the basic structure of HTML remains the same, different programming languages can be used to write programs that can parse multiple pages for information.

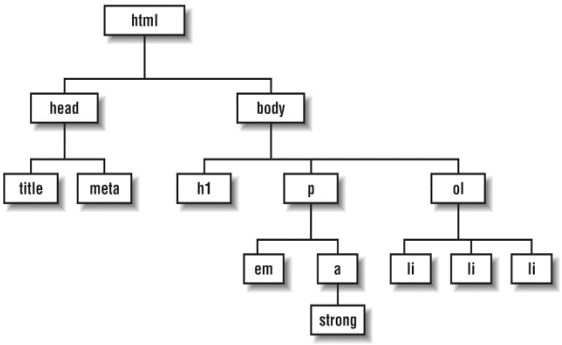

Figure 2. HTML Structure

3. DOM Parsing: Document Object Model Parsing is an advanced method of HTML parsing, in which user-generated scripts parse the contents of a web page into a Document Object Model tree, to dynamically modify and inspect the web page. By incorporating the program into the internet browser, we can recover the data from the tree making navigation through the page easier and enhancing our possibilities to address specific parts of the web page. This method is extensively used in CSS (Cascading Style Sheet) and JavaScript.

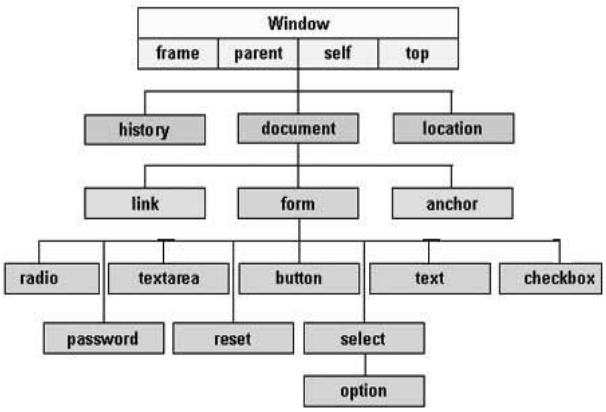

Figure 3. Document Object Model Tree

4. XPath: XML Path Language is a query language for determining nodes from an XML document. and are used to easily navigate through elements and attributes of a XML document. It is also extensively used for computing values (e.g., numbers, Boolean values, or string) from the content of an XML document thereby is convenient in scraping numeric data.

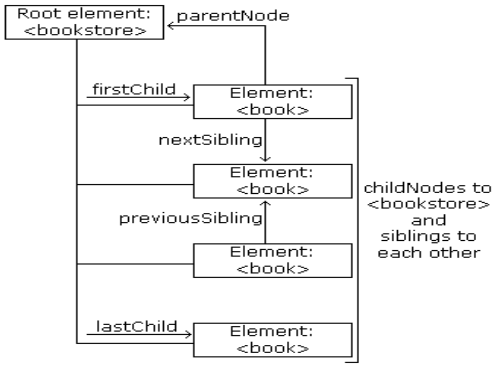

Figure 4. XPath Navigation

5. APIs: An Application Programming Interface is an interface that defines interactions between multiple software intermediaries, i.e. allows one software to talk to another. They are widely used to extract data from websites and are usually available for free online. The free API will let you send around ten to a hundred requests per day and thus are extensively used for this purpose.

6. Open Source Frameworks: It is not necessary to start from scratch when scraping data. Developers can benefit from open source web scraping tools which allow them to save time by using pre-defined generic modules and working on them for specific needs. Some of the best open-source tools available in the market are:

Frame work	**Comparison data points**		
	Language	***Data Format***	***Suitable for broad crawling***
Scrapy	Python 2.7+	CSV,JSON, XML	Yes
PySpider	Python 2.7+	CSV,JSON	No
Portia	Docker	CSV,JSON, XML	No
ApifySDK	Node.js	CSV,HTM. EXCEL	Yes
Selenium	Python 2.7+	Custom	No
Puppeteer	Node v6.4	JSON	No
Heritirix	Java 5+	ARC file	Yes

Table 1. Comparison b/w different open source software

IV. LEGAL ASPECT OF WEB SCRAPING

With a tremendous volume, velocity, and variety of data available on the internet, manual extraction and analysis of this data are is next to impossible. This is the reason we have innumerable tools and scraping technologies present in the market. However, between scraping such data for collection and performing analysis on it, one of the key factors that are overshadowed is the legality of the entire process.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

From a legal standpoint, whether scraping of data is permitted or not is still a grey area. In most cases, it depends on how the scraping is being executed and for what purpose. We will discuss the instances where scraping becomes illegal.

Copyright Material	Scraping, storing and publishing data owned and copyrighted by a website can result in a 'copyright infringement' claim from the owner. Original literature, artistic creations, and original photography can be few examples for this.
Protected Data	Scraping confidential and protected data from a website without the permission of the owner counts as fraud. Any individual should collect 'personal data' without proper authorization and permission
Damage	The user should not overload the website server and intentionally crash it while scraping data. In such a case, the user will be held liable for the damage cost if it can be proven that he violated the terms and conditions of the website.
Trademark infringement	A user cannot scrape and reproduce a registered website trademark as its own. He cannot copy the business name or description that may potentially mislead the public into believing that the business belongs to the website owner.

Table 2. The legal aspect of scraping

As such, website scraping is used extensively for research and various other purposes and most websites allow scraping of their data but the user should be aware of the legal and ethical requirements before collection and using such data. The tools available should not be used to fraudulently collect and store any 'personal information' and the user must make sure he is meeting all necessary compliances and local data protection laws. Along with this, the user should not scrape and republish any content that is copyrighted without permission from its owner.

V. OVERVIEW OF THE PROPOSED MODEL

This tool has been developed on python using a flask framework to build a web application for aiding business analysts with no prior knowledge of coding . The end goal of this system is to automatically extract financial statements of any given number of companies from Yahoo finance and present that information in an Excel sheet.

A. BASIC MODEL

Our first step is to navigate and inspect the webpage to search for the information we need and then extract it.
The main libraries we will use for this purpose are:

- **BeautifulSoup4:** A library in python used for getting information from a XML, HTML, and other mark-up languages. It is helpful in extracting names, URLs, etc. from the HTML page and makes reading and storing data easier. After importing it at the starting we import another library Urlib.request
- **Urlib.request:** An open-source python library that can be utilised to open URLs and parse content from that webpage.
- **Request** library for handling the interaction with the web page (Using HTTP requests).

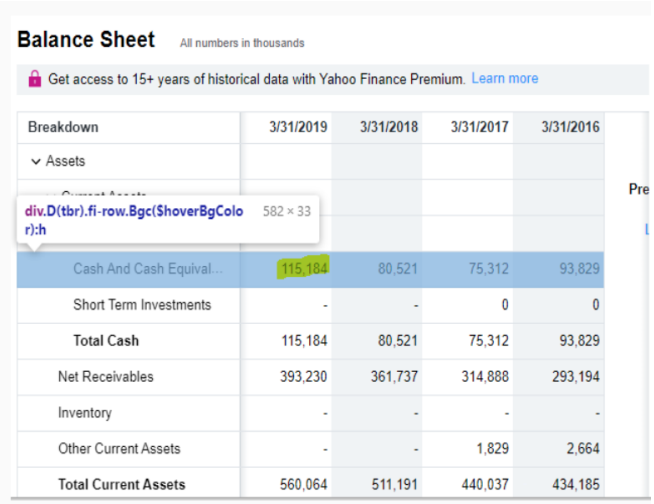

Fig 5. Yahoo Finance page

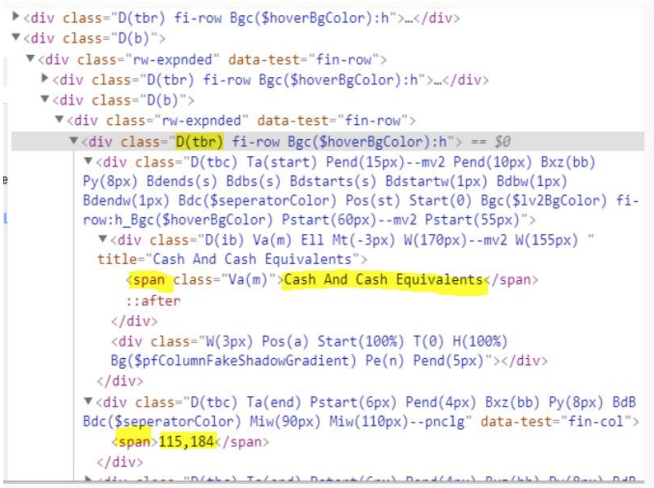

Fig 6. Inspect element on the page

While inspecting the page we find the data points that are required for the purpose of scraping and write our code to scrape these data points.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

```
def parse_rows(table_rows):
    parsed_rows = []
    currency = table_rows[0].xpath("./text()[1]")[0]
    currency = currency.split('Currency in ')[1]
    parsed_rows.append(['{}'.format(currency)])
    for table_row in table_rows[1:]:
        parsed_row = []
        el = table_row.xpath("./div")

        none_count = 0

        for rs in el:
            try:
                (text,) = rs.xpath('.//span/text()[1]')
                parsed_row.append(text)
            except ValueError:
                parsed_row.append(np.NaN)
                none_count += 1

        if (none_count < 4):
            parsed_rows.append(parsed_row)


    return parsed_rows
```

```
def scrape_table(url):
    page = get_page(url)
    tree = html.fromstring(page.content)
    table_rows = tree.xpath("//span[contains(text(),
   'Currency in')] | //div[contains(@class, 'D(tbr)')]")
    assert len(table_rows) > 0

    return parse_rows(table_rows)
def text(elt):
    return elt.text_content().replace(u'\xa0', u' ')
```

Fig 7. Code Snippet

Once the code is executed, we get the generated data which after cleaning and manipulation is presented in an articulate manner as shown below.

	Date	Cash And Cash Equivalents	Short Term Investments	Total Cash	Net Receivables	Other Current Assets	Total Current Assets
1	3/31/2019	115184.0	NaN	115184.0	393230.0	NaN	560064.0
2	3/31/2018	80521.0	NaN	80521.0	361737.0	NaN	511191.0
3	3/31/2017	75312.0	0.0	75312.0	314888.0	1829.0	440037.0
4	3/31/2016	93829.0	0.0	93829.0	293194.0	2664.0	434185.0

Fig 8. A snippet of extracted data

Here, we scraped the Balance Sheet for the company we wanted using python. Following a similar procedure, we can scrape Cash Flow and Income statement as well, and download it in an excel sheet for further analysis.

This is the basic idea of our proposed model, to automate the process of extracting the data available.

The problem with this existing approach is that it extracts information of only one company at a time, and every time a user needs to use it they would have to execute the program and get the data. Even though this is faster than manual extraction, it is still a tedious process and takes a substantial amount of time to get data for multiple companies. Moreover, the model is not at all user friendly and has minimal functionality. Since our end goal is to make it user friendly for people who have no knowledge of coding and provide them with an economical, effective and efficient way of automating this process, we propose a new modified model.

B. PROPOSED MODEL WITH USER INTERFACE

In this section, we will discuss how our tool makes extraction more efficient and productive. The user interface is a Flask-Based Web application build using Python. The front end is being hosted on the local server 8000 running on the device because there is no central server host for the app. The backend coding is done on python and the webpages are designed using HTML and CSS.

The main features of the proposed system are as follows:

1. The user can input an Excel/CSV file with the names of any given number of companies for which the data needs to be extracted. Hence, the user can get information about as many companies as he wants in one go.
2. The tool reads each row of the given excel file to generate company names. Each name is then automatically searched on Yahoo finance to generate its ticker symbol and extract its information.
3. A lot of companies give multiple search results with the same name (the difference is in their ticker symbol). This poses a problem for the program as it cannot automatically decide which company to choose. Therefore, a functionality to let the user choose from the options is provided.
4. These options are displayed right after the user inputs his excel sheet with the company name. Only two companies are displayed on one page. The user can choose among the options by pressing radio buttons in front of them.
5. A key feature of this functionality is that, as soon as the user chooses his options and proceeds to the next page, simultaneous extraction of data begins. The program does not wait for the user to select all the options. This is achieved through threading, and it saves the user a lot of time by parallelly running the application in the background.
6. After the user has selected all the companies, the tool displays the list of companies that could not be found, giving the user an option to download the list or to move to the next page.
7. The last page shows the user list of companies that were found with an option to download the excel sheet generated.
8. To download, the user can choose his own directory and save the file there.

Thus, the generation of data of any number of companies, which might have taken hours if done manually, can now be achieved in few minutes using our new proposed system, which is not only more effective but also an economical and efficient way of integrating coding techniques for finance.

To increase its scalability and usefulness, the program is converted into an executable file format that is capable of running as a program in a computer without its dependencies. Therefore, the user does not need to download any programming language to use it. We can run it through the command prompt or by double-clicking the icon.

C. WORKING OF THE WEB APPLICATION

1. Enable the web app by double-clicking the program file named 'main.exe'. It will open your web browser and accesses the localhost: eight thousand server because there is no central server host for the application. (Fig 9a.)
2. The main page will ask you to upload your input excel file. On clicking 'Choose file', you can navigate through your directory and select the file to upload. Once selected, click "Submit" to move to the next page. (Fig 10a.)
3. The next page will show you the names of the companies of your input file that match with the name of companies present on Yahoo finance. Each page consists of 2 companies and you can select your choice by clicking on the radio button. Once selected, you need to press 'Submit' to proceed to the next page. As soon as you press 'Submit', extraction of data of those companies starts in the background. (Fig 10b.)
4. You need to repeat the process of choosing 2 companies at a time and press 'Submit' on each page.
5. Once the selection process is completed, the next page displays the list of companies that were present in the input document but could not be found on Yahoo finance. Here, we have an option to download this data in csv/excel format by clicking "Download" or proceed to the next page by clicking "Go to next page". (Fig 10c.)
6. The last page displays the list of companies that were successfully found. You can click "Download data", after which you will have to navigate and select the directory where you want the downloaded excel file to be saved. (Fig 10d.)

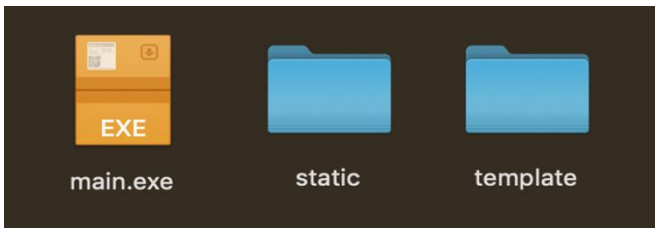

Fig 9a. Main.exe is the file to enable Flask app

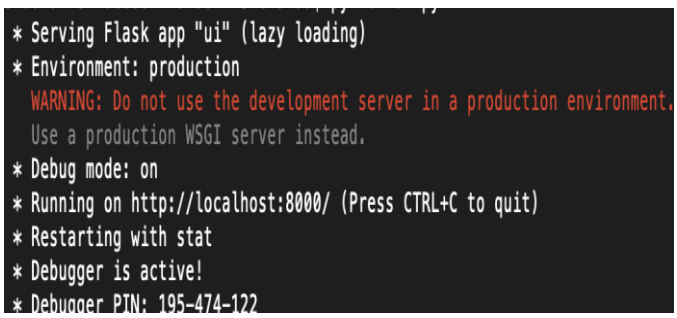

Fig 9b. Flask app running in the command prompt

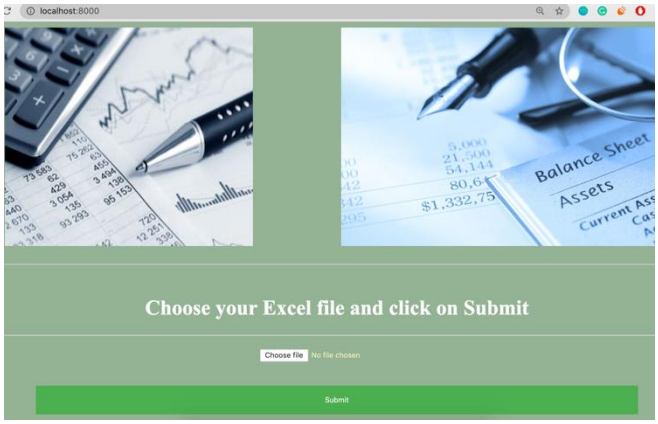

Fig 10a. Main Page

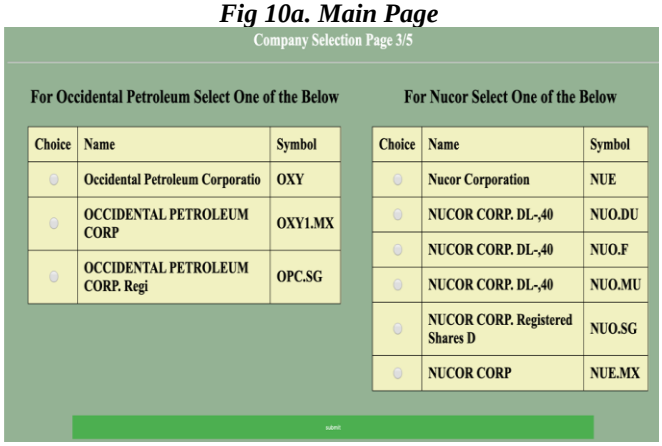

Fig 10b. Company selection page

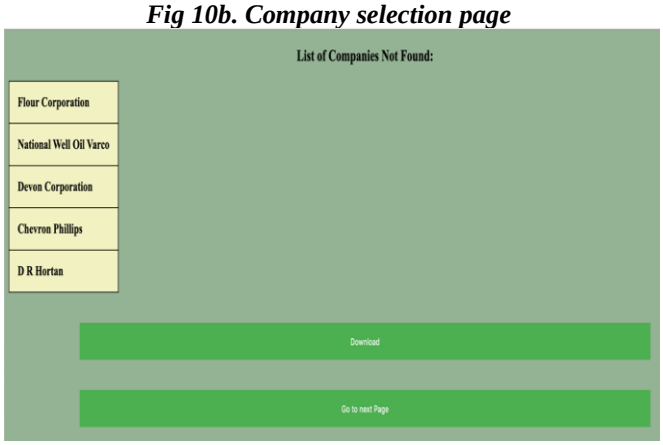

Fig 10c. List of companies not found

Fig 10d. List of companies found

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

VI. RESULT AND DISCUSSION

The goal was to take raw data of company names as our input and return cleaned data which can be used for analysis in our desired format as the output.

	A	B	C
1	Schlumberger		
2	Fluor Corporation		
3	Occidental Petroleum		
4	Apache Corporation		
5	Devon Corporation		
6	Enbridge		
7	Woodside		
8	Chevron		
9	Phillips 66		
10	Chevron Phillips		
11	National Well Oil Varco		

Fig 11. Input Data

The above figure shows us our input file. It has the list of all the companies we need to extract data from in an excel sheet. Each company name is in a different row and empty rows are avoided.

135	Company : BHP		Currency : USD				
136							
137		Breakdown	Cash And Cash Equ	Other Short Te	Total Cash	Net Receivables	Inventory
138		9/30/2019	347,943	52,960	400,903	461,774	149,653
139		9/30/2018	284,355	41,461	325,816	530,859	158,134
140		9/30/2017	521,375	44,491	565,866	398,348	137,204
141		9/30/2016	905,561	44,148	949,709	286,998	124,325
142							
143							
144	Company : Occidental		Currency : MXN				
145							
146		Breakdown	Cash And Cash Equ	Total Cash	Net Receivable	Inventory	Other Current A
147		12/31/2019	3,032,000	3,032,000	6,373,000	1,447,000	1,323,000
148		12/31/2018	3,033,000	3,033,000	4,893,000	1,260,000	746,000
149		12/31/2017	1,672,000	1,672,000	4,145,000	1,246,000	733,000
150		12/31/2016	2,233,000	2,233,000	3,989,000	866,000	1,340,000
151							
152							
153	Company : Woodside		Currency : USD				
154							
155		Breakdown	Cash And Cash Equ	Other Short Te	Total Cash	Net Receivables	Inventory
156		12/31/2019	5,792,178	39,966	5,832,144	296,888	251,213
157		12/31/2018	2,371,777	76,509	2,448,286	376,877	219,609
158		12/31/2017	407,692		407,692	364,103	238,462
159		12/31/2016	393,864		393,864	273,632	205,915
160							

Balance Sheet | Income Statements | Cash Flows | +

Fig 12. Output Data

Figure 12. shows us the extracted data. All the data is presented in an excel sheet and different tabs are generated for the balance sheet, income statement, and the cash flow.

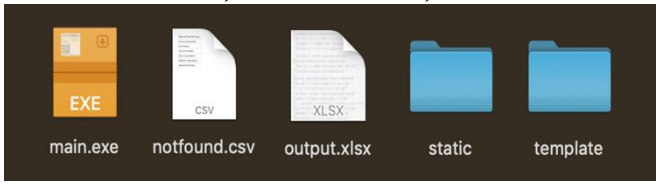

Fig 13. Downloaded file in a folder

In Figure 13. The '*notfound.csv'* is the csv file with the list of companies that were not found. The '*output.xlsx'* is the excel sheet with the extracted data. Thus, we have been successful in automating the process of collecting data from Yahoo finance. This tool can easily be used by anyone irrespective of their coding knowledge. We have removed any potential human errors that could have been generated by manual extraction and have significantly reduced the time it takes to get the data.

VII. CONCLUSION

This paper explains the various ways to extract data from the internet and discusses the legality of it. The application successfully extracts information from Yahoo finance and presents it in an articulate manner in an Excel Sheet. This could be an aid to all the business analysts who are not well versed with coding and spend their time manually extracting the information. In future works, we can add more websites to the application from which data can be extracted, and instead of only the balance sheet, cash flow and the income statement, we can add more data points for scraping. A useful application would be using it to extract stock prices from various financial websites and using them for analysis. Big Data Analysis is another major field where we can make use of this application and we can work on adding more modules which would automatically analyse the data generated from this according to our need.

REFERENCES

1. Đ. Petrović and I. Stanišević, "Web scrapping and storing data in a database, a case study of the used cars market," 2017 25th Telecommunication Forum (TELFOR), Belgrade, 2017, pp. 1-4
2. Vandana Shrivastava, "A Methodical Study of Web Crawler", Journal of Engineering Research and Application, Vol . 8, Issue 11(Part-1), Nov 2018, pp 01-08
3. S. Goel, M. Bansal, A. K. Srivastava and N.Arora, "Web Crawling-based Search Engine using Python," 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2019, pp. 436-438
4. E. Uzun, "A Novel Web Scraping Approach Using the Additional Information Obtained From Web Pages," in IEEE Access, Vol. 8, pp. 61726-61740, 2020
5. Mini Singh Ahuja, Jatinder Singh Bal and Varnica, "Web Crawler: Extracting the Web Data", International Journal of Computer Trends and Technology (IJCTT), Vol. 13, no. 3, Jul 2014.
6. Ayar Pranav and Sandip Chauhan, "Efficient Focused Web Crawling Approach for Search Engine", International Journal of Computer Science and Mobile Computing, Vol. 4, No. 5, May 2015.
7. R. P. N. Budiarti, N. Widyatmoko, M. Hariadi and M. H. Purnomo, "Web scraping for automated water quality monitoring system: A case study of PDAM Surabaya," 2016 International Seminar on Intelligent Technology and Its Applications (ISITIA), Lombok, 2016, pp. 641-648
8. Rasha Ashraf , Scraping EDGAR with Python, Journal of Education for Business, 2017, 92:4, 179-185
9. D. M. Thomas and S. Mathur, "Data Analysis by Web Scraping using Python," 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2019, pp. 450-454
10. Li Yanhong, L. Peng and Q. Zheng, "An Analysis Model of Financial Statements Based on Data Mining," 2006 3rd International IEEE Conference Intelligent Systems, London, 2006, pp. 847-850
11. V. Krotov, L. Silva ; Legality and ethics of web scraping, Proc. of 24th Americas Conference on Information Systems, August 2018, pp. 16-18

AUTHORS PROFILE

Siddhant Vinayak Chanda is a 4th year undergraduate pursuing Computer Science and Engineering at Vellore Institute Of Technology. An ardent reader, he has a keen interest in field of FinTech, world economics and is passionate about Data Science, Statistics and Finance. Email: svchanda98@gmail.com

Arivoli A is an Assistant Professor (Sr.) in the Department of Computer Science and Engineering at VIT University. She pursed my Doctorate, M.E, and B.E under Anna University. Her research area is Wireless Ad-Hoc networks, and she has published her papers in IEEE conferences and in wireless related Journals. Currently, she is working under Natural Language Processing and information security domain. Email: arivoli.a@vit.ac.in