Web Scraping And Data Acquisition Using Google Scholar

D.PRATIBA
Asst.Professor
Dept. of CSE,
RVCE, VTU
BANGALORE
INDIA

ABHAY M S
B.E Student
Dept of CSE
RVCE, VTU
BANGALORE,
INDIA

AKHIL DUA
B.E Student
Dept of CSE
RVCE, VTU
BANGALORE,
INDIA

GIRIDHAR K SHANBHAG
B.E Student
Dept of CSE
RVCE, VTU
BANGALORE,
INDIA

NEEL BHANDARI
B.E Student
Dept of CSE
RVCE, VTU
BANGALORE,
INDIA

UTKARSH SINGH
B.E Student
Dept of CSE
RVCE, VTU
BANGALORE,
INDIA

ABSTRACT

In a world where technological advancement forms the forefront of our lives, the foundation of this advancement is characterized by an exponential growth in research output from the scientific community. Research papers and their authors, however, have to go through an exhaustive ordeal to keep up to pace with the ever-changing demands of the publication committees. Hence, there is significant need for an interface that would allow researchers to easily download, sort and maintain their papers and publications with minimum work from the user end.
This paper attempts to set up an interface that would use web scraping techniques and Python modules to link a researcher's list of publications present on Google Scholar to a MySQL database and Excel application, allowing them to access and manipulate their works in minimal steps.

KEYWORDS - Python, MySQL, Excel, Google Scholar

I. INTRODUCTION

Academic Institutions often face several problems on their path of publishing and presenting their works to Universities or scientific journals. In the current scenario, academics have to manually copy-and-paste their works from Google Scholar to Excel sheets or databases, according to their preferred presentation medium. This often involves the laborious task of manually extracting the papers they have written along with number of citations, date of publication and journal it was submitted to for review. This is coupled with the drudging task of placing this data individually onto a spreadsheet and tweaking sizes.
A subsequent yet underlying problem to these submission methods is the varying format of each committee. Every committee and University requires a unique submission pattern and this becomes an exhaustive regime for every a group of individuals to manually scrape all the data required by each committee.
In the last year itself, over 2.5 million scientific research papers were published. Considering this massive scale and the underlying problems faced by each academic, there is a dire need for a user friendly interface which would allow the user to download their data in real time, with a parallel task of organizing this data in a manner that will allow them to change the presentation in minimum number of steps.

II. LITERATURE SURVEY

It quickly becomes a tedious manner to manually store the data from Google Scholars. Hence, we can use a code to do implement it. *The Data Journalism Handbook* [1] gives us information about the various ways in which we can retrieve the data from the webpage. This can be done by the usage of web-based APIs, extract data from PDFs or Screen scrap websites. The advantage of scrapping is that we can do it for any website, even if that site does not have an API for raw data access. Different tools are available for scraping which perform different functions. Readability helps us to extract the text from the web page. Scraper Wiki is a website that allows the user to code scrapers in a number of different programming languages. The HTML code, used to compose the website, are used by these scrappers which are codes written in python, ruby or PHP [1][2].
In order to make the data extracted useful for analysis there is a need to store it in a structured form. *Collecting data from the Modern Web* [3] explains the different methods which can be employed to store the data. 1. Media files can be stored either by its reference or by downloading the file itself. Storing by its reference means including its URL. The advantage of using the URL is Scrapers run much faster, require

ISBN: 978-1-5386-6078-2 © 2018 IEEE

less bandwidth and saves a lot of memory space [3]. 2. CSV or comma-separated is a popular file format which can be used to store the in a spreadsheet and it is also supported by Microsoft Excel because of its simplicity. MySQL is very popular open source relational database management. It is robust, flexible, full-featured DBMS and used by top websites: YouTube [4], Twitter [5] and Facebook. It enables to store, sort and analyze the data.

The interface used in this paper uses the principle of web scrapers. The interface extracts the text using the HTML code. This is done by a python program. The data is exported to Excel sheet and MySQL database which is performed using CSV and PyMySQL respectively. The advantages of using MySQL database is that it is the most secured and reliable relational database used in management applications. Therefore, the data of the research paper is secured. MySQL features a distinct storage-engine framework that facilitates system administrators to configure the MySQL database server for a flawless performance [6]. MySQL uses the unencrypted connection between the client and server by default. The MySQL serves checks for three scopes: IP address, username, password. The server stores the data in the form of tables in the MySQL database. MySQL provides functions to encrypt and decrypt data values [7].

III. PROPOSED SYSTEM

OBJECTIVES

This paper aims to provide a distinctive analyses on approaching web scraping with respect to Google Scholar and using the data acquired to build a user friendly GUI that would allow them to easily access their works with minimal work.

The aforementioned aims produce the following paper objectives:

- To analyze and extract key HTML links from Google Scholar source code [8][9].
- To disseminate the extracted links so as to provide a precise dataset linked with the users end goal.
- To create a database on MySQL that would allow for the key sections to be displayed.
- To congregate extracted data and display according to given database parameters.

After the initial data collection and screening, it will be displayed in both an Excel spreadsheet and MySQL database, providing the user with comfortable choices. The long term goal of this paper is to provide the user to access a user friendly interface that would minimize the time it would take them to complete these tasks. It also aims to build a foundation for the creation of industry-applicable software that would provide global access to this technology.

METHODOLOGY AND IMPLEMENTATION

The following methodology was adopted in designing this interface.

- Firsty, before fetching the required content the URL of the page has to be accessed for which the method adopted was to store the URLs of different authors in an external file like a excel sheet and then using the attributes of the file fetch the particular URL requested by the user.

- The google scholar page which gives the list of articles that has been written by a certain author has a basic html architecture. Python has simple methods to access the html code and take the required information i.e web scraping.

- The methods used to store the collected information was to create both the database table and the excel file within the program itself so the only the required amount of memory would be utilised by the storing files. The statergy used to store the real time data was to overwrite the data in the excel sheet and in the database the whole table was truncated and the contents were re-inserted.

The following interface was created using advanced python and basics of MySQL. The following interface was created in the following fragments.

- First step is to have python installed in the system and it should have the following python modules installed in it – pymysql, csv, requests, Beautiful soup, pandas, and tkinter. If not preinstalled user can install it using pip install command.
- Next step is to create an excel sheet which contains the author names and their respective Google scholar URLs in an indexed manner which can be further be used in the program to fetch the URL.
- Then a GUI has been created to get the user input to make the interface more users friendly.
- Then the beautiful soup module is used to scrap the entire html code of the specific URL.
- Then the required data is selected and the data is stored temporarily in lists according

ISBN: 978-1-5386-6078-2 © 2018 IEEE

to the format in which the user would like to see the data.

- The next part is to connect to the database by providing the user's credentials and create a table in a format that the client has demanded of. In this case the format is-

Field	Type	Null	Key	Default	Extra
TITLE	varchar(255)	NO		NULL	
AUTHOR	varchar(255)	NO		NULL	
JOURNAL	varchar(255)	NO		NULL	
CITATIONS	int(4)	YES		NULL	
YEAR	int(4)	YES		NULL	

Fig 1. Structure of the MySQL table

- Then the data was exported to the excel sheet using the csv module. Since the data had to be stored in real time and rewritten according to the user input each time the program was run, that's why the 'w+' file mode has been used.
- Then the SQL commands have to be written using the pymysql module to insert the data items that have been stored in the list. Each time the program is executed the existing data items in the table are truncated and the new data items that have been stored in the lists are inserted into the table. This helps in storing the real time data.
- Finally check the output compatibility with the database table.

This completes the creation of the interface.

The programming language employed for the stated objective was python. This particular programming language was chosen as python is rich in the set of modules it entertains for the users. Using these modules made it easier to provide interfaces between the program and the Google scholar web page, the interface and the destination of the results i.e. the MySQL database and the Excel spreadsheet. Furthermore it reduced the size of the code, helped in faster outputs with efficient results and it also makes the code much simpler and readable to the user. Python also provides inbuilt functions for data acquisitions directly from the web page and also to make the interface more user friendly by making simple GUIs for input purposes. Due to these advantages which python offers, it was the go to choice for us to make this interface.

The following interface was completely built on the python IDLE which can be downloaded from the www.python.org website. The platform on which the whole interface works on is Windows 10. The results and outcomes were stored in MySQL database and MS Excel. Python IDLE provides a very simple platform to write the python code and as the following implementation only does data acquisition using web scraping and storing it. Excel and MySQL were used for two contrary purposes one for security of the data (MySQL) and other for the ease accessing data (Excel). Using the database for storing the data also requires the user's credentials already written in the program, so the storage of the results is at the programmer's discretion. And MySQL is also very compatible with python. Excel was chosen due to the ease of manipulating data stored in it. Overall, all these platforms were chosen for better connectivity among them and also the ease with which the user could program them.

During the creation of this whole interface a lot of obstacles were encountered. And as python was used to create it, they were comfortably overcome. The difficulties encountered and how they were tackled –

- The first obstacle was to acquire the Google scholar URL for the respective person [6]. This obstacle became bigger because searching for a particular person itself was a very difficult task as the search bar needed to write the name in a specific manner like first name followed by last name or sometimes author name with the institute nameetc,. This problem was dealt by making a separate excel sheet to store the author name and also the associated Google scholar URL. These URLs where then indexed and called in the interface according to the input given by the user as to which author's data the user wants to access.
- Another major obstacle during creating this interface was to sort the data according to the client's needs. This problem was solved by temporarily storing the data in the form of lists where the user has the control over the dimension and the number of elements in each list. This data was then exported to the database.
- Exporting to the database according to the client's needs was the roadblock to this interface but this was overcome by creating the database and table in which the data has to be stored using python itself. This lets the user create the database and table with the required amount of sorting.
- Since the interface works on real-time data and stores it, there came the problem of updating the database and the excel sheet each time. This was dealt by using few basic MySQL and file modes that are used in python.

- As the SQL commands are very sensitive in their syntax it created a problem as the data on all the WebPages aren't the same. This was dealt by creating a exception block to execute the SQL commands .

```
sql = """CREATE TABLE PAPER37 (
        {} INT(4) AUTO_INCREMENT PRIMARY KEY,
        {} VARCHAR(255) ,
        {} VARCHAR(255) ,
        {} VARCHAR(255) ,
        {} INT(4),
        {} INT(4)  )""".format('SLNO','TITLE','AUTHOR','JOURNAL','CITATIONS','YEAR')

with open('test1.csv','w+',newline='') as fp:
    a = csv.writer(fp,delimiter=',')
    a.writerows(list_of_rows)


delete = "TRUNCATE PAPER37"
```

Fig 2. Python file

Fig 2 shows the python file appending mode for the excel sheet and how the data items were truncated so as to store the real time data.

Overall, using the knowledge of MySQL and the rich set of modules offered by python this interface was created even though there were few hiccups along the process of creation of this interface.

IV. RESULTS AND ANALYSIS

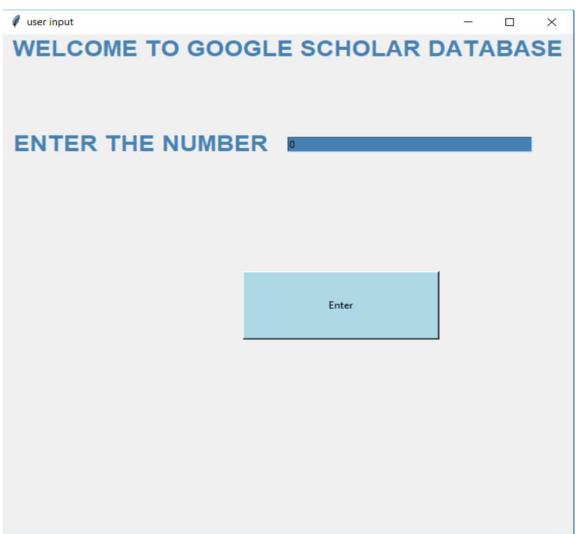
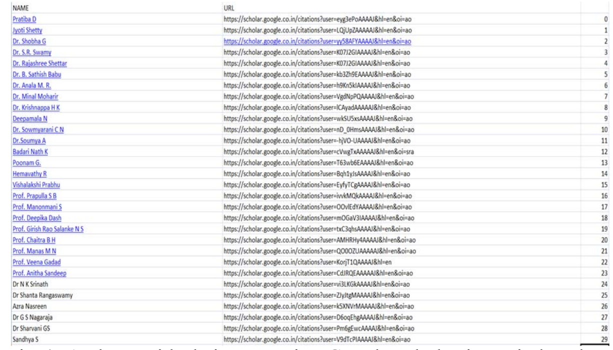
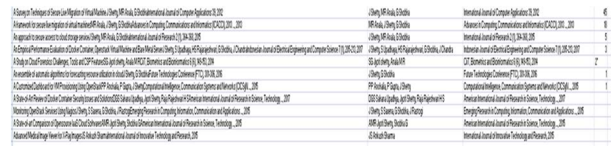


Fig 3. Image of the GUI created to take the user input



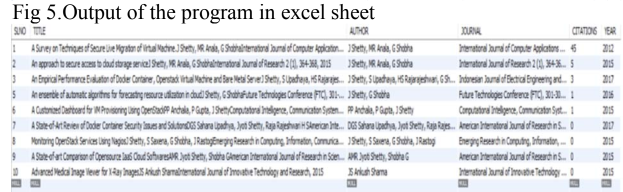
NAME	URL	
Pratiba D	https://scholar.google.co.in/citations?user=eyg3ePoAAAAJ&hl=en&oi=ao	0
Jyoti Shetty	https://scholar.google.co.in/citations?user=LQjUpZAAAAAJ&hl=en&oi=ao	1
Dr. Shobha G	https://scholar.google.co.in/citations?user=yy58AFYAAAAJ&hl=en&oi=ao	2
Dr. S.R. Swamy	https://scholar.google.co.in/citations?user=K07J2GIAAAAJ&hl=en&oi=ao	3
Dr. Rajashree Shettar	https://scholar.google.co.in/citations?user=K07J2GIAAAAJ&hl=en&oi=ao	4
Dr. B. Sathish Babu	https://scholar.google.co.in/citations?user=kb3Zh9EAAAAJ&hl=en&oi=ao	5
Dr. Anala M. R.	https://scholar.google.co.in/citations?user=h9Kn5kIAAAAJ&hl=en&oi=ao	6
Dr. Minal Moharir	https://scholar.google.co.in/citations?user=VgdNpPQAAAAJ&hl=en&oi=ao	7
Dr. Krishnappa H K	https://scholar.google.co.in/citations?user=lCAyadAAAAAJ&hl=en&oi=ao	8
Deepamala N	https://scholar.google.co.in/citations?user=wkSU5xsAAAAJ&hl=en&oi=ao	9
Dr. Sowmyarani C N	https://scholar.google.co.in/citations?user=nD_0HmsAAAAJ&hl=en&oi=ao	10
Dr.Soumya A	https://scholar.google.co.in/citations?user=-hjVO-UAAAAJ&hl=en&oi=ao	11
Badari Nath K	https://scholar.google.co.in/citations?user=cVwgTxAAAAAJ&hl=en&oi=sra	12
Poonam G.	https://scholar.google.co.in/citations?user=T63wb6EAAAAJ&hl=en&oi=ao	13
Hemavathy R	https://scholar.google.co.in/citations?user=Bqh1yJsAAAAJ&hl=en&oi=ao	14
Vishalakshi Prabhu	https://scholar.google.co.in/citations?user=EyfyTCgAAAAJ&hl=en&oi=ao	15
Prof. Prapulla S B	https://scholar.google.co.in/citations?user=ivvkMQkAAAAJ&hl=en&oi=ao	16
Prof. Manonmani S	https://scholar.google.co.in/citations?user=OOvlEdYAAAAJ&hl=en&oi=ao	17
Prof. Deepika Dash	https://scholar.google.co.in/citations?user=mOGaV3IAAAAJ&hl=en&oi=ao	18
Prof. Girish Rao Salanke N S	https://scholar.google.co.in/citations?user=txC3qhsAAAAJ&hl=en&oi=ao	19
Prof. Chaitra B H	https://scholar.google.co.in/citations?user=AMHRHy4AAAAJ&hl=en&oi=ao	20
Prof. Manas M N	https://scholar.google.co.in/citations?user=QO0OZUAAAAAJ&hl=en&oi=ao	21
Prof. Veena Gadad	https://scholar.google.co.in/citations?user=KorjT1QAAAAJ&hl=en	22
Prof. Anitha Sandeep	https://scholar.google.co.in/citations?user=CdJRQEAAAAAJ&hl=en&oi=ao	23
Dr N K Srinath	https://scholar.google.co.in/citations?user=vi3LKGkAAAAJ&hl=en&oi=ao	24
Dr Shanta Rangaswamy	https://scholar.google.co.in/citations?user=ZJyJtgMAAAAJ&hl=en&oi=ao	25
Azra Nasreen	https://scholar.google.co.in/citations?user=k5XNVrMAAAAJ&hl=en&oi=ao	26
Dr G S Nagaraja	https://scholar.google.co.in/citations?user=D6oqEhgAAAAJ&hl=en&oi=ao	27
Dr Sharvani GS	https://scholar.google.co.in/citations?user=Pm6gEwcAAAAJ&hl=en&oi=ao	28
Sandhya S	https://scholar.google.co.in/citations?user=V9dTcPIAAAAJ&hl=en&oi=ao	29

Fig 4. Authors with their respective Google scholar in an indexed format



A Survey on Techniques of Secure Live Migration of Virtual Machine J Shetty, MR Anala, G ShobhaInternational Journal of Computer Applications 39, 2012	J Shetty, MR Anala, G Shobha	International Journal of Computer Applications 39, 2012	45	2012
A framework for secure live migration of virtual machinesMR Anala, J Shetty, G ShobhaAdvances in Computing, Communications and Informatics (ICACCI), 2013 ..., 2013	MR Anala, J Shetty, G Shobha	Advances in Computing, Communications and Informatics (ICACCI), 2013 ..., 2013	18	2013
An approach to secure access to cloud storage serviceJ Shetty, MR Anala, G ShobhaInternational Journal of Research 2 (1), 364-368, 2015	J Shetty, MR Anala, G Shobha	International Journal of Research 2 (1), 364-368, 2015	5	2015
An Empirical Performance Evaluation of Docker Container, Openstack Virtual Machine and Bare Metal ServerJ Shetty, S Upadhaya, HS Rajarajeshwari, G Shobha, J ChandraIndonesian Journal of Electrical Engineering and Computer Science 7 (1), 205-213, 2017	J Shetty, S Upadhaya, HS Rajarajeshwari, G Shobha, J Chandra	Indonesian Journal of Electrical Engineering and Computer Science 7 (1), 205-213, 2017	3	2017
A Study on Cloud Forensics: Challenges, Tools and CSP FeaturesSG Jyoti shetty, Anala MRCiiT, Biometrics and Bioinformatics 6 (6), 149-153, 2014	SG Jyoti shetty, Anala MR	CiiT, Biometrics and Bioinformatics 6 (6), 149-153, 2014	2	2014
An ensemble of automatic algorithms for forecasting resource utilization in cloudJ Shetty, G ShobhaFuture Technologies Conference (FTC), 301-306, 2016	J Shetty, G Shobha	Future Technologies Conference (FTC), 301-306, 2016	1	2016
A Customized Dashboard for VM Provisioning Using OpenStackPP Anchalia, P Gupta, J ShettyComputational Intelligence, Communication Systems and Networks (CICSyN) ..., 2015	PP Anchalia, P Gupta, J Shetty	Computational Intelligence, Communication Systems and Networks (CICSyN) ..., 2015	1	2015
A State-of-Art Review of Docker Container Security Issues and SolutionsDGS Sahana Upadhya, Jyoti Shetty, Raja Rajeshwari H SAmerican International Journal of Research in Science, Technology, ..., 2017	DGS Sahana Upadhya, Jyoti Shetty, Raja Rajeshwari H S	American International Journal of Research in Science, Technology, ..., 2017		2017
Monitoring OpenStack Services Using NagiosJ Shetty, S Saxena, G Shobha, J RastogiEmerging Research in Computing, Information, Communication and Applications ..., 2015	J Shetty, S Saxena, G Shobha, J Rastogi	Emerging Research in Computing, Information, Communication and Applications ..., 2015		2015
A State-of-art Comparison of Opensource IaaS Cloud SoftwaresAMR Jyoti Shetty, Shobha GAmerican International Journal of Research in Science, Technology, ..., 2015	AMR Jyoti Shetty, Shobha G	American International Journal of Research in Science, Technology, ..., 2015		2015
Advanced Medical Image Viewer for X-Ray ImagesJS Ankush SharmaInternational Journal of Innovative Technology and Research, 2015	JS Ankush Sharma	International Journal of Innovative Technology and Research, 2015		2015

Fig 5.Output of the program in excel sheet



SLNO	TITLE	AUTHOR	JOURNAL	CITATIONS	YEAR
1	A Survey on Techniques of Secure Live Migration of Virtual Machine.J Shetty, MR Anala, G ShobhaInternational Journal of Computer Application...	J Shetty, MR Anala, G Shobha	International Journal of Computer Applications ...	45	2012
2	An approach to secure access to cloud storage serviceJ Shetty, MR Anala, G ShobhaInternational Journal of Research 2 (1), 364-368, 2015	J Shetty, MR Anala, G Shobha	International Journal of Research 2 (1), 364-36...	5	2015
3	An Empirical Performance Evaluation of Docker Container, Openstack Virtual Machine and Bare Metal ServerJ Shetty, S Upadhaya, HS Rajarajes...	J Shetty, S Upadhaya, HS Rajarajeshwari, G Sh...	Indonesian Journal of Electrical Engineering and...	3	2017
5	An ensemble of automatic algorithms for forecasting resource utilization in cloudJ Shetty, G ShobhaFuture Technologies Conference (FTC), 301-...	J Shetty, G Shobha	Future Technologies Conference (FTC), 301-30...	1	2016
6	A Customized Dashboard for VM Provisioning Using OpenStackPP Anchalia, P Gupta, J ShettyComputational Intelligence, Communication System...	PP Anchalia, P Gupta, J Shetty	Computational Intelligence, Communication Syst...	1	2015
7	A State-of-Art Review of Docker Container Security Issues and SolutionsDGS Sahana Upadhya, Jyoti Shetty, Raja Rajeshwari H SAmerican Inte...	DGS Sahana Upadhya, Jyoti Shetty, Raja Rajes...	American International Journal of Research in S...	0	2017
8	Monitoring OpenStack Services Using NagiosJ Shetty, S Saxena, G Shobha, J RastogiEmerging Research in Computing, Information, Communica...	J Shetty, S Saxena, G Shobha, J Rastogi	Emerging Research in Computing, Information, ...	0	2015
9	A State-of-art Comparison of Opensource IaaS Cloud SoftwaresAMR Jyoti Shetty, Shobha GAmerican International Journal of Research in Scien...	AMR Jyoti Shetty, Shobha G	American International Journal of Research in S...	0	2015
10	Advanced Medical Image Viewer for X-Ray ImagesJS Ankush SharmaInternational Journal of Innovative Technology and Research, 2015	JS Ankush Sharma	International Journal of Innovative Technology ...	0	2015
NULL	NULL	NULL	NULL	NULL	NULL

Fig 6. Output of the program in the MySQL database

V. CONCLUSIONS

Thus by using the above discussed methodologies, the platfrom was created succesfully which references an excel sheet with the codes randomly allotted to each scholar, scrapes data from the Google scholar platform and presents it in a tabular form in an excel sheet as well as a MySQL database[10]. The program has involved the inclusion of several Python modules so that the length of the code was reduced substantially. Therefore this program enables any individual concerned present all the research statistics of any particular scholar whose URL on the Google Scholar platform has been indexed in our reference excel sheet as and when required and necessary.

VI. FUTURE WORK

The program as of now uses an excel sheet containing the numerical codes assigned arbitrarily to every professor/scholar as a reference which it then uses to fetch the associated URLs. Hence development can be made so that anyone's work can be examined by the use of their name itself instead of an indexing number. Work can be done in order to extend the program to other scholar platform like Scopus etc., by knowing the data structure the website uses for hoarding.

Another problem that arises is even if the name of a particular scholar is used, the user won't realise if he has used the exact name as that used by

the former in his bio. So there is a possibility of mismatch occuring if the user input has even the slightest error.

Countering all the above mentioned limitations can be considered as an area of interest and the program can be developed further. Also developing applications based on the program for various platforms and making available to those in need can also be considered as a venture further in this direction.The methodology used in this program can also be used in various other aspects like business, stock marketing, getting details about a particular person's journey record by some tourism provider etc. All these involve manipulation and management of database which has been an integral and core portion of the methodology used here.

This provides a wide area of improvements on which future work can be done.

VII. REFERENCES

1.Gray, J., Bounegru, L. and Chambers, L. (2012). The data journalism handbook. Sebastopol, CA: O'Reilly Media.

2.S. K. Malik and S. Rizvi, "Information Extraction Using Web Usage Mining, Web Scraping and Semantic Annotation," 2011 International Conference on Computational Intelligence and Communication Networks, Gwalior, 2011, pp. 465-469.

3.Mitchell, R. (2015). Web Scrapping with Python. 1st ed. O'Reilly Media.

4.Joab Jackson, "YouTube Sales MySQL with Go Code," PC World, December 15, 2012.

5.Jeremy Cole and Davi Arnaut," MySQL at Twitter Engineering Blog, April 9, 2012.

6.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.

7.I. Zoratti, "MYSQL Security Best Practices," 2006 IET Conference on Crime and Security, London, 2006, pp. 183-198.

8.D. Yang, A. N. Zhang and W. Yan, "Performing literature review using text mining, Part I: Retrieving technology infrastructure using Google Scholar and APIs," 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, 2017, pp. 3290-3296.

9.C. Yang, Y. Yan and Q. Zhu, "The Face Database Development of Science and Technology Expects Based on Web Mining," 2012 Fourth International Conference on Multimedia Information Networking and Security, Nanjing, 2012, pp. 388-391

10.K. I. Satoto, R. R. Isnanto, R. Kridalukmana and K. T. Martono, "Optimizing MySQL database system on information systems research, publications and community service," 2016 3rd International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE), Semarang, 2016, pp. 1-5.