European Journal of Molecular & Clinical Medicine
ISSN 2515-8260 Volume 07, Issue 3, 2020

Automatic Retrieval of Updated Information Related to COVID-19 from Web Portals

[1]Prateek Raj, [2]Chaman Kumar, [3]Dr. Mukesh Rawat

[1,2,3] Department of Computer Science and Engineering, Meerut Institute of Engineering and Technology, Meerut 250005, U.P, India

Abstract

In the world of social media, we are subjected to a constant overload of information. Of all the information we get, not everything is correct. It is advisable to rely on only reliable sources. Even if we stick to only reliable sources, we are unable to understand or make head or tail of all the information we get. Data about the number of people infected, the number of active cases and the number of people dead vary from one source to another. People usually use up a lot of effort and time to navigate through different websites to get better and accurate results. However, it takes lots of time and still leaves people skeptical. This study is based on web-scraping & web-crawlingapproach to get better and accurate results from six COVID-19 data web sources.The scraping script is programmed with Python library & crawlers operate with HTML tags while application architecture is programmed using Cascading style sheet(CSS) &Hypertext markup language(HTML). The scraped data is stored in a PostgreSQL database on Heroku server and the processed data is provided on the dashboard.

Keywords: Web-scraping, Web-crawling, HTML, Data collection.

I. INTRODUCTION

The internet is wildly loaded with data and contents that are informative as well as accessible to anyone around the globe in various shape, size and extension like video, audio, text and image and number etc. This variation in the data format provides irregularity which leads to various difficulties in the retrieval process of the data since not every data is significant to every user. This abundance in data sometimes becomes offensive rather than blessing [1]. The rise in demand of such data to advocate people to develop latest methods and application has increased over time therefore the retrieval process should be quick and standard.The procedure of retrieving the data is depended upon storing the data into the database and then to return the data in accordance to the need of the user. However, almost 54% of the data searched on the internet is for administration and merchandise, 47% of the information for education purposes, 38% data for clinical data and health, 27.8% for work finding process, 24% information for administrative organizations and governmental [2].Nevertheless in the months since COVID-19 hit the world, scholars have been trying hard to discover the aspect of the virus, its affecting reasons and its preventive measures. Unlike scholars, public is accustomed tosee procession of numbers and charts that shows the spread of coronavirus, it's affect to certain place and certain age group of people. Therefore the data is important for decisions on how to react and keep ourselves safe [3][4]. People invest a lot of time and energy trying to get the latest data from various sources which is oppressive therefore we need a platform which can provide better and comprehensive data for COVID-19 available on different websites and provide user with latest updates in a timely manner. Since the automatic update and scheduler will provide the updated data to the users at their preferred preset time hence it will prevent users to login in everyday to get updates which the current system lacks.

Web scraping is a technique used to gather abundance of data through various websites and

store it into our database. The process of data retrieval can be categorized in two steps consecutively:

1. Fetching websites.
2. Extracting required information through these websites.

Accordingly, these kinds of program imitate people's inquiry on the internet by applying Hypertext-Transfer-Protocol, or by simulating through a complete browsing application e.g. Firefox [5]. Data extraction can mainly be seen in web indexing where the extraction is done using spider or crawler and the extracted data is recorded and this process is applied in most search engines. Presently, use of web-scraping has increased broadly in different domains like E-commerce, Finance reports, news, jobs scraping, blogs etc. while public opinion gathering, detecting fraudulent reviews, custom analysis and curation, search engines etc. being different purposes of scraping [6].

I. PROPOSED WORK-PLAN

The network is devised to carry out the search for statistical data on COVID-19 through HTML CSS-root framework with the help of data extraction techniques.
We have used six COVID-19 data websites as information source, which are: covidindia.org, oneindia.com, mygov.in, grainmart.in, ndtv.com, tooloogle.com.
Each website is chosen on the ground of statistical data. The data would be scraped at the interval of 12 hours (i.e., twice a day say 6:00am and 6:00 pm)every day.
The operational layer is accountable for recovering required attributes from websites. It is built on Python scripts and uses scrapy library to parse data. The web application and information scripting are shown in fig.1 and fig.2. The procedure initiates by scheduler that scraps the data at the scheduled time and stores the scraped data into database. The user provides the required state and district data as well as the routine notification time on the sign up page. After retrieving the data from 1st website the data is stored in the temporary variables and the retrieved data from the other five websites are added into the previous stored data and then the overall data is averaged. The processed data is stored into the database and then presented on the dashboard. The same data is also presented into the graphical format to give a better view.

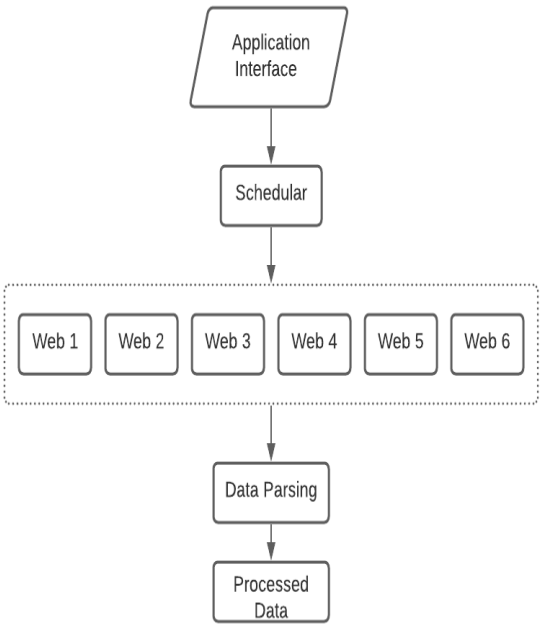

Fig. 1. System Overview

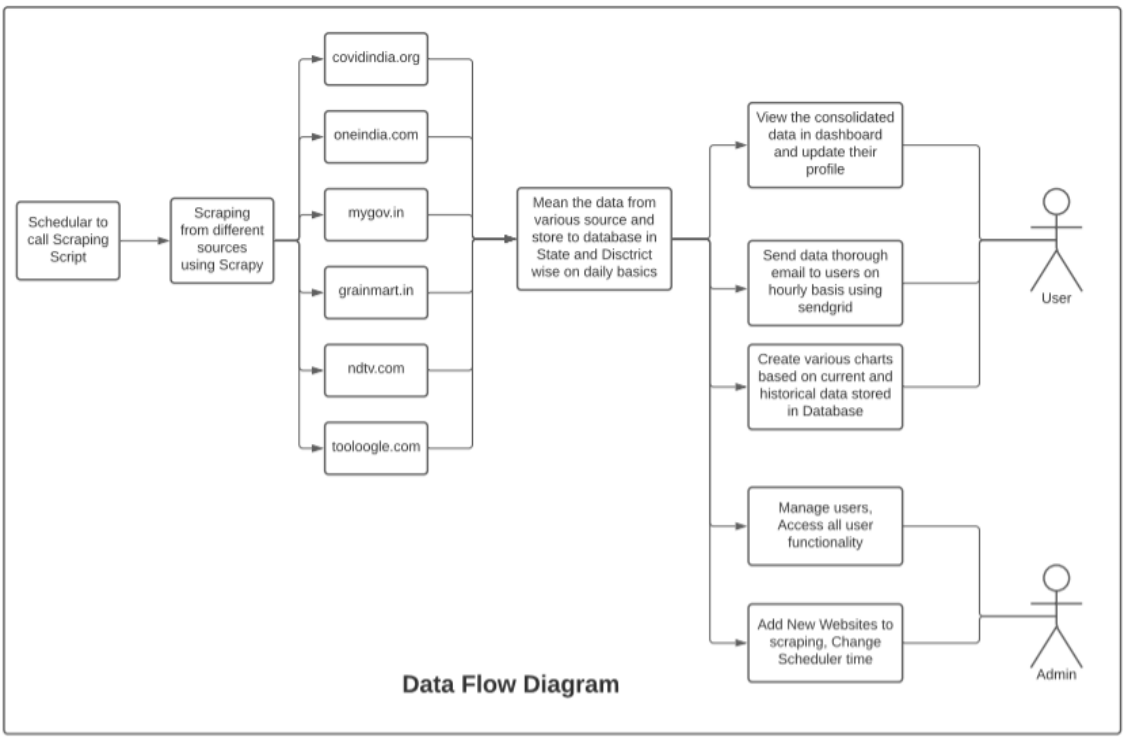

Fig. 2. Data Flow Diagram

The user details are also stored into the database and the updated state as well as district data is retrieved from the database and mailed to the user at the scheduled time.

A. Working Principle

The working principle of the application can be explained in following steps:

1. Importingrequired Python library.
2. Fetch the URL-address with scrapy& request library and saving it in temp variables.
3. Parsing HTML into temporary variable.
4. Scrape data like Cases, Deaths, Active, Cured.
5. Store the data into the database.
6. View processed data in the Dashboard.

The 1st and 2nd step comes within the domain of web-crawling, which is executed with the help of Python library& the 3rd and 4th step comes within the domain of web-scraping.

B. Implementing scraping

We have used Python as a primary language in this implementation since it has plethora of library and has better readability. We have used scrapy & request library to scrape various sites which are explained below:

1. Requests Module

The request module allows you to send HTTP requests using Python. We can also use the

request module to download files from a URL. And if the URL redirects to the actual file, the request get() method follows it and download the actual file. It is simple and easy to use library [5][7]. It also have a lot of features from passing parameter to send custom header and SSL verifications. The passing parameter function allows you to get any specific result from webpage by providing these query string as a dictionary. Therefore this is a very handy module when we are trying to scrape some webpages for information [8].

2. Scrapy

Python has a web scraping framework called Scrapy which provides an overall package to the programmers without any worry for the maintenance of the code [9]. In scrapy we define classes called spiders that are used by scrapy to scrap the data from the websites and these are self-contained crawler with given instructions. Scrapy also provides you with pipelines that allow you to validate your data, remove unnecessary data, store data into the database and also allows us to manage lots of variable and so on [7][8]. The algorithm used to scrape the data from websites is shown in Fig. 3.

```
start
        Declaration
                list scraping site
                object statedata

        function parse(Arugument response):
                if response site is equal to covidindia
                        get state data using css selectors from response
                        for state in states
                                store in statedata

                else if response site is equal to oneindia
                        get state data using css selectors from response
                        for state in states
                                store in statedata

                else if response site is equal to mygov
                        get state data using css selectors from response
                        for state in states
                                store in statedata

                else if response site is equal to grainmart
                        get state data using css selectors from response
                        for state in states
                                store in statedata
                                get district data related to state using css selectors
                                for district in districts
                                        store in district in statedata

                else if response site is equal to ndtv
                        get state data using css selectors from response
                        for state in states
                                store in statedata

                else if response site is equal to tooloogle
                        get state data using css selectors from response
                        for state in states
                                store in statedata
                                get district data related to state using css selectors
                                for district in districts
                                        store in district in statedata

                return statedata

        function save_to_database(Arugument statedata)
                for state in statedata
                        mean the list of data available
                        store to database in StateData Table
                        for district in districts of state
                                mean the list of data available
                                store to database in DistrictData Table
                return stateData

        function periodic_parsing():
                if time is 06:00 AM:
                        call function parse
                if time is 06:00 PM:
                        call function parse

        function hourly_email():
                for everyhour in hour:
                        get users list subscribed on that hour
                        send email

stop
```

Fig. 3. Algorithm Used

II. RESULTSAND DISCUSSION

The study executes a Python (Django framework) framed web-application that provide facilities of acquiring latest & comprehensive COVID-19 data from six websites. User can also specify particular state and district to get his/her district data on COVID-19 at a specific time of the day. This web-application is the necessity of this time as various web sources have been giving different data and stats. Because of fast pace life and increasing workload people generally lack time for accessing various websites to acquire data or information. The flow is

given in fig.2.

Before moving towards the results, we must take into consideration about the legality issues when administrating these kinds of applications [10]. This web application accesses information from various websites to provide a comprehensive data to users. The problem that generally arises with such applications is data usage legality.

However, this web-application accesses that information that isfreely available on these websites and as a matter of fact available to anyone who accesses these websites so there is no violation done. Regular update of the websites used by their respective administrators is also one of the major concerns but the websites used here are regularly updated while the changes made in these websites are acquired with the help of strategy to scrape these web-pages every 12 hours. So every-time updated websites are used for additional processing.

Although, we do have to save user data in the database to provide user specific data at the scheduled time. The complete framework of the application is given in the figure below:

Fig. 4. Complete Framework

A. Information Retrieval Results

The data on COVID-19 including States, Cases, Cured, Active and Deaths etcetera are scraped by scrapy. The scrapy also parse the data and remove all unnecessary content such as tag and explanatory data. The data represented is informative but the response time and success rate of multiple user hit gives the overall quality of the service. To check the average response time and successful response rate we conducted a test and checked the overall quality of the service.

1).Test Case

We ran a stimulation test in which we started from 100 user hits in a time period of 1 minute to 1000 user hits in the same time period and recorded the response time of user hit set as shown in the fig.5 and the over all response time(i.e., 419.3 ms) is calculated by using equation 1 to find out the performance.

$$\overline{X} = \frac{\sum X}{N} \quad \text{... (1)}$$

Where N is the total number to test set(i.e.,10) and X is the time(in ms).

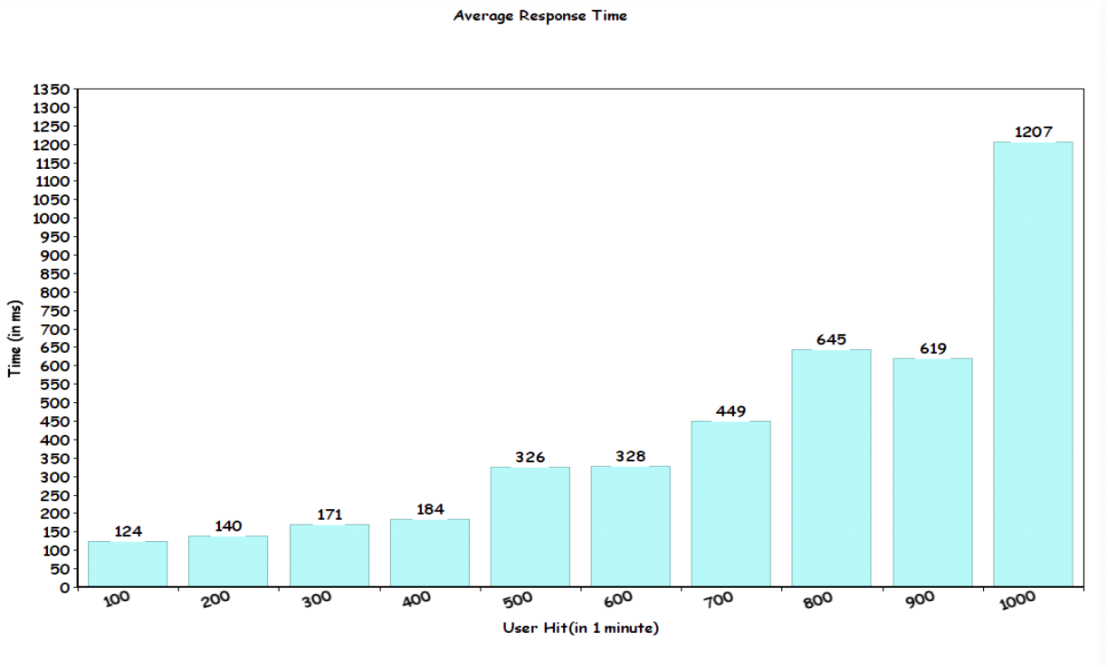

Fig. 5. Average Response Graph

We also created another test set to check the accurate response rate of user hits by increasing the user hit loads in 1 minute and setting the response time out for 5 seconds for each set. The results of the test are shown in the fig.6 and the overall accuracy was calculated using the same equation above, though here X is the success percent of each set. The overall accuracy of the response rate was 95.1%.

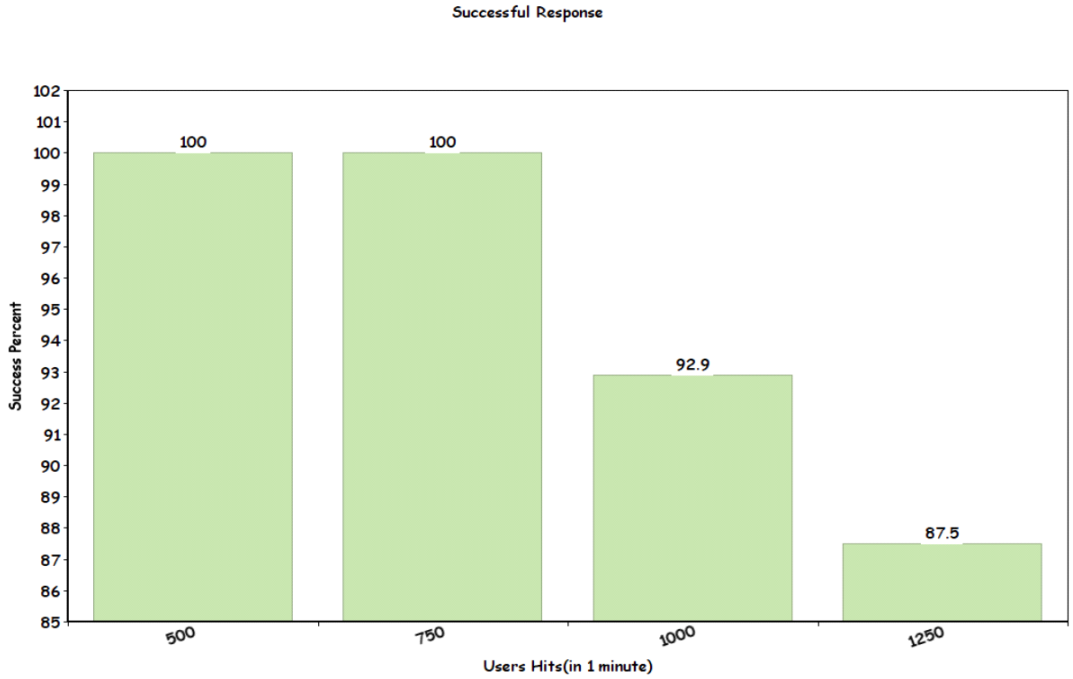

Fig. 6. Successful Response Graph

III. CONCLUSION

The web world contains an insurmountable amount of data and with time passing the amount of data is exponentially increasing therefore the process of fetching the information with the help of search engines leads to nonspecific and inconveniently wide range of data. But with the help of these data extraction methods we can reduce this problem and narrow down our search results. Similarly we have created a web application to collect better and comprehensive COVID-19 data from various websites with the help of web-scraping technique. Although there are couple of things that should be kept in mind before implementing such application techniques such as no law should be broken while collecting any data since in our case we have accessed only those data that are accessible to all public hence our method is legal and sound. Hereafter, we are looking to create phone software to increase our reach.

REFERENCES

[1] H. Xuqldzdwl, G. Dndnrp, and D. F. Lg, “RQ H & RPPHUFH : HEVLWHV,” pp. 5–8.
[2] F. Aulia and W. Dhewanto, “Formulation of ECommerce Website Development Plan UsingMultidimensional Approach for Web Evaluation,”in Procedia - Social andBehavioral Sciences, 2014,vol. 115, no. Iicies 2013, pp. 361–372.
[3] NGUYEN, DONGTHI THAO, and KIEUTHI THU CHUNG. "NEW TRENDS IN TECHNOLOGY APPLICATION IN EDUCATION AND CAPACITIES OF UNIVERSITIES LECTURERS DURING THE COVID-19 PANDEMIC." International Journal of Mechanical and Production Engineering Research and Development (IJMPERD)10.3, Jun 2020, 1709-1714
[4] D. K. Mahto and L. Singh, “A Dive into WebScraper World,” pp. 689–693, 2016.
[5] S. Upadhyay, V. Pant, and S. Bhasin, “Articulatingthe Construction of a Web Scraper for Massive DataExtraction.”
[6] Singh, Brijesh P. "Modeling and Forecasting Novel Corona Cases in India Using Truncated Information: A Mathematical Approach." *International Journal of Applied Mathematics & Statistical Sciences* 9.4 (2020): 13-24.
[7] L. Junjoewong, S. Sangnapachai, and T. Sunetnanta,“ProCircle : A promotion platform usingcrowdsourcing and web data scraping technique,” 2018 Seventh ICT Int. Student Proj. Conf., pp. 1–5, 2018.
[8] Kavali, Janardhan, and Arvind Mittal. "Analysis of various control schemes for minimal Total Harmonic Distortion in cascaded H-bridge multilevel inverter." *Journal of Electrical Systems and Information Technology* 3.3 (2016): 428-441.
[9] E. Vargiu and M. Urru, “Exploiting web scraping in a collaborative filtering- based approach to web advertising,” vol. 2, no. 1, pp. 44–54, 2013.
[10] W. Scrapping and S. Annotation, “2011International Conference on Computational Intelligence and Communication Systems,” 2011.
[11] K. Sundaramoorthy, “NEWSONE- AN AGGREGATION SYSTEM FOR NEWS USING WEB SCRAPING METHOD,” 2017.
[12] By, Saved, and Krishna Pada Das. "Mathematical modelling on outbreak and spread of corona-virus (COVID-19) disease and its control strategies." (1983).
[13] H. Lo, M. Reboiro-jato, F. Fdez-riverola, and D.Glez-pen, “Web scraping technologies in an APIworld,” vol. 15, no. 5, pp. 788–797, 2013.
[14] NGUYEN, DONGTHI THAO, and KIEUTHI THU CHUNG. "NEW TRENDS IN TECHNOLOGY APPLICATION IN EDUCATION AND CAPACITIES OF UNIVERSITIES LECTURERS DURING THE COVID-19 PANDEMIC." International Journal of Mechanical and Production Engineering Research and Development (IJMPERD)10.3, Jun 2020, 1709-1714
[15] A. J. Park and H. H. Tsang, “Phishing WebsiteDetection Framework Through Web Scraping andData Mining,” pp. 680–684, 2017.