The Academic
International Journal of Multidisciplinary Research

An Online Peer Reviewed / Refereed Journal
Volume 3 | Issue 1 | January 2025
ISSN: 2583-973X (Online)
Website: www.theacademic.in

Web Scraping using Python

Mr. Himanshu Tarale

M.Tech (CS), MIT-WPU, Pune, Maharashtra

Email: himanshu.tarale@mitwpu.edu.in

ARTICLE DETAILS

Research Paper

ABSTRACT

Web Scraping functions as a process to obtain website data through automated extraction which converts it into basic structures including spreadsheets and databases and CSV files. The process of Web Scraping requires complex operations while demanding significant time and resources because it mostly depends on manual execution. Multiple automated solutions in Web scraping have been developed by previous investigation findings. This paper returns to evaluate the various Web Scraping approaches and categories of tools alongside their zones of application.

DOI : https://doi.org/10.5281/zenodo.14852767

INTRODUCTION

Web scraping functions as a method which extracts data from websites. The data saving function remains inaccessible to users during their web surfing activities on several websites. Manual data copy-paste proves to be both tedious and time-consuming for data extraction. Web Scraping represents automated data extraction methods that operate on websites. Web scraping software which we refer to as web scrapers enables the execution of this event. The web scrapers implement automatic website loading and data extraction functions according to user specifications. The scraping software exists in two versions that either function as unique site-specific tools or operate across any website. Professional and personal usage of Web Scraping exists for a diverse set of applications.

PROBLEM STATEMENT

Web scraping is a valuable tool for data extraction, but it faces significant challenges due to the variability in the Document Object Model (DOM) structure of web pages. Despite consistent visual content, underlying DOM differences can lead to inconsistent scraper behavior and data extraction failures. These issues require careful planning and adaptation to ensure accurate and reliable data collection.

OBJECTIVES

The primary objective of this report is to explore the applications, challenges, and methodologies of web scraping, with a focus on its use in various industries such as eCommerce, marketing, real estate, and machine learning. Specifically, this report aims to:

1. Introduce the concept of structured data and its importance in modern data-driven decision-making.
2. Discuss how data can be extracted from web pages and the common techniques involved in web scraping.
3. Provide real-world examples of how web scraping is utilized across different sectors like eCommerce for price monitoring, marketing for lead generation, real estate for property data collection, and machine learning for training datasets.
4. Identify the challenges and solutions involved in effective web scraping, including issues related to data consistency and the Document Object Model (DOM).
5. Propose methods for overcoming common scraping challenges to improve the accuracy and efficiency of data extraction.

LITERATURE REVIEW

Big data analytics has become a crucial tool for organizations to analyze vast datasets, uncover new insights, and drive business performance. As data availability on the web increases from various sources like social media, websites, and online platforms, the importance of extracting useful knowledge from this data is paramount. In this context, web scraping has emerged as a fundamental

technique for mining web data. This paper aims to provide a comprehensive review of advanced web scraping methods, technologies, and their applications across different sectors. (Sirisuriya, 2015)

The literature on web scraping begins with an exploration of its design and various applications. Sirisuriya (2015) provides a comparative study of web scraping techniques, highlighting the different methods employed for extracting data from websites. The applications of web scraping span multiple industries, including eCommerce, marketing, and real estate, offering a wide range of data collection opportunities. (Sirisuriya, 2015)

Web scraping methods are diverse, and various technologies have been developed to optimize data extraction. Phan (2019) discusses the development of applications powered by web scraping, exploring how businesses can leverage this technology to gather and analyze online data. Additionally, Rahmatulloh and Gunawan (2020) focus on the use of the HTML DOM method for scraping scientific articles from platforms like Google Scholar, illustrating the specialized approaches for academic data collection. (Phan, 2019; Rahmatulloh & Gunawan, 2020)

The process of developing a web scraper is also explored in the literature, with several tools and techniques being presented for effective data extraction. Resources such as Python tutorials and JavaScript DOM navigation (W3Schools, n.d.; Python Software Foundation, n.d.) provide foundational knowledge for building scrapers and understanding how web data is structured. GeeksforGeeks (n.d.) further explains the practical steps and tools needed to implement web scraping projects, making it accessible to developers and researchers alike. (W3Schools, n.d.; Python Software Foundation, n.d.; GeeksforGeeks, n.d.)

In conclusion, web scraping is an essential tool for data extraction and analysis in the modern, data-driven world. By reviewing various methods and technologies, this paper equips scholars and business professionals with the necessary knowledge to harness the power of web scraping for their specific needs. (Sirisuriya, 2015; Phan, 2019)

PROPOSED METHODOLOGY

Web Scraping refers to an automatic system which extracts data and collects information from websites on the worldwide internet. A few websites employ strategies to stop bots from viewing their pages through web scraping detection systems. The implementation of web scraping relies on DOM (Document Object Model) and computer vision and natural language processing through techniques

which enable the simulation of human browsing to collect web page content for offline parsing. Web scraping solutions today operate from basic ad-hoc manual work to complete automated systems which transform whole websites into structured information while encountering system constraints.

The requests library functions as the default tool for HTTP requests in Python programming language. This system removes complex HTTP request logic through its easy-to-use API so users can handle services while processing application data.

Selenium WebDriver functions as a web framework to allow browser testing execution. The tool executes web-based application testing through automation which ensures expected functionality. The Selenium WebDriver provides users the flexibility to select their programming language for developing test scripts. Selenium WebDriver represents an updated form of Selenium Remote Control which provides solutions to multiple constraints. Selenium WebDriver lacks capabilities to interact with windows however users can implement the solutions provided by tools such as Sikuli and Auto IT to address this limitation.

Beautiful Soup Library enables Python programmers to parse HTML and XML documents including tag soups from malformatted markups (hence its name "tag soup"). The parse tree made by the system allows users to extract data from HTML pages through web scraping operations. Beautiful Soup maintains its development under the leadership of Leonard Richardson who continues to develop the project while the organization Tidelift provides paid open-source maintenance services.

MongoDB serves as an open-source document-oriented database maintaining large-scale data storage while providing simplified access to the data through its efficient features. The MongoDB belongs to the NoSQL category since its documents do not follow traditional relational table-based patience and retrieval models.

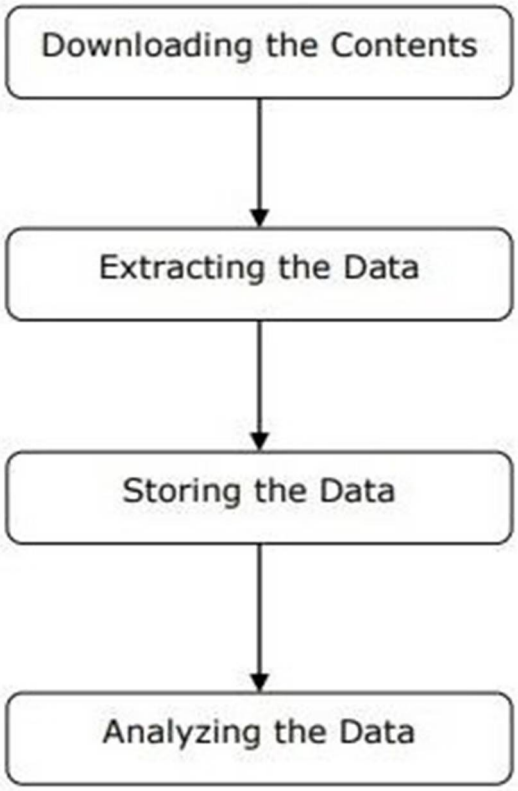
Embedded JavaScript Templating stands as one of the most popular template engines for JavaScript with the name EJS. Through its name it reveals how you can embed JavaScript code inside template language which creates HTML output.

WORKING PROPOSED SYSTEM

The information obtained above requires modifications before direct usage becomes possible. The source material requires cleaning procedures in order to be suitable for deployment. String manipulation and regular expression represent two methods that work effectively for this task. Extraction and transformation steps may run together as one unit.

Storage Module: The data needs processing according to our storage requirements before saving it. A standard output format from the storage module enables data storage either in database or JSON or CSV formats.

Working of a Web Scraper: Web scraper functions as a software which downloads web page content from multiple sites while extracting needed data from that information.



SYSTEM IMPLEMENTATION AND TESTING

Sr No.	Name of theResources	Specification
1.	Hardware :ComputerSystem	Computer (I3-I5 Preferable) RAM minimum4 GB and Onwards

2.	OperatingSystem	Windows 10-11
3.	DevelopmentSoftware	Python IDE,EJS, Visual Studio
4.	Libraries	Py mongo, Selenium, Beautiful Soup

Table : SETTING ENVIRONMENT

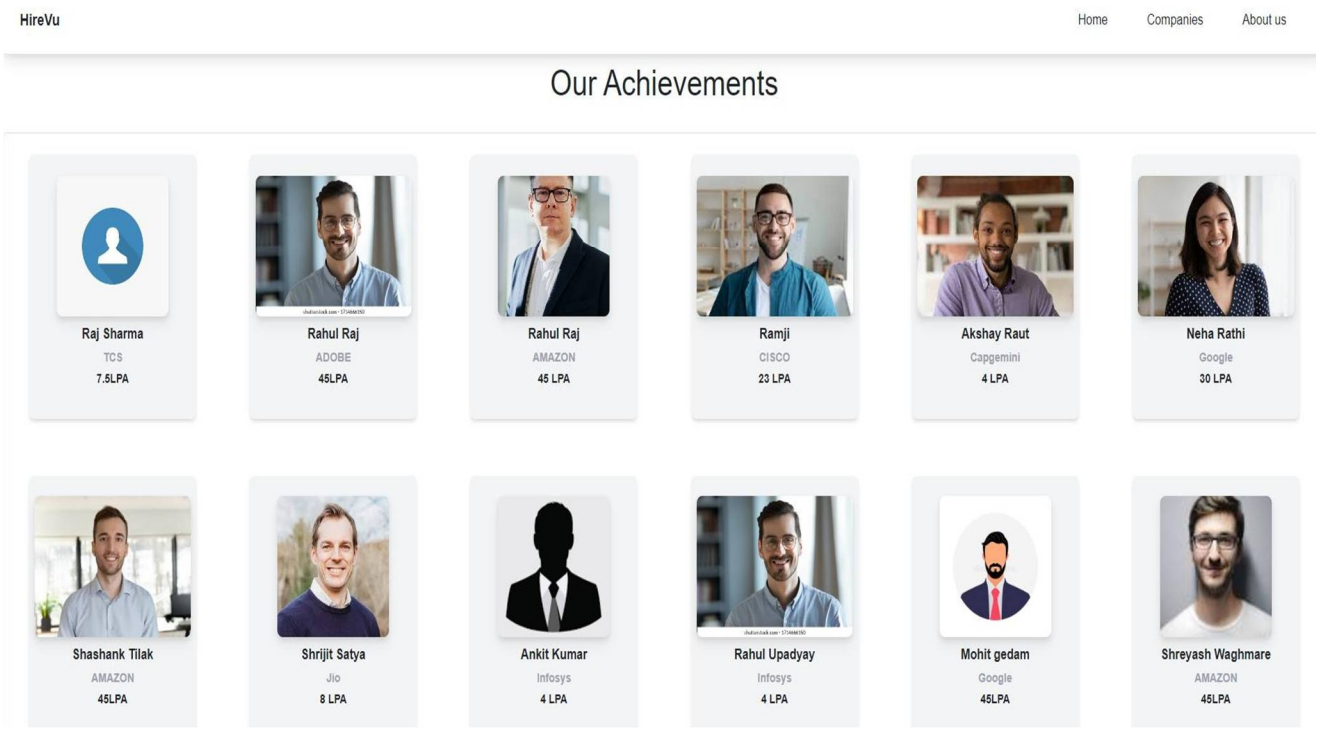
SYSTEM EXECUTION DETAILS

Module 1: Home Page

The Home Page serves as the main entry point, offering quick access to essential sections of the website.

- **Company Overview:** Briefly introduces the company's mission and values.
- **About Us:** Provides detailed information about the company's history and goals.
- **Achievements:** Highlights key milestones and recognitions to build credibility.

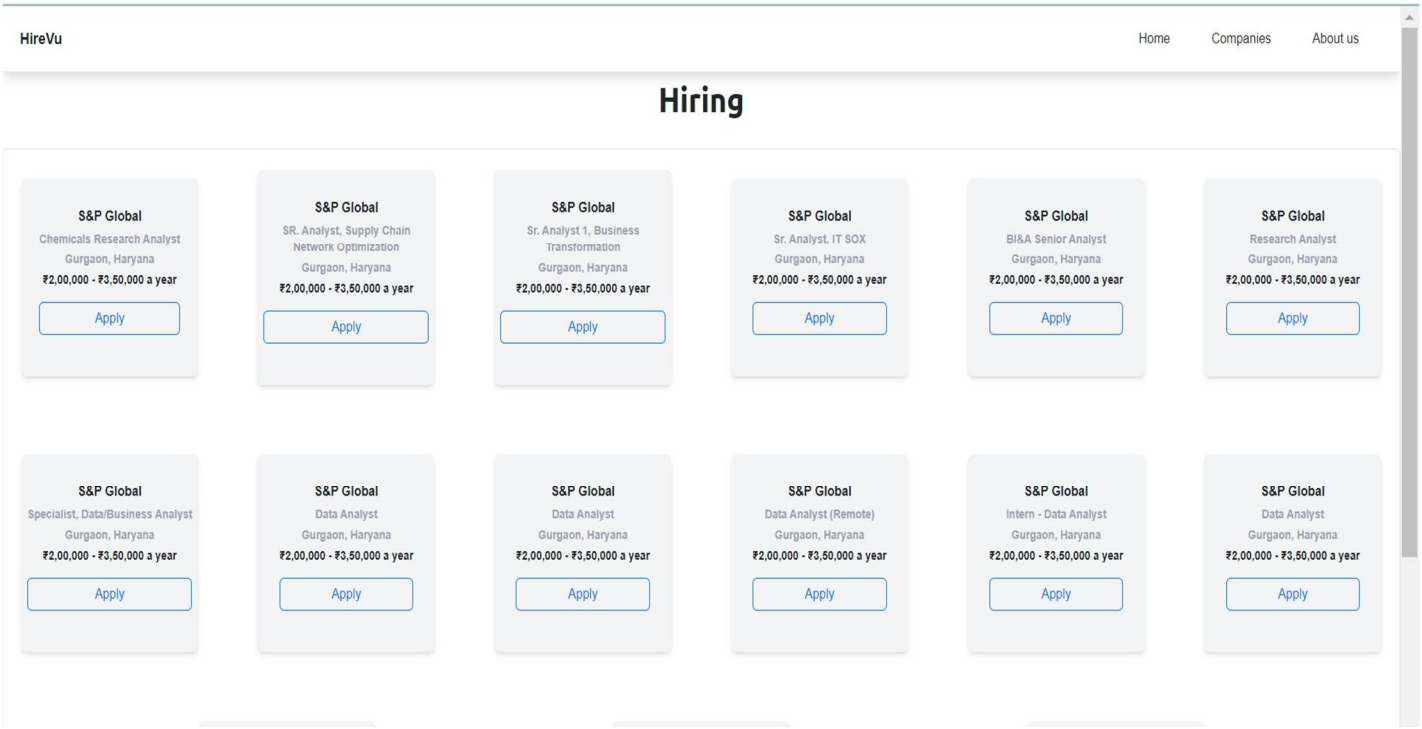
The page features clear navigation links and a clean layout, ensuring an intuitive user experience and easy access to other sections.



Module 2: Companies Page - Apply Tab

The **Apply Tab** on the Companies Page is designed for job seekers to easily apply for open positions within the company.

- **Job Listings:** Displays available job positions with key details like job title, location, and requirements.
- **Application Form:** Allows users to submit their resume, personal details, and cover letter directly through the form.
- **Easy Navigation:** Provides clear instructions and a user-friendly interface for seamless submission.



- This module helps streamline the application process, offering a straightforward path for potential employees to apply for jobs.

Module 3: About Us

The **About Us** section provides visitors with an in-depth overview of the company.

- **Company Overview:** Highlights the company's history, mission, vision, and core values.
- **Team & Leadership:** Introduces key leadership members, their roles, and contributions to the company's success.
- **Culture & Values:** Focuses on the company's work culture, ethical standards, and commitment to innovation and excellence.

This module is designed to build trust and help visitors understand the company's background, values, and its impact in the industry.

Module 4: Add Student

The **Add Student** module allows administrators or authorized users to input new student information into the system.

- **Student Information Form:** Provides fields for entering essential details like name, contact information, course enrollment, and academic details.
- **Submit Functionality:** Once the form is completed, the data can be submitted to the database for processing and record-keeping.
- **Validation:** Ensures that all required fields are filled out correctly before submission, preventing incomplete or inaccurate data entry.

This module helps streamline the process of adding and managing student information, making it easy to maintain an up-to-date student database.

Microsoft Store

Module 4: Add new Company Hiring

ADVANTAGES

- *Cost-Effective*

The essential web scraping services come with price packages that support customers through every stage of their operations. Websites need to provide data retrieval services together with analysis for maintaining regular internet operations. Web scraping services carry out this task at both affordable prices and efficient rates.

- *Low Maintenance and Speed*

During its operational period Web Scraping requires minimal maintenance costs. Web scraping allows organizations to set accurate budget estimates. The web scraping procedure performs daily manual tasks more efficiently because it requires only several hours to complete.

- *Data Accuracy*

Data extraction errors of any nature might eventually trigger extensive problems for companies. The quality of obtained data must be verified because accuracy represents a critical requirement. The accuracy of data scraping matches its fast nature. This special reputation enhances the ability to extract essential data types including sales price alongside financial data.

- *Easy to Implement*

Users have no reason to worry when website scraping services begin their operation since they automatically gain full domain data. A one-time spending on the system enables access to massive amounts of data.

Conclusion

This study reviews recent literature on web scraping applications, techniques, and tools. It reveals that while many web scrapers are designed for general tasks, Scrapy stands out due to its speed, extensibility, and asynchronous request handling. Its architecture, based on web crawlers, makes data extraction easier, and its use of CSS and XPath selectors ensures precise scraping. Scrapy is ideal for complex projects, especially when integrating with VPNs and proxies. Additionally, tools like Scraper API simplify the process by supporting browsers, proxies, and CAPTCHAs, making it an excellent choice for efficient web scraping.

Future Work

1. **Enhanced Web Scraper and User Interface:** Future work can focus on developing an automated web scraper with a user-friendly interface, allowing users to easily select the data they wish to scrape. The scraper can be expanded to handle various data types and improve its capabilities, such as scraping additional content from platforms like TimeEdit.
2. **Data Mining and Analysis:** Future research can explore applying data mining methods to analyze larger datasets, potentially spanning a year's worth of data. This can provide insights into student success and course arrangements. The development of a platform where users can input links for web scraping, coupled with data mining techniques, could offer valuable statistics for program improvement.

REFERENCES

1. https://en.wikipedia.org/wiki/Web_scraping

2. https://www.geeksforgeeks.org/what-is-web-scraping-and-how-to-use-it/

3. https://docs.python.org/3.9/tutorial/index.html

4. https://www.w3schools.com/js/js_htmldom_navigation.asp

5. Sirisuriya, D. S. (2015). A comparative study on web scraping. In the Proc. 8th Int. Res. Conf. KDU, 135– 140.

6. Phan, H. (2019). Building Application Powered by Web Scraping. Doctoral Thesis
7. Rahmatulloh, A. and Gunawan, R. (2020). Web Scraping with HTML DOM Method for Data Collection of Scientific Articles from Google Scholar. Indonesian Journal of Information Systems, 2(2):95-104.