A

Project Report

on

Indeed Job Vacancies (Web Scrapping)

Submitted by

Priyesh Patil	**S190614308**
Mayur Autade	**S190614212**
Mangesh Wagh	**S190614358**
Lalit Patil	**S190614303**
Shubham Bhardwaj	**S190614219**

Under Guidance of

Prof. Tanvi Deshmukh

Sandip Foundation's

Sandip Institute of Technology and Research Centre, Nashik

Department of Computer Engineering

AY 2021-22

Index

Abstract:-

Everywhere you look nowadays there are machines doing things for humans. A lot of things are being automated very easily with the help of the development of technology and production.
We just need all of it right now and why wouldn't you make it easier for yourself too?

That is exactly what **Web Scraping** is about. It is a term used for getting data from Web Pages online.
Once you get the data you desire, you can do a lot of things with it and that is up to you.
Web scraping is a method of extracting and restructuring information from web pages. This project will introduce basic techniques for web scraping using popular open-source tools.

- The first part of the project will provide an overview of basic HTML elements and Python tools for developing a custom web scraper.
- The second part will enable us to practice accessing websites, parsing information, and storing data in an Excel file.

Problem Statement

Say you're a surfer, both online and in real life, and you're looking for employment. However, you're not looking for just *any* job. With a surfer's mindset, you're waiting for the perfect opportunity to roll your way!

There's a job site that offers precisely the kinds of jobs you want. Unfortunately, a new position only pops up once in a blue moon, and the site doesn't provide an email notification service. You think about checking up on it every day, but that doesn't sound like the most fun and productive way to spend your time.

Thankfully, the world offers other ways to apply that surfer's mindset! Instead of looking at the job site every day, you can use Python to help automate your job search's repetitive parts. **Automated web scraping** can be a solution to speed up the data collection process. You write your code once, and it will get the information you want many times and from many pages.

Gathering Requirements

WEB SCRAPING:-

Web scraping is a method used to get great amounts of data from websites and then data can be used for any kind of data manipulation and operation on it.

For this technique, we use web browsers. You usually do not have the built-in option to get the data you want. That is why we use Web Scraping to automate the process of getting that data and not having to do it manually. Web Scraping is the technique of automating this process instead of manually copying the data from websites.

Is Web Scraping Legal?

Talking about whether web scraping is legal or not, some websites allow web scraping and some don't. To know whether a website allows web scraping or not, you can look at the website's "robots.txt" file. You can find this file by appending "/robots.txt" to the URL that you want to scrape. For this example, I am scraping the Flipkart website. So, to see the "*robots.txt*" file, the URL is www.flipkart.com/robots.txt.

Benefits and Usages of Web Scraping

If you can get the data from websites just imagine what you can make. Data manipulation is key here.

Here are some examples:

Analysis: Gather data and make an Analysis Tool, which tracks your data. You can use this method for research. Maybe even predict behavior with Machine Learning or more complex ideas

Price compare: Get prices from different websites and compare them to get an overview of the market that way you can save money!

Email lists: Collect email addresses for marketing and promotions. There are so many emails you receive daily from companies you never even heard of, well that's how.
Jobs: Searching for a job can get hard because of the listings being spread on different websites, which is confusing
Data for Machine Learning Projects: Retrieval of data for machine learning projects depends upon web scraping

Challenges of Web Scraping

The Web has grown organically out of many sources. It combines many different technologies, styles, and personalities, and it continues to grow to this day. In other words, the Web is a hot mess! Because of this, you'll run into some challenges when scraping the Web:

- **Variety:** Every website is different. While you'll encounter general structures that repeat themselves, each website is unique and will need personal treatment if you want to extract the relevant information.
- **Durability:** Websites constantly change. Say you've built a shiny new web scraper that automatically cherry-picks what you want from your resource of interest. The first time you run your script, it works flawlessly. But when you run the same script only a short while later, you run into a discouraging and lengthy stack of traceback!

Unstable scripts are a realistic scenario, as many websites are in active development. Once the site's structure has changed, your scraper might not be able to navigate the sitemap correctly or find the relevant information. The good news is that many changes to websites are small and incremental, so you'll likely be able to update your scraper with only minimal adjustments.

However, keep in mind that because the Internet is dynamic, the scrapers you'll build will probably require constant maintenance. You can set up continuous integration to run scraping tests periodically to ensure that your main script doesn't break without your knowledge.

An Alternative to Web Scraping: APIs

Some website providers offer application programming interfaces (API) that allow you to access their data in a predefined manner. With APIs, you can avoid parsing HTML. Instead, you can access the data directly using formats like JSON and XML. HTML is primarily a way to present content to users visually.

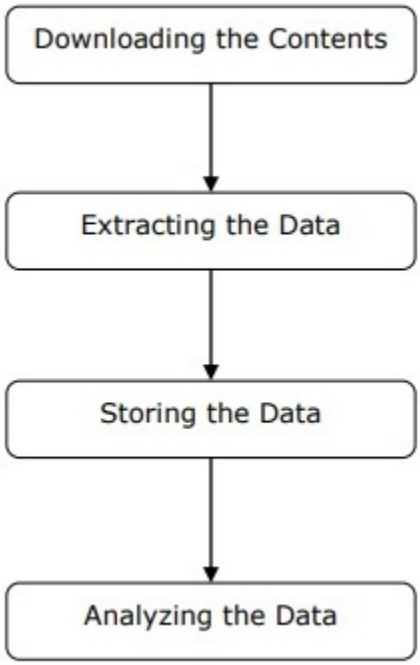
When you use an API, the process is generally more stable than gathering the data through web scraping. That's because developers create APIs to be consumed by programs rather than by human eyes.

The front-end presentation of a site might change often, but such a change in the website's design doesn't affect its API structure. The structure of an API is usually more permanent, which means it's a more reliable source of the site's data.

However, APIs *can* change as well. The challenges of both variety and durability apply to APIs just as they do to websites. Additionally, it's much harder to inspect the structure of an API by yourself if the provided documentation lacks quality.

Design and Modeling

UML Diagram



Technology Used-

- **Python**
- **Some Python Module:-**

    1. **request:-** It is a simple Python web scraping library. It is an efficient HTTP library used for accessing web pages. With the help of Requests, we can get the raw HTML of web pages which can then be parsed for retrieving the data
    2. **BeautifulSoup:-** we are using BeautifulSoup for parsing that HTML data.
    3. **openpyxl:-** By using this we can create Excel file and store data in it.

- **Microsoft Excel**

Implementation /Execution

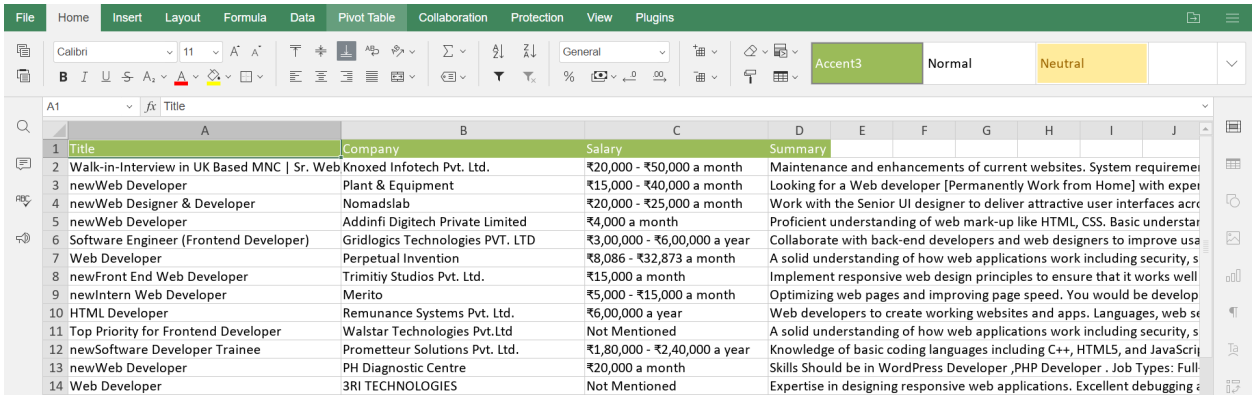
How Do You Scrape Data From A Website?

When you run the code for web scraping, a request is sent to the URL that you have mentioned. As a response to the request, the server sends the data and allows you to read the HTML or XML page. The code then parses the HTML or XML page, finds the data, and extracts it.

To extract data using web scraping with Python, you need to follow these basic steps:

1. Find the URL that you want to scrape
2. Inspecting the Page
3. Find the data you want to extract
4. Write the code
5. Run the code and extract the data
6. Store the data in the required Excel format

Output:



	Title	Company	Salary	Summary
2	Walk-in-Interview in UK Based MNC | Sr. Web	Knoxed Infotech Pvt. Ltd.	₹20,000 - ₹50,000 a month	Maintenance and enhancements of current websites. System requiremer
3	newWeb Developer	Plant & Equipment	₹15,000 - ₹40,000 a month	Looking for a Web developer [Permanently Work from Home] with exper
4	newWeb Designer & Developer	Nomadslab	₹20,000 - ₹25,000 a month	Work with the Senior UI designer to deliver attractive user interfaces acr
5	newWeb Developer	Addinfi Digitech Private Limited	₹4,000 a month	Proficient understanding of web mark-up like HTML, CSS. Basic understar
6	Software Engineer (Frontend Developer)	Gridlogics Technologies PVT. LTD	₹3,00,000 - ₹6,00,000 a year	Collaborate with back-end developers and web designers to improve usa
7	Web Developer	Perpetual Invention	₹8,086 - ₹32,873 a month	A solid understanding of how web applications work including security, s
8	newFront End Web Developer	Trimitiy Studios Pvt. Ltd.	₹15,000 a month	Implement responsive web design principles to ensure that it works well
9	newIntern Web Developer	Merito	₹5,000 - ₹15,000 a month	Optimizing web pages and improving page speed. You would be develop
10	HTML Developer	Remunance Systems Pvt. Ltd.	₹6,00,000 a year	Web developers to create working websites and apps. Languages, web se
11	Top Priority for Frontend Developer	Walstar Technologies Pvt.Ltd	Not Mentioned	A solid understanding of how web applications work including security, s
12	newSoftware Developer Trainee	Prometteur Solutions Pvt. Ltd.	₹1,80,000 - ₹2,40,000 a year	Knowledge of basic coding languages including C++, HTML5, and JavaScri
13	newWeb Developer	PH Diagnostic Centre	₹20,000 a month	Skills Should be in WordPress Developer ,PHP Developer . Job Types: Full
14	Web Developer	3RI TECHNOLOGIES	Not Mentioned	Expertise in designing responsive web applications. Excellent debugging a

Outcome

- **Time-saving:** For gathering the required data on a particular website
- **Easy access:** Only run the Program and get automatically Excel generated.
- **Easy Sharing Data:** It automatically creates files that file you can share with anyone.
- **Fewer errors:** It automatically creates files that why it has fewer errors.

References

- https://www.tutorialspoint.com/python_web_scraping/python_web_scraping_tutorial.pdf
- https://www.geeksforgeeks.org/scraping-indeed-job-data-using-python/
- https://www.edureka.co/blog/web-scraping-with-python/