2023 3rd Asian Conference on Innovation in Technology (ASIANCON)
Pune, India. Aug 25-27, 2023

Food Genie, Recipe Search Algorithm using Web Scraping

Nikhil Suwalka
Master of Computer Application Department
Sardar Patel Institute of Technology
Mumbai, India
nikhil.suwalka@spit.ac.in

Nishit Shanbhag
Master of Computer Application Department
Sardar Patel Institute of Technology
Mumbai, India
nishit.shanbhag@spit.ac.in

Prof. Sakina Salmani
Master of Computer Application Department
Sardar Patel Institute of Technology
Mumbai, India
sakina_shaikh@spit.ac.in

Dr. Pooja Raundale
Master of Computer Application Department
Sardar Patel Institute of Technology
Mumbai, India
pooja@spit.ac.in

2023 3rd Asian Conference on Innovation in Technology (ASIANCON) | 979-8-3503-0228-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ASIANCON58793.2023.10270597

***Abstract*— Coping up with our busy lifestyle and unhealthy eating habits, we are compelled to eat diet foods containing specific ingredients and recipes. It's not easy to ask the chef to make a range of recipes that contain the required ingredients. There are several recipe websites containing a wide range of recipes to choose from. This allows the chef to make new meals containing the specified calories and other nutritional values. The challenge is to find a list of all recipes containing the required ingredients along with the maximum calories allowed without having any data in-hand. This paper presents a method to get a list of all recipes along with its details like methods, ingredients, nutritional information which contains all or some of the mentioned ingredients. The method will allow users to search for recipes easily without worrying about the ingredients. Web scraping using Scrapy and Neo4j graph database is used to implement the algorithm. Web scraping takes the contents of a web page and saves them in a Neo4j graph database using the Python scrapy package. This database will be used to do more study on the food genie application in order to improve search results. Chefs may use the online app to prepare a variety of recipes using particular components to maintain a healthy lifestyle.**

Keywords*— *web scraping, web extracting, Smart chef

I. INTRODUCTION

Cooking is a skill that may assist you in maintaining a healthy lifestyle. Depending on the availability of ingredients or the medical practitioner's diet plan, stepping into the kitchen conjures up a myriad of meals. Chef's look for recipes of different cuisines that other chefs have appreciated on the internet. They go through a number of websites throughout their search which provides a wide range of recipes along with its details. It will be tough for them to locate recipes that include the identical products and do not contain the ones indicated if they need to follow a diet plan to stay healthy.

Web scraping has been used to collect data from a wide range of websites. To automatically extract information from various web pages, web scraping is used. Web scraping parses hypertext elements and obtains text in required format from huge amounts of web data. If you come across data on the internet that you can't immediately download, web scraping allows you to extract the information needed by scraping the web page. It provides you with a format you can use easily.

Web scraping strategies are divided into six stages. (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 format.

This study proposes an algorithm for extracting recipes based on ingredients. Using this data, the cook may quickly choose a recipe without having to double-check all of the components that must be included and/or eliminated. The end product is a web application that interacts with the user by allowing them to search for ingredients to add and exclude. This introduction is the first of four sections that make up the paper. Section 2 describes the food genie web app's suggested web scraping algorithm. In Section 3, the outcome is addressed. Finally, based on the suggested method, Section 4 summarises the work.

Finally, Section 4 brings the article to a close by summarising the suggested algorithm and the acquired findings.

II. RELATED WORKS

Because there is only one study related to web scraping for chefs in the existing literature, this section contains information about web scraping extraction.

Shilpa Chaudhari et al.[1] proposed a scraping algorithm for scraping recipes based on their name or ingredients and storing them in a MongoDB database. It just saves the recipe's name, ingredients, and URL. It scrapes all of the recipes from the web once and saves them in the database, so that when a user searches for a recipe, it will look for that in the database.

Malik et al. [2] discussed how to use web mining/scraping to collect data from various websites, which might be beneficial for Food Genie application in extracting ingredients from recipes. The parsing of websites method uses specifically built algorithms to extract data from a website's HTML and transform it to a different format. The practice of collecting and storing web data in a structured database or spreadsheet is known as web scraping.

Chaulagain et al. [3] proposed a cloud-based scraping architecture for extracting unstructured data from the web using modules such as Selenium, Scrapy, the web driver API, BeautifulSoup and Python's HTMLParser module. The

Selenium and web driver API technologies were chosen for automated web page data extraction. A VM based on Amazon AWS' Elastic compute cloud is used to construct this cloud-based scraping architecture (EC2).

Mahto et al. [4] used a web crawler, also known as a web spider, which crawls a website from the first page to find links, which is saved in a particular format which stores the link to the original website and can be used to open the webpage.. The Extractor extracts relevant data and changes it to the required format.

Hernandez-Suarez et al. [5] described web scraping approaches for collecting historical tweets over any time period while avoiding Twitter API limitations. Plaintext data is retrieved using hypertext tags. Using Scrapy, an open source framework for extracting data from websites developed in Python, scraped the Twitter Search API and changed search fields to improve searching capabilities for gathering history of tweets inside a specific timeline.

Slamet et al. [6] explained in web scraping with Naive Bayes Classification is utilised in four steps for the job search engine. (1) Specifying HTML documents from the website whose data is scraped in the scraping template. (2) Using information gathered from websites, develop a website navigation exploration system. (2) assessing the site's navigation (3) Navigation and extraction automation: automation of data and information obtained from websites is carried out based on processes 1 and 2.

Kunang et al. [7] presented a web scraping technique for creating a weather dataset by gathering and updating real-time weather data from a variety of websites.

Sundaramoorthy et al. [8] presented NewsOne - An Aggregation System for News, a platform that scrapes and crawls content from multiple news websites. It gathers all of the most recent news stories from a variety of sources and summarises them in a clear and concise manner. It enables service-oriented communication between people all over the internet.

Ujwal et al. [9] proposed an automated system of web-scraping that extracts data from internet pages' repeated blocks. Using a new classification-based technique, each block represents a product-offer object and contains properties such as offer-title, offer-description, offer expiry, and so on.

Dastidar et al. [10] explained that structure-based web scraping technologies must be manually updated. describes a self-contained web scraping system that adapts to changes in structure and extracts data from online pages made up of repeating blocks.

Upadhyay et al. [11] demonstrated the web-scraping architecture, which provides a simple and viable method for collecting learning objects for an eLearning application by parsing and retrieving data from a number of websites on a wide scale without the need of human intervention.

D. M. Thomas et al. [12] showed the technique used in the research is to collect all of the data retrieved from various sources utilising the vivid features of the web crawler scrapy using python scripts and then analyse it according to the customer's needs before storing it in the company's database. The data was retrieved using Web scraping and saved as a csv file.

E. Uzun et al. [13] demonstrated an unique methodology that uses string approaches and other information collected from a website's web pages during the crawling process to provide time efficiency. The average extraction time of the basic method in this technique is roughly 60 times faster than the current extraction time of the Angle- Sharp parser.

Vasvi Bajaj et al. [14] showed the use of a data collection including hundreds of ingredients, this programme offers a range of meals based on any set of ingredients. This programme also assists in the discovery of a list of probable recipes depending on the availability of ingredients with a user. Because the Recipe Recommender System is built on a highly linked network of recipes and their ingredients, a Graph Database was the best option for storing this vast amount of interconnected data.

C. Gu et al. [15] introduced the idea of science and technology (S&T) Web mining, summarises the current application situation and forecasts future application prospects, and examines the technological challenges that must be overcome in the future.

It can be concluded that for extracting data from various web pages, scraping is an effective method, which can then be used for a variety of applications. Web scraping is a very economical and easy to implement technique that can be used in our food genie application to extract ingredients from multiple websites containing various recipes.

III. PROPOSED WEB SCRAPING FOR FOOD GENIE

This paper studies extracting of recipes by providing a list of ingredients which helps chefs to search for recipes with provided ingredients. It also allows users to exclude individual substances, which is useful for those who are allergic to or dislike a particular ingredient.

Initially, a single website is selected, which contains a wide range of recipes from which eleven parameters such as Recipe Name, list of Ingredients used, number of calories, preparation time, image path, cooking time, nutrients, total time, details, directions, and link to the recipe website are retrieved. The information gathered is saved in the Neo4j database. Neo4j is an open-source graph database. Instead of rows and columns, a graph database has nodes, edges, and properties. For many use situations, it is more suited to specific big data and analytics applications than row and column databases or free-form JSON document databases. The process of web scraping is used to transfer web data into a storable format. The chef can design a diet plan for the clients based on the ingredient information.

Scraping recipe websites, storing extracted recipes in databases, retrieving data from recipe databases, and implementing the food genie application are all part of the functionality of this proposed project.

Scraping recipe websites involves getting raw data about recipes from the websites and converting it to required format. Then stores the extracted recipe into Neo4j database - basically stores 11 fields namely recipe name, ingredients, number of calories, preparation time, image path, cooking time, nutrients, total time, details, directions, and link to recipe from the web scraping results. The recipes stored in the database will be used in the food genie app for extracting recipes. The algorithm of food genie takes three data items namely ingredients to include, ingredients to exclude and the

maximum number of calories in the form of a list and provides a list of recipes for mentioned ingredients.

As indicated in Figure 1, the primary block of the algorithm is aimed to achieve efficiency and flexibility in the ingredient/recipe gathering method for food genie. PyCharm and Neo4j Desktop are used in this paper.

The Django framework, which is used for web apps written in Python, makes web scraping or web crawling easier via the scrapy Python module. The Neo4j graph database stores the results.

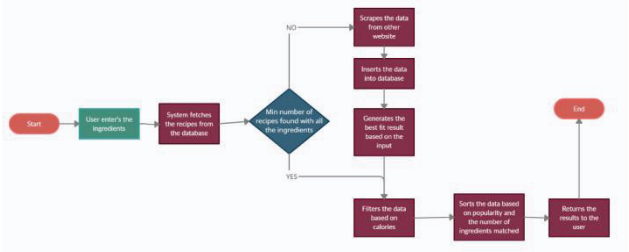

Fig. 1. Web Scraping Block diagram for Food Genie

Scraping: Scraping here refers to getting a list of recipes along with its information from various websites by scraping through them. A website [16] is used in the algorithm proposed for scraping recipes.

Eleven fields namely recipe name, list of Ingredients, number of calories, preparation time, image path, cooking time, nutrients, total time, details, directions, and the link of the original recipe website are accessed from this website. Scrapy library allows us to get the content required in a particular format.

Store results into database: scrapped results are already in the required format. It contains attributes Recipe Name, Ingredients of the recipe, number of calories, preparation time, image path, cooking time, nutrients, total time, details, and link to the recipe which are automatically stored in the Neo4j database for future execution of the algorithm. Database created contains 3 graphs namely Ingredient, Recipe and ScrapedIngredients.

The ingredient Graph stores all the unique ingredients found in recipes, it contains two labels namely id and name (name of the ingredient). When a new recipe is scraped, the ingredients are processed using a Natural Language Processing (NLP) library to remove the stop words and fetch the main ingredient from the ingredient name. For example, "10 small cubes of paneer" is converted to just "paneer". Then the processed ingredient is checked if it already exists or not, if it does not then it is added to the graph. This process helps in making the process faster as more and more recipes will be fetched, more ingredients will already be present in the database which in turn will reduce the processing time.

The Recipe Graph stores all the recipes scraped till now. It contains fifteen fields namely id, calories, cooking_time, details, directions, image_path, ingredients, link, name, nutrients, preparation_time, recipe_id, total_time, view_count. Whenever a new recipe is scraped, the recipe is added to the Recipe graph and then relationships are added to connect it with the ingredients. An is_in relation is added from all the ingredients of the recipe with the recipe. This connection helps in searching for recipes using ingredients as each ingredient can be connected with multiple recipes. The original ingredients are stored in a list in the recipe graph which is shown on the web app.

Getting data from database: the database is searched for existing recipes required for the algorithm.

The algorithm takes three data sets namely the ingredients to include, the ingredients to exclude and the maximum number of calories, in the form of a list and outputs a list of recipes for mentioned input.

The exclude part is optional, the users can just provide the ingredients to include, and it will give the results.

The result will consider all the 3 inputs while fetching the results.

The algorithm creates a hash table with keys starting with the count of the total number of ingredients and values with the recipes found with those ingredients to count of 1 ingredient and the recipes with at least 1 count. There is a threshold variable which is set to 7 as default which means if at least 7 recipes are not found with all the ingredients, then the scraping will start to get more recipes. The result will be in the descending order of the count of requested ingredients present in the recipe. A sorting option is also provided which has two more options namely Views which sorts in descending order based on the total views on the recipes and Calories which sorts in ascending order based on the total calories of the recipes.

The algorithm is being implemented by an interactive web application which allows the user to input ingredients of his/her choice to include and exclude them and can use the slider to set the maximum number of calories and the information of those recipes can be obtained in the form of a list of recipes. The recipes can be opened which will display all the information like the description, directions etc.

The database model objects created to store the recipe details along with its data types are given in Table 1.

TABLE I. DATABASE OBJECTS INVOLVED IN STORING RECIPES FOR FOOD GENIE

nodeType	**nodeLabels**	**propertyName**	**propertyTypes**	**mandatory**
:`Recipe`	[Recipe]	name	[String]	TRUE
:`Recipe`	[Recipe]	calories	[String,Double]	TRUE
:`Recipe`	[Recipe]	recipe_id	[Long,String]	TRUE
:`Recipe`	[Recipe]	preparation_time	[String]	TRUE
:`Recipe`	[Recipe]	image	[String]	FALSE
nodeType	**nodeLabels**	**propertyName**	**propertyTypes**	**mandatory**
:`Recipe`	[Recipe]	cooking_time	[String]	TRUE
:`Recipe`	[Recipe]	nutrients	[String]	TRUE

:`Recipe`	[Recipe]	view_count	[Long]	TRUE
:`Recipe`	[Recipe]	total_time	[String]	TRUE
:`Recipe`	[Recipe]	details	[String]	TRUE
:`Recipe`	[Recipe]	directions	[StringArray]	TRUE
:`Recipe`	[Recipe]	ingredients	[StringArray]	TRUE
:`Recipe`	[Recipe]	link	[String]	TRUE
:'Ingredient`	[Ingredient]	name	[String]	TRUE
:`Ingredient`	[Ingredient]	ingredient_id	[Long]	TRUE
:'ScrapedIngredients'	[ScrapedIngredients]	combination	[StringArray]	TRUE

Figure 2 depicts the communication between the many objects involved. The recipe method is presented in Algorithm 1, where the counter array consists of 15 recipe names in which the searched components are present, excluded ingredients are not present, and the recipes have fewer calories than the maximum value set.

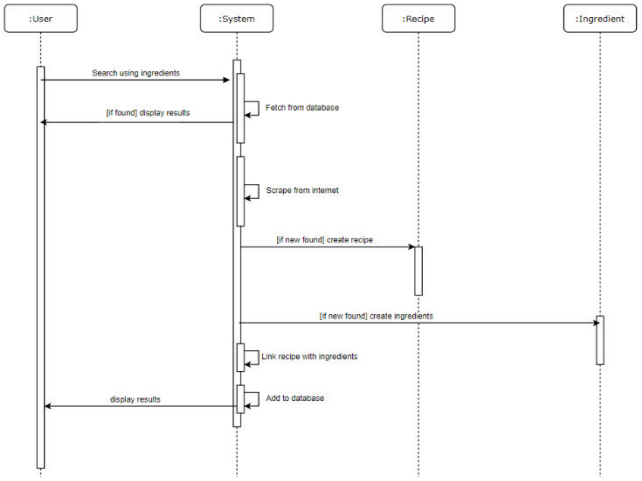

Fig. 2. Objects Communication of Web Scraping for Food Genie

Algorithm 1:

Input: Ingredients to include & exclude and Max calories.
Output: List of recipes based on the input.
included_ingredients: The ingredients to be included
excluded_ingredients: The ingredients not to be included
all_ingredient_recipes: minimum number of recipes with all ingredients
return_recipe_count: number of recipes to be returned
calorie_slider: maximum number of calories

```
recipes_dict_count = getRecipesFromIngredients(included_ingredients)
objects_excludes = getRecipesFromIngredients(excluded_ingredients)

i = 0
while i < len(objects):
    if list(objects.keys())[i] in objects_excludes:
        del objects[list(objects.keys())[i]]
    else:
        i += 1

counter = {}
for i in recipes_dict_count.items():
    if i[0] in objects:
        counter[i[1]] = counter.get(i[1], []) + [i[0]]

counter = sorted(counter.items(), key=lambda x: x[0], reverse=True)

if len(counter) < all_ingredient_recipes:
Start scraping for more recipes

Return the counter to get the output
```

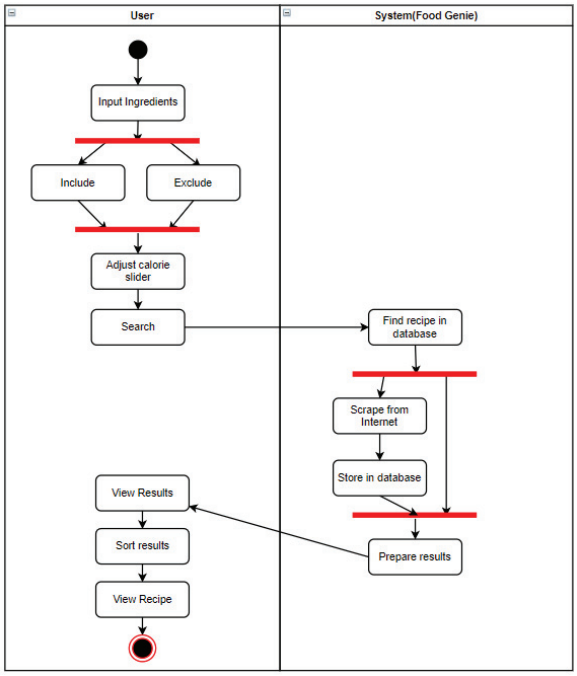

Fig. 3. Activity diagram depicting the various activities.

IV. EXPERIMENTAL ANALYSIS

The findings of web scraping recipe websites and web app results for ingredient-based recipe listing for Food Genie based on inputs are discussed in this part.

The web app home page of our website shows a calorie slider which can be used to set the maximum number of calories, a text field in which ingredients can be entered, the ingredients can be set as include or exclude. A Search button to get the results.

Finally, based on the ingredients, the results are obtained. Figure 4 shows the list of recipes found based on the search criteria. Each recipe item in result shows the recipe name, image, and a few lines of details. It also shows the number of calories the recipe contains and the number of views on that recipe. On top right, there is an option to sort the recipes

based on 3 filters. Clicking on the list item will take us to the recipe page with much more information about it.

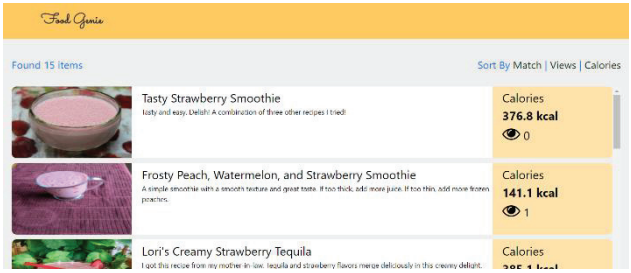

Fig. 4. Web Scraping result

When a user clicks on a recipe, a new page will open which will show more information about the recipe like ingredients, directions, nutritional information, and preparation & cooking time etc. as shown in Figure 5.

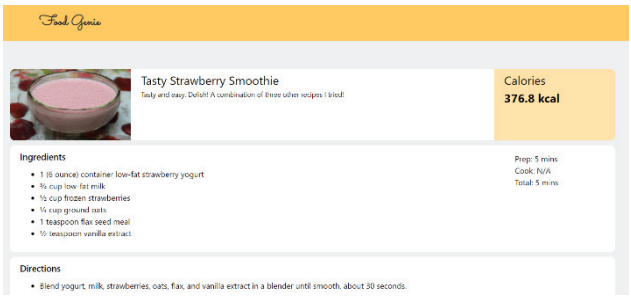

Fig. 5. Recipe information page

To analyse the performance of any animation is Frame Rate which is measured in FPS and the frequency (rate) at which images called frames appear on a display consecutively. DevTools is used to calculate FPS taken by recipe-search is given in Figure 6 and 7 where the Summary tab provides accurate details based on recipe requests, load time, scripting time and the data transfer between applications. Figure 6 Shows the time taken when scraping is required and figure 7 shows the time taken when scraping is not required.

The application spends the majority of its time on scripting and idle state, where rendering refers to transferring data among its components and idle refers to the user not actively interacting within that time frame, suggesting successful application execution without the user's input. The Scripting time is high when scraping happens and is low when there is no need to scrape. The time taken when scraping isn't required is 94.092% less than when scraping is required.

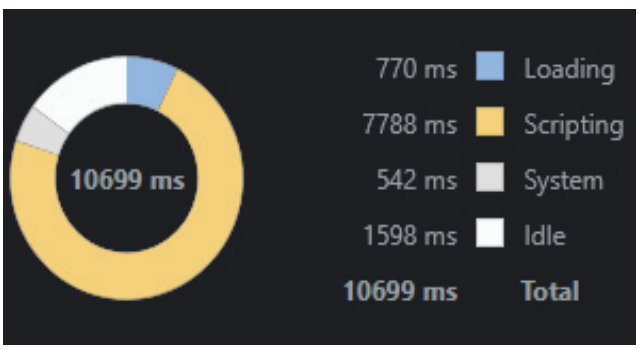

Fig. 6. DevTools CPU chart including scraping time

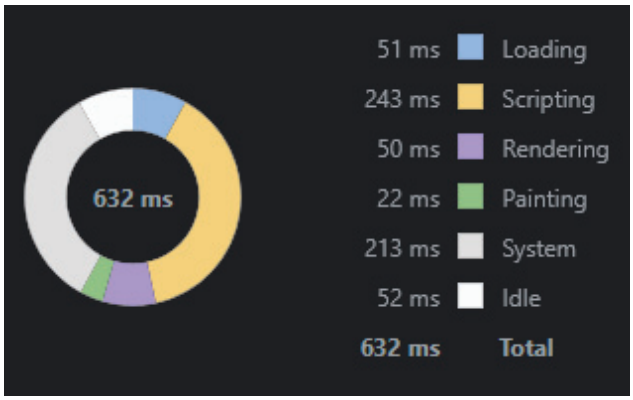

Fig. 7. DevTools CPU chart without the need to scrape

V. CONCLUSION

This paper shows that web scraping, which takes too much time, if used in an efficient manner with a good algorithm can be very useful. It helps to get recipes automatically from one or more websites if required. The results are then formatted and inserted in the Neo4j database and relationships are formed between the Ingredients and the Recipes which makes it faster to retrieve. The users can use this to get recipes with the ingredients they want it to have and without the ones they don't want. This can be further improved with an algorithm to automatically scrap random combinations of ingredients which will improve the database.

A similar algorithm can be developed and used for various use cases which requires getting some kind of data from multiple sources and formating & displaying it. The examples include fetching a list of real estates, ecommerce products, etc. The major changes would include changing the list of source URLs, changing the HTML selectors, formatting the data and adding to the database.

REFERENCES

[1] Shilpa Chaudhari, R. Aparna, Vinay G Tekkur, G L. Pavan and Shreekanth R Karki, 2020, September. Ingredient/Recipe Algorithm using Web Mining and Web Scraping for Smart Chef. In 2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONNECT). IEEE.

[2] Malik, S.K. and Rizvi, S.A., 2011, October. Information extraction using web usage mining, web scraping and semantic annotation. In 2011 International Conference on Computational Intelligence and Communication Networks (pp. 465-469). IEEE.

[3] Chaulagain, R.S., Pandey, S., Basnet, S.R. and Shakya, S., 2017, November. Cloud based web scraping for big data applications. In 2017 IEEE International Conference on Smart Cloud (SmartCloud) (pp. 138-143). IEEE.

[4] Mahto, D.K. and Singh, L., 2016, March. A dive into Web Scraper world. In 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom) (pp. 689-693). IEEE.

[5] Hernandez-Suarez, A., Sanchez-Perez, G., Toscano-Medina, K., Martinez-Hernandez, V., Sanchez, V. and Perez-Meana, H., 2018. A web scraping methodology for bypassing twitter API restrictions. arXiv preprint arXiv:1803.09875.

[6] Slamet, C., Andrian, R., Maylawati, D.S.A., Darmalaksana, W. and Ramdhani, M.A., 2018, January. Web scraping and Naïve Bayes classification for job search engine. In IOP Conference Series: Materials Science and Engineering (Vol. 288, No. 1, p. 012038). IOP Publishing.

[7] Kunang, Y.N. and Purnamasari, S.D., 2018, October. Web Scraping Techniques to Collect Weather Data in South Sumatera. In 2018 International Conference on Electrical Engineering and Computer Science (ICECOS) (pp. 385-390). IEEE.

[8] Sundaramoorthy, K., Durga, R. and Nagadarshini, S., 2017, April. Newsone—an aggregation system for news using web scraping method. In 2017 International Conference on Technical Advancements in Computers and Communications (ICTACC) (pp. 136-140). IEEE.

[9] Ujwal, B.V.S., Gaind, B., Kundu, A., Holla, A. and Rungta, M., 2017, December. Classification-Based Adaptive Web Scraper. In 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA) (pp. 125-132). IEEE.

[10] Dastidar, B.G., Banerjee, D. and Sengupta, S., 2016. An Intelligent Survey of Personalized Information Retrieval using Web Scraper. International Journal of Education and Management Engineering, 6(5), pp.24-31.

[11] Upadhyay, S., Pant, V., Bhasin, S. and Pattanshetti, M.K., 2017, February. Articulating the construction of a web scraper for massive data extraction. In 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT) (pp. 1-4). IEEE.

[12] D. M. Thomas and S. Mathur, "Data Analysis by Web Scraping using Python," 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), 2019, pp. 450-454, doi: 10.1109/ICECA.2019.8822022.

[13] E. Uzun, "A Novel Web Scraping Approach Using the Additional Information Obtained From Web Pages," in IEEE Access, vol. 8, pp. 61726-61740, 2020, doi: 10.1109/ACCESS.2020.2984503.

[14] V. Bajaj, R. B. Panda, C. Dabas and P. Kaur, "Graph Database for Recipe Recommendations," 2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 2018, pp. 1-6, doi: 10.1109/ICRITO.2018.8748827.

[15] C. Gu and L. Huang, "Web Mining in Technology Management," 2008 International Seminar on Business and Information Management, 2008, pp. 88-91, doi: 10.1109/ISBIM.2008.127.

[16] Allrecipes.com, Inc. [Online]. Available: https://www.allrecipes.com/