Auto-mobile Search Engine

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY 4.0

SUBMISSION DATE / POSTED DATE

19-12-2021 / 28-12-2021

CITATION

hameed, mariam; Shahbaz, Raja; Iqbal, Javed (2021): Auto-mobile Search Engine. TechRxiv. Preprint.
<https://doi.org/10.36227/techrxiv.17286794.v1>

DOI

[10.36227/techrxiv.17286794.v1](https://doi.org/10.36227/techrxiv.17286794.v1)

Auto-mobile Search Engine

Mariam Hameed
Department of Computer Science
Iqra University
Islamabad, Pakistan
mariamh997@icloud.com

Javed Iqbal
Department of Computer Science
Iqra University
Islamabad, Pakistan
javed.iqbal.jaferi@gmail.com

Raja Shahbaz
Department of Computer Science
Iqra University
Islamabad, Pakistan
rajashahbaz980@gmail.com

Abstract—The majority of people are looking for vehicles. To find their desired vehicle, people search on the most popular websites. The sellers post their ads on multiple sites to reach out to a larger audience. This means buyers have to find their desired vehicle on multiple sites to compare the prices because buyers always look for lower prices. This search needs to visit and find ads on multiple vehicle entertaining websites, which requires much time. In this paper, we are proposing a unique “Automobile Search Engine” (ASE) which grabs all the automobile ads posted on different websites. Our solution will give a single search for automobile ads. The user can get all the ads based on his/her search criteria. ASE will find all the ads for different websites and will display the results. The user can then compare the searched ads on one single view. ASE supported Multi Thread Crawler.

Index Terms—Web Crawler, Search Engine, Proxy, Multi-thread Crawler

I. INTRODUCTION

An Automobile Search Engine is a web portal that allows users to search automobile-relevant ads. There are already some websites that show ads for automobiles. Buyers will always check some popular websites for different advertisements to find the product they need. It takes too much time to view all the websites and select the choice-related ad that fulfills all the requirements. The ASE makes searching a lot easier for people to search for all ads and later compare them on the same platform. ASE includes all the websites that are most popular among people. All the ads crawl from different websites, such as OLX, PakWheels, Auto Deals etc. The search engine can only show the ads from those websites that are being crawled. This system also provides optimization. The main features of ASE are that users can perform on search results, such as comparison of selected ads, filters to optimize searches, ads details, similar ads, and recent ads. The structure of ASE is shown below:

Fig. 1. Structure of ASE.

II. WEB CRAWLER

Web crawlers are the core of search engines and extract information from websites. The web crawler starts with the provided URLs, these URLs are known as seed URLs. Once the web crawler is started, it can go to the website, and visiting all the links in websites. The process is continue until the crawler is stop. The crawler works this way and obtains the data of the web pages and indexes them in the search engine. A crawler can crawl the data that is not necessary for search engines. After visiting so many links, it happens crawler starts crawling the same links. The crawler needs to stop when necessary. There are the algorithms used how deep the crawler can go. The same crawler does not work for ASE. There are three reasons to not using the crawler that is mostly used by the search engines.

- 1) Websites contain so much information and data that it is not needed.
- 2) All the websites are not only showing automobiles ads. There are different ads posted related to cell phones and many others.
- 3) Single crawler cannot work for every website. Because the content is different.

In ASE, we need the advertisement from the website and the detailed information page of the ads. Therefore, we have to make a crawler that collects the information we want. A crawler can make based on the requirements to get the relevant content from the websites [1]. The crawler needs to run daily to get the latest ads from the website. Some of the websites are updated frequently. The crawler can run using schedulers and fetch new ads every day [4]. The crawler is written in Java programming language as well as multi-threads and proxy.

The Jsoup library is used to fetch data from the website. Jsoup works well with the proxy, and also, the other benefit is quickly finding the attributes and elements of HTML5 and CSS selectors [5]. Every website has different content and ways of representing the data. Hence, a simple crawler is not sufficient for all websites. So, every website that includes ASE has a different crawler. In order to create a crawler, we have to manually inspect the website and parse those mandatory elements. First, the crawler hits the website URL and collects the ad URL. Next, the crawler adds them into a database. The crawler fetches the ad URL from the database and hits the URL to get detailed information. The same process is used to

getting the data from all the websites. All the collected data is stored in the database. The architecture of the crawler is illustrated below in Figure 2

Fig. 2. Architecture of Crawler.

A. Multi Instances

Threads are the simple execution of commands in a sequence. Single thread crawlers cannot meet the requirements because it takes too much time to crawl websites. Hence, in modern search engines, multi-thread crawlers are used to handle big data [3]. Crawlers can get more data from the website in less time using the multi-threads. Multi-thread is used to divide the work into multiple threads. All the threads are worked in parallel. Multi threads are used to reduce the time and to increase performance. For example, let's assume that there are 1 million ads in OLX, Pak Wheel, and Auto-Deals. When we combine these ads, the total sum of these ads is three million. One ad takes one second, so three million ads will take 3,000,000 seconds, 50000 minutes, 833 hours, and 35 days. If we use multi-threading, then one thread handles the one ad, and in 100 seconds, we crawl 100 ads because threads are working in parallel.

B. Avoiding data redundancy

The main problem after implementing the crawler is redundancy. The single-thread access predefined website URL and

get the ad URLs. It happens that the crawler parses the same ad URL more than once and stores it into the database [6]. So, there are many duplicate URLs in the database. To solve this problem, the crawler must check whether the URL already exists in a database. If it exists in a database, do not add a URL, otherwise, add it.

In a single thread, the crawler works fine. After implementing the multi-threaded crawler, the redundancy problem appeared again. Multiple threads get the same URL from the database, and also it never fetches the new URL and traps it in the loop. Each thread that is created needs to insert some value in the database. After insertion, the unique ID is assigned to the thread. Each advertisement link stored in the database table should have a zero value inserted in another column to avoid redundancy. Every thread that retrieves the URL should change the value to 1. So, each new thread should check the value and never retrieves a URL with a value of 1. It cannot happen that the thread fetches the same URL at the same time. The number of threads generated is dependent upon the programmer. The maximum time the threads can take is dependent on internet speed.

III. PROXY

Crawlers use URLs to parse HTML elements. Because the crawler clicks the URL multiple times to get the data, it may be possible that the website can block the IP address. If the IP address is blocked, a crawler can never get the data from the website using the same IP address. There are several ways of content blocking. One of them is URL-based blocking. URL-based blocking means the add the IP address on the block list [7]. Then, it will never allow these blocked IP addresses to access the website. The solution to URL-based blocking is the proxy. Proxy attaches the new IP address to computers before accessing the websites on the internet. This way, the actual IP address cannot be used on the internet. Instead, the proxy available IP address is used to request the resources on the internet. There are many types of proxies. It depends on which type meet the users needs. In ASE, the forwarding proxy type uses for masking the actual IP address. It can hide the true identity of the client's on the internet.

Some proxy services provide an IP address that is dedicated to one client at a time. Just one IP address is not enough, because the website may block an IP address again. The crawler requires more IP addresses. The available public IP addresses on the internet are stored in a database. MySQL is used to save all the data that comes from the crawler. Every time the crawler starts to work, it will change the IP address. To change the IP address, the crawler randomly selects any IP address from the database and uses it. A new IP address is used every time for parsing the website. Therefore, the IP address can never be blocked and tracked by any website. Many users use public IP addresses on the Internet, so the speed may be slow.

IV. SEARCHING

A. Advanced Search

Advanced search is a unique technique offered by many websites. It is a set of filters that refines your searching and removes irrelevant data. This further helps to find the exact content for the user. The user searches for any keyword (such as Honda) on ASE. The results shown on the web page are not according to the user's requirements and want to search for more specific content. Users can filter results by price, model, year, company, category, and color. All the features of advanced search can be applied to any search result. Users want to search ads through price, model, year, company, category, and color; to achieve this, advanced search filters out this specific search, which gives appropriate information to the user according to their desired filter. So, it becomes easy for the user to decide which ad is suitable.

Fig. 3. Showing the Advanced Search.

B. Search Ads

Users can search ads on the search bar and, while typing, quickly view keyword-related suggestions in the search bar. It helps the user to identify the proper title of the desired ads. It will auto-complete the remaining words. This feature develops by using the JavaScript language.

C. Recent Search

A recent search allows users to view information to know which automobiles are popular in the market as it just makes it easy for the user to decide which vehicles are best to purchase. Users search any keyword, all of the keywords are stored in the database. The user can view the recently last five added keywords below the search bar.

V. SEARCH ENGINE RESULT PAGE (SERP)

Search engine result pages are the pages that are shown against any search query done by a searcher. The main feature of SERP is the collection of results in a list that are returned by the search engine in response to a query [2]. The SERP will appear with the following details ad title, description, location, price, website, and model.

Fig. 4. Showing the SERP.

A. Paging

Whenever a user searches for any query, the ASE will show the collection of query-related results on the next web page. On the next page, ten ads will be shown as a result, and there are at least ten pages, and each page has ten ads.

B. Change Search Filters

After showing the complete result of the search. Users can filter out ads based on advanced search features, as mentioned above. This helps the user to find the more specific ads that they want to search.

C. Ad Detail

ASE shows the query results then the user can find more details of ads. The ad detail page includes all the details posted on the website. Users can also view the original ad. Buyers can also check similar relevant ads. Similar ads are shown on the basis of the similarity of the name, company, and year.

Fig. 5. Showing the Detail Page.

VI. COMPARISON BETWEEN ADS

Users can get the best ad, as per requirements, by selecting three to four ads before purchasing the product. After that, they have to compare by viewing multiple ads that are opening in the different tabs. It is a difficult, laborious, and time taking process. This feature helps the user to get a quick comparison. They compare their selected ads on a single view without wasting their time. When users get their searching results, the user can also compare multiple selected ads based on different and same websites. The minimum comparison between ads is two and the maximum comparison is four. The comparison page includes all the details of the ads with the pictures of vehicles.

Fig. 6. Showing the Comparison of Ads.

VII. CONCLUSION

Time is valuable for everyone. People who use this system can save time because they do not need to view advertisements on different websites. They can find ads on a single platform, and users can apply the advanced search filters to refine the search. Filters help the user to easily find out the basis of the ad on some specific conditions. When the user views the ads. He/she can perform a quick comparison on selected ads to identify which vehicle is best for him/her. We use the best and fastest method of multi-threading in crawling. Due to this facility, users can find excellent products in a minimum time.

Data is updated daily through the schedulers in Windows and Cron Job in Linux. So the user stays up-to-date with the latest ads.

VIII. FUTURE WORK

The system mainly focused on getting the data from the website and showing it to the user. Due to a shortage of time, some of the ideas are not incorporated and require further research. The idea that I would like to implement is to update the system by adding more websites and crawling new ads like spare parts, cars on installments, and car accessories, etc. I will use Elasticsearch to auto-complete the keyword and apply filters on the search as Elasticsearch stores, searches, and analyzes big volumes of data in less time.

REFERENCES

- [1] H. Lu, D. Zhan, L. Zhou, and D. He, "An Improved Focused Crawler: Using Web Page Classification and Link Priority Evaluation," in Hindawi Publishing Corporation, 2016.
- [2] "SERP 101: All About Search Engine Results Pages," WordStream. [Online]. Available: <https://www.wordstream.com/serp>. [Accessed: 09-Jul-2021].
- [3] Sun, G., Xiang, H., Li, S.: "On Multi-Thread Crawler Optimization for Scalable Text Searching" Journal on Big Data, 2019, 1, (2), pp. 89–106.
- [4] S. I. N. G. H. A. L. NIRAJ, in Design of a priority based frequency regulated incremental crawler, 2014.
- [5] jsoup Java HTML Parser, with the best of HTML5 DOM methods and CSS selectors. [Online]. Available: <https://jsoup.org/>. [Accessed: 06-Jul-2021].
- [6] L.-K. Soon, "Reducing Redundant Web Crawling Using URL Signatures," in Web-Based Information Technologies and Distributed Systems, S. H. Lee, Ed. Paris, France: Atlantis Press, 2010, pp. 115–140.
- [7] N. *, "Types of proxy: Everything you need to know about.," RapidSeedbox.com, 07-Jul-2021. [Online]. Available: <https://www.rapidseedbox.com/blog/types-of-proxy04>. [Accessed: 09-Aug-2021].

Mariam Hameed received the B.S. degree in Computer Science from the Iqra University at Islamabad in 2020. Her main research interests include computer vision, functional programming languages, and object-oriented programming language etc. In a spare time love to do sketching. Currently working in a Mubeen Maqbool Industries.

Javed Iqbal is the Solutions Architect for FraudLens developing intelligent analytic fraud solutions for FraudLens. Working directly with the FraudLens CTO and CMIO, Javed is a key member of the design team managing development of the FraudLens healthcare fraud solutions. Javed is an accomplished professional with twelve years of experience guiding teams through all phases of software development; from requirements gathering and analysis to design, development, testing, implementation, and support.

His development achievements include incorporating Agile mythology to strengthen test driven development, validation, and change management activities in the development of software solutions. Javed is also the co-author of several award-winning papers on healthcare fraud detection. Prior to joining FraudLens, Javed worked as Lead Software Engineer for five years at Allainet Pvt. Ltd, a leading application, e-commerce and interactive media solution provider. Javed Iqbal received a Masters of Computer Science degree from University of Arid Agriculture, Rawalpindi.

Raja Shahbaz received the B.S. degree in Computer Science from the Iqra University at Islamabad in 2021. His main research interests include programming languages, network security, and data science and big data etc. Currently working in a government agency FIA.