Journal of Knowledge Retrieval and Semantic Systems
Vol. 11, No.41, Winter 2024; PP.199-225.
jks.atu.ac.ir
DOI: 10.22054/jks.2024.80235.1660

ATU PRESS

Original Research
Received: 09, June, 2024 Accepted: 09, August, 2024
ISSN: 2980-8243 eISSN: 2783-1795

A Mechanism to Manage Time and Increase Data Accuracy When Using the Selenium Library

Farnaz Taghizadeh Kourayem — Ph.D. Student in Information Technology Management, Central Tehran Branch, Islamic Azad University, Tehran, Iran

Mohammadreza Kabaranzad Ghadim* — Associate Professor, Department of Industrial Management, Central Tehran Branch, Islamic Azad University, Tehran, Iran

Seyed Abdollah Amin Mousavi — Assistant Professor, Department of Information Technology Management, Central Tehran Branch, Islamic Azad University, Tehran, Iran

Abstract

Today, data, as one of the valuable assets of various organizations and industries, plays an important role in the development and progress of businesses. In fact, every organization uses different sources to collect its data, one of which is the web platform, where a lot of data is produced and published by different users or even robots all over the world every day. Examining, researching, studying and analyzing such data can provide useful information and knowledge for the organization. For this purpose, during the past decades, various tools have been developed that have greatly helped in extracting information from the web platform, among which we can mention Request, Selenium, Scrapy, Beautiful Soup, etc. libraries in the Python programming language. However, each of these libraries faces challenges. In this article, by studying the Selenium library and considering the existence of many challenges in it, we have presented a solution for time management and improving the challenge of its Asynchronous. Our experiments show that the use of the proposed solution increases the accuracy of the information retrieved from the

∗ Corresponding Author: kabaranzad@yahoo.com
This article is taken from the doctoral thesis of Information Technology Management field, Islamic Azad University, Central Tehran Branch

How to Cite: Taghizadeh Kourayem, F., Kabaranzad Ghadim, M.R., Mousavi, S.A.A. (2024). A Mechanism to Manage Time and Increase Data Accuracy When Using the Selenium Library. *Journal of knowledge retrieval and semantic systems*, *11* (41), 199-225. DOI: 10.22054/jks.2024.80235.1660.

web platform and thus improves the challenge of Asynchronous and also reduces the time to retrieve information from the web platform.

1. Introduction

The Internet platform is a very powerful source of information that can be collected with the help of various tools and techniques and used after analysis in order to make better and more efficient decisions. According to previous researchers, when it comes to automatically extracting information from the web, Selenium is always the best option, however, this library has many challenges. One of the challenges of using the Selenium library is its asynchronous and the other is the slowness of the Selenium library, which we are trying to investigate and improve in this article.

Research Question(s): How to improve the challenge of slowness and asynchronous of the Selenium library?

2. Literature Review

Selenium library, which is one of the best web scripting tools, has been used in different studies and with different purposes. This library is a free, open-source automated testing framework used to verify web applications across multiple platforms and browsers. Various programming languages such as Java, C#, and Python can be used to create Selenium test scripts (Teotia et al., 2023). But despite its many advantages, selenium also has disadvantages, including: 1. Slowness, 2. Brittleness, 3. Flakiness, 4. Maintainability, 5. Asynchronous, 6. Time-consuming, 7. Cross-browser, 8. failure analysis, 9. Infrastructure, 10. Scalability, 11. Assertability, 12. Documentation and 13. Support (Leotta et al., 2023).

3. Methodology

The first thing we examined in the Selenium library is the lack of time management. Time management actually refers to the fact that the waiting time for downloading information from the web platform in this library is not known. To solve the problem of slowness and asynchronous of the Selenium library, we have used a solution that includes three different steps:

Step 1) According to the manual checks, first we define the variable t1:

$$t1 = 0.5$$

where t1 is the value used in the sleep function (time required to open the main page in normal mode).

Step 2) We use the while loop and try, except inside it. If the page does not open after 3 seconds, or after 6 attempts with different sleep times, the desired page and products do not appear, the error of the site not being available or the internet being slow will be printed:

```
t1 = 0.5
try:
        while t1 <= 3:
                try:
                        driver.get("""The Study site""") # to open the page

                        time.sleep(t1) # Waiting for the page to open
                        …
                        # Code related to page scrolling
                        time.sleep(t1) # Waiting for information to be displayed after scrolling
                        # Information collection codes
                except:
                        t1 += 0.5
Except:
            print('The internet speed is very slow or the intended site is down')
```

In this code, the program with time t1=0.5 first tries to display the page information in full and if it fails, it adds half a second to t1 and this repetition continues up to 6 times. If the page is displayed in full, we use the new value of t1 for the next pages.

Step 3) If the page opens, we will enter the third step, which is related to collecting the basic information of the products and we must avoid the problem of asynchronous.

At this stage, according to the existence of five different types of information (such as product name, price, amount of discount, price after discount, type of discount) of each product, we first define five different and empty lists for product information.

Then, with the help of commands related to information collection, we take the information about each product and put it in the corresponding list. Then we run the following script to prevent wrong items in the list:

```
check1 = [[product_prices], [product_off], [product _prices2], [Product_off_type] ]

for j in range(0,4):
    if len(product_name) < check1[len[j]]:
        product_name.append('error')
    elif len(product_name) > check1[len[j]]:
        check[j].append('error')
```

In other words, after taking the information of a product and adding them to the predefined main lists, the program calculates the number of items in the existing lists with the help of the len function and puts them in the checklist. Then, with the help of the for loop, the length of each list is compared with the rest of the lists, and if the number of items in a list is low, the word "error" is added to it.

4. Results

In order to evaluate the solutions presented in this research, we have reviewed the information related to supplementary drugs on the DigiKala site, which was almost 3000 different medicines, 13 times on 13 different dates, from September 23, 2023, to March 15, 2024. In this study, the codes written in order to retrieve information from the mentioned site were executed the first time with the proposed solution and the second time without the proposed solution, and in each execution, both the time of information retrieval and the number of information related to each list or the same column was recorded and compared with each other, and the error rate and its percentage were calculated based on the difference in the time of information collection in the first and second execution and the difference in the number of information collected for each column in the first and second execution. After the implementation and use of the proposed solution, the investigations show that the accuracy and correctness of the collected information have increased compared to not using the proposed solution, and the time of information collection has also improved.

5. Discussion and Conclusion

In this article, we studied and evaluated the challenges of being slow, time-consuming and asymmetric of the Selenium library. Our studies were conducted using the Python programming language. Studies show that it is very important to use the solution of checking the list and the

same length of the list at the end of the collection of each product from the web platform, so that not using it in 12 out of 13 cases of information collection from the web platform makes us encountered an error. Also, using a constant value for the sleep function significantly increases the time to retrieve information compared to using a variable value for it. In general, the findings show that the use of the proposed solution when using the Selenium library in order to extract information from the web platform helps to increase the accuracy of the information and also improves the time of complete information retrieval from the web platform.

Keywords: Web Crawler, Web Scraping, Selenium Library, Asynchronous, Data Accuracy, Data Retrieval Time

فصلنامه بازیابی دانش و نظام‌های معنایی
سال یازدهم، شمارهٔ ۴۱، زمستان ۱۴۰۳، ص ۱۹۹ تا ۲۲۵
jks.atu.ac.ir
DOI: 10.22054/jks.2024.80235.1660

سازوکاری برای مدیریت زمان و افزایش دقت اطلاعات هنگام استفاده از کتابخانه سلنیوم

فرناز تقی‌زاده کورایم — دانشجوی دکتری مدیریت فناوری اطلاعات، واحد تهران مرکزی، دانشگاه آزاد اسلامی، تهران، ایران

محمدرضا کاباران زاد قدیم* — دانشیار، گروه مدیریت صنعتی، واحد تهران مرکزی، دانشگاه آزاد اسلامی، تهران، ایران

سیدعبداله امین موسوی — استادیار، گروه مدیریت فناوری اطلاعات، واحد تهران مرکزی، دانشگاه آزاد اسلامی، تهران، ایران

مقاله پژوهشی
تاریخ دریافت: ۱۴۰۳/۰۳/۲۰
تاریخ پذیرش: ۱۴۰۳/۰۵/۱۹
ISSN: 2980-8243
eISSN: 2783-1795

چکیده

امروزه داده‌ها به‌عنوان یکی از دارایی‌های ارزشمند سازمان‌ها و صنایع مختلف، نقش مهمی را در توسعه و پیشرفت کسب‌وکارها ایفا می‌کنند. درواقع هر سازمانی برای جمع‌آوری داده‌های خود از منابع مختلفی استفاده می‌کند که یکی از این منابع بستر وب است که در آن روزانه داده‌های زیادی توسط کاربران مختلف و یا حتی ربات‌ها در سراسر جهان تولید و منتشر می‌شود. جمع‌آوری و تحلیل چنین داده‌هایی، می‌تواند اطلاعات مفیدی را برای سازمان فراهم نماید. به همین منظور طی دهه‌های گذشته ابزارهای مختلفی توسعه یافته‌اند که به برداشت اطلاعات از بستر وب کمک شایانی نموده‌اند که ازجمله آن‌ها می‌توان به کتابخانه‌های ریکوئست، سلنیوم، اسکرپی، سوپ زیبا و ... در زبان برنامه‌نویسی پایتون اشاره نمود. بااین‌حال، هر یک از این کتابخانه‌ها با چالش‌هایی مواجه هستند. ما در این مقاله با مطالعه کتابخانه سلنیوم و با توجه به وجود چالش‌های متعدد در آن، راه‌حلی را برای مدیریت زمان و بهبود چالش نامتقارن بودن آن ارائه نموده‌ایم. آزمایش‌های ما نشان می‌دهد که استفاده از راه‌حل پیشنهادی، دقت اطلاعات برداشت‌شده از بستر وب را افزایش و درنتیجه چالش نامتقارن بودن را بهبود می‌دهد و همچنین زمان برداشت اطلاعات از بستر وب را نیز کاهش می‌دهد.

کلیدواژه‌ها: خزنده وب، وب اسکرپینگ، کتابخانه سلنیوم، نامتقارن بودن داده، دقت اطلاعات، زمان برداشت اطلاعات

* نویسنده مسئول: kabaranzad@yahoo.com

مقاله حاضر برگرفته از رساله دکتری رشته مدیریت فناوری اطلاعات دانشگاه آزاد اسلامی، واحد تهران مرکزی است.

مقدمه

امروزه داده‌ها به‌عنوان یکی از دارایی‌های ارزشمند سازمان‌ها، نقش مهمی را در توسعه و پیشرفت کسب‌وکارها ایفا می‌کنند. این داده‌ها در بسترهای مختلفی موجود هستند که از مهم‌ترین آن‌ها می‌توان به اینترنت و شبکه‌های اجتماعی اشاره نمود که در هر دو محیط شاهد وجود حجم بالایی از داده‌ها هستیم که در صورت پردازش و استفاده درست از آن‌ها در زمان مناسب می‌تواند کمک شایانی به سازمان‌ها و کسب‌وکارها نماید.

همان‌طور که گفته شد، بستر اینترنت منبع اطلاعاتی بسیار قدرتمندی است که می‌توان اطلاعات موجود در آن را با کمک ابزارها و فنون مختلف برداشت و پس از تحلیل در راستای تصمیم‌گیری‌های بهتر و کاراتر استفاده نمود. یکی از روش‌های موجود برای برداشت اطلاعات از بستر وب، استفاده از زبان‌های برنامه‌نویسی نظیر پایتون، سی شارپ و جاوا است که در آن‌ها با استفاده از کتابخانه‌های مختلف نظیر ریکوئست[۱]، سلنیوم[۲]، اسکرپی[۳]، سوپ زیبا[۴] و ... می‌توان نسبت به خزش یا برداشت اطلاعات از بستر وب اقدام نمود.

طبق صحبت‌های پژوهشگران پیشین وقتی صحبت از برداشت خودکار اطلاعات از بستر وب به میان می‌آید، سلنیوم همیشه بهترین گزینه است، بااین‌حال این کتابخانه چالش‌های متعددی را به همراه دارد. یکی از چالش‌های استفاده از کتابخانه سلنیوم نامتقارن بودن آن است. نامتقارن به این معناست که اسکریپت‌ها یا همان کدهای مربوط به برداشت اطلاعات از بستر وب باید تعاملات ناهم‌زمان را مدیریت کنند که اغلب برطرف نمودن آن‌ها دشوار است. تعاملات ناهم‌زمان، هنگامی اتفاق می‌افتد که بخش‌های مختلف یک صفحه در وب، اطلاعات خود را از بخش‌های مختلف یک سرور، یا چند سرور مختلف و یا حتی واسط‌های برنامه‌نویسی[۵] مختلفی دریافت می‌کنند، درنتیجه ممکن است این اطلاعات در یک‌زمان و باهم نمایش داده نشوند و بارگذاری و نمایش برخی از اطلاعات در صفحه با کندی و تأخیر مواجه شود و یا در برخی موارد حتی بخش‌هایی از صفحه وب نمایش داده نشود و نیاز به زمان بیشتر یا بارگذاری مجدد صفحه باشد. به‌عبارت‌دیگر، محتوای ناهم‌زمان در صفحه وب

---

1. Requests
2. Selenium
3. Scrapy
4. Beautifulsoup
5. Application Programming Interface (API)

موجب می‌شود تا بخشی از داده‌ها را از سرور بدون بارگیری مجدد کل صفحه بازیابی کند؛ و اگر اسکریپت بخواهد با محتوای صحیح صفحه تعامل داشته باشد، باید منتظر بمانید تا محتوای ناهم‌زمان به‌روزرسانی شود (پاسخ از سرور دریافت شود و صفحه مطابق آن به‌روزرسانی شود). چالش اصلی در رسیدگی به تعاملات ناهم‌زمان، دانستن زمان انتظار برای بارگیری محتوای صفحه وب است که گاه توسط چندین به‌روزرسانی و یا افزایش زمان انتظار برای بارگذاری کامل اطلاعات، برطرف می‌گردد (Leotta et al., 2023).

اما سؤال اینجاست که بدون دخالت انسان، سیستم چه طور تشخیص دهد که چقدر باید منتظر برداشت کامل اطلاعات باشد و یا چند بار باید صفحه مجدد بارگذاری شود که اطلاعات درست را نمایش دهد؟ ما در این مقاله با مطالعه اطلاعات داروهای مکمل در سایت دیجی کالا و با استفاده از راه‌حل پیشنهادی در تلاش هستیم تا بهترین زمان بارگذاری را یافته و به‌عنوان ماژولی از آن در کتابخانه سلنیوم استفاده نماییم تا ضمن اعلام مشکل احتمالی وب، بتواند بدون دخالت انسان قادر به تشخیص بهترین زمان انتظار برای بارگذاری کامل صفحه و یا تشخیص نیاز به بارگذاری مجدد صفحه باشد.

در راستای هدف مقاله، نیازمند مرور ادبیات پژوهش هستیم که در ادامه به آن پرداخته‌ایم، سپس به تشریح راه‌حل پیشنهادی برای مدیریت زمان و رفع مشکل نامتقارن بودن سلنیوم پرداخته و پس‌ازآن راه‌حل پیشنهادی خود را مورد ارزیابی قرار داده‌ایم و در ادامه نیز بحث و نتیجه‌گیری و پیشنهادهای آتی ذکر شده‌اند.

پیشینهٔ پژوهش

امروزه، دیجیتالی کردن و مجازی‌سازی فزاینده فرایندهای اجتماعی منجر به زتابایت (میلیاردها گیگابایت) داده‌های موجود در وب شده است. این داده‌ها یک نمایش بلادرنگ از فرایندها، روابط و تعاملات متعدد در فضای مادی-اجتماعی ارائه می‌دهد؛ بنابراین، این حجم وسیع از داده‌های وب، فرصت‌های فراوانی را در اختیار پژوهشگران دانشگاهی قرار می‌دهد تا به سؤالات پژوهش‌های مربوط به پژوهش‌های جدید و قدیمی با دقت بیشتر و زمان‌بندی بهتر پاسخ دهند. همچنین شاغلین می‌توانند از این داده‌ها برای توسعه و درک بهتر مشتریان خود، تدوین راهبردها بر اساس این یافته‌ها و درنهایت بهبود عملکرد سازمانی استفاده کنند (Krotov et al., 2020).

همان‌طور که گفته شد، وب به‌عنوان بستری غنی از اطلاعات، دهه‌هاست که توسط پژوهشگران مختلف مورداستفاده قرار می‌گیرد؛ اما قطعاً بررسی چشمی تک‌تک اطلاعات و یا یادداشت‌برداری از حجم بالایی اطلاعات و یا به‌عبارت‌دیگر بازیابی اطلاعات به‌صورت دستی توسط انسان کار بسیار زمان‌بر، پرهزینه و دشواری است. به همین منظور طی دهه‌های گذشته ابزارهای مختلفی توسعه یافته‌اند که می‌توان با کمک آن‌ها اطلاعات موردنظر را به‌صورت خودکار و با صرف زمان بسیار کمتری بازیابی نمود که غالباً به این ابزارها و فنون برداشت اطلاعات از بستر وب، وب اسکرپینگ[1] گفته می‌شود (Henrys, 2021).

وب اسکرپینگ که به‌عنوان داده‌کاوی نیز شناخته می‌شود، فرایند جمع‌آوری مقادیر زیادی داده از وب و سپس قرار دادن آن در پایگاه‌های داده برای تجزیه‌وتحلیل آینده و استفاده‌های بعدی است. اسکرپینگ وب روشی است که استفاده از آن، بینشی در مورد داده‌های قیمت، پویایی بازار، روندهای غالب، شیوه‌های به کار گرفته‌شده توسط رقبا و چالش‌هایی که آن‌ها با آن روبرو هستند ارائه می‌دهد (Henrys, 2021).

طبق مطالعات انجام‌شده در سال ۲۰۲۰ که مبتنی بر جستجوی مقاله بر اساس نام یا همان عنوان مقالات بود، نشان می‌دهد که وب اسکرپینگ، یک پدیده نوظهور است؛ چراکه نخست، بیشتر مقالات حاوی این دو عبارت تنها به چند سال پیش بازمی‌گردد. دوم، تعداد کل مقالات مجلات و کنفرانس‌هایی که به‌صراحت به وب اسکرپینگ در چندین پایگاه داده معروف (برای نمونه EBSCO Business Source Complete، IEEE Xplore Digital Library، JSTOR و غیره) اختصاص داده شده است، کمتر از دویست است. ازجمله این مقالات می‌توان به مقاله پژوهشی کروتوف و تنیسون[2] (۲۰۱۸) اشاره نمود. این مقاله، به مطالعه داده‌های مالی و زبان‌های نشانه‌گذاری محبوب استفاده‌شده برای ذخیره و تبدیل این داده‌ها از یک متن ساده به یک متن قابل‌درک‌تر و زیباتر در بستر وب می‌پردازد و نشان می‌دهند که چگونه زبان R، همراه با کتابخانه‌هایش، می‌تواند برای جمع‌آوری، سازمان‌دهی و پیش‌پردازش داده‌های مالی مورداستفاده قرار گیرد. مقاله آموزشی مشابه دیگری توسط نیومن و همکاران[3] (۲۰۱۷) توضیح می‌دهد که چگونه می‌توان از فناوری IXPath برای برداشت فراداده از مخازن کتابخانه دیجیتال توسط پژوهشگران علاقه‌مند به

---

1. Web Scraping
2. Krotov & Tennyson
3. Neumann et al.

پژوهش‌های علمی استفاده نمود. همچنین مقاله بوییِنگ و وادل[1] (۲۰۱۷) نشان می‌دهد که چگونه داده‌های به‌دست‌آمده از Craigslist را می‌توان بازیابی و تجزیه‌وتحلیل کرد تا درک بهتری از بازار اجاره در ایالات‌متحده به دست آورد. درمجموع، این مقالات نشان می‌دهد که علاقه فزاینده‌ای به وب اسکرپینگ در بسیاری از رشته‌ها و صنایع علمی وجود دارد (Krotov et al., 2020).

کتابخانه سلنیوم که ابزاری برای برداشت اطلاعات از بستر وب است، در مطالعات مختلف و با اهداف مختلفی مورداستفاده قرار گرفته است، برای نمونه مقاله تئوتیا و همکاران[2] (۲۰۲۳) از آن به‌منظور مطالعه و تجزیه‌وتحلیل داده‌های وب‌سایت‌های فیلم چینی برای درک توزیع روند ژانرهای فیلم و رتبه‌بندی آن‌ها استفاده نموده است. مقاله یوان[3] (۲۰۲۳) روش جدیدی را برای برداشت اطلاعات از بستر وب پیشنهاد می‌کند که در آن سلنیوم (به‌عنوان یک ابزار مشهور برداشت اطلاعات از بستر وب) را با شبکه‌های عصبی کانولوشنال[4] ترکیب نموده تا به‌طور خودکار عناصر صفحات وب را شناسایی کند. در این روش، از سلنیوم برای پیمایش صفحات وب، دست‌کاری عناصر و همچنین گرفتن اسکرین‌شات استفاده می‌کند. این اسکرین‌شات‌ها متعاقباً با استفاده از شبکه‌های عصبی کانولوشنال پردازش می‌شوند تا از شناسایی و طبقه‌بندی عناصر صفحه وب اطمینان حاصل شود. مقاله نوشته‌شده توسط سوگانتی و وارون[5] (۲۰۲۴) با استفاده از قابلیت‌های سلنیوم، وب‌سایت‌ها را پیمایش نموده و اطلاعات مرتبط را به‌طور مؤثر استخراج می‌کند. همچنین با برطرف نمودن چالش‌های ناشی از محتوای پویا و ساختارهای پیچیده وب‌سایت، از استخراج یکپارچه داده‌ها اطمینان می‌دهد. این پژوهش به‌عنوان یک‌راه حل برای خودکار نمودن کارهای اسکراپی وب، توانمندسازی کاربران برای تعیین وب‌سایت‌های هدف، تعریف معیارهای استخراج و جمع‌آوری داده‌های موردنظر بدون مداخله دستی انسان عمل می‌کند و درواقع بهره‌وری و کارایی را در بازیابی داده‌ها از چشم‌انداز وسیع اینترنت به‌طور قابل‌توجهی افزایش می‌دهد (Suganthi & Varun, 2024).

---

1. Beoing & Waddell
2. Teotia et al.
3. Yuan, S.
4. Convolutional Neural Network (CNN)
5. Suganthi & Varun

همچنین مقاله نوشته‌شده توسط هنریس[1] (۲۰۲۱) به اهمیت وب اسکرپینگ در تجارت الکترونیک و بازاریابی پرداخته و ازجمله دلایل اهمیت آن را در ارائه اطلاعات مرتبط با موارد ذیل می‌داند:

نظارت بر قیمت و تحقیقات محصول؛ مقایسه قیمت آنلاین؛ تجزیه‌وتحلیل بهتر مشتری؛ تحلیل بازار؛ تبلیغات بهتر؛ تأثیرگذاری بر راهبرد بازاریابی و فروش؛ نظارت بر برند؛ استخراج جزئیات کسب‌وکار از فهرست کسب‌وکار؛ کمک به تحلیل آینده و غیره (Henrys, 2021).

سلنیوم یک چهارچوب آزمون خودکار متن‌باز و رایگان است که برای تأیید برنامه‌های وب در بسیاری از سکوها و مرورگرها استفاده می‌شود. برای ساخت اسکریپت‌های تست سلنیوم، می‌توان از زبان‌های برنامه‌نویسی متنوعی ازجمله جاوا، سی شارپ و پایتون استفاده کرد. سلنیوم در سال ۲۰۰۴ کشف شد. جیسون هاگینز[2] مردی بود که سلنیوم را تولید کرد. او یک توسعه‌دهنده و مهندس سلنیوم است. او زمانی شروع به کار کرد که به‌طور فعال سعی کرد خود را وقف کارهای شناختی کند و متوجه شد که به‌جای حرکت فیزیکی در این فعالیت‌ها، ممکن است از زمان خود حداکثر استفاده را بکند. او شروع به آزمایش با جاوا اسکریپت کرد و یک چهارچوب برنامه‌نویسی جاوا ایجاد کرد که می‌تواند با یک وب‌سایت ارتباط برقرار کند و به‌طور طبیعی تعدادی از برنامه‌ها را آزمایش کند که درنتیجه تلاش‌های موفقیت‌آمیز او کتابخانه سلنیوم ایجاد گردید (Teotia et al., 2023).

همان‌طور که گفته شد، وقتی صحبت از برداشت خودکار اطلاعات از بستر وب به میان می‌آید، سلنیوم همیشه بهترین گزینه است، چراکه: ۱- سلنیوم به‌طور رایگان در دسترس است؛ ۲- این نرم‌افزار از زبان‌های مختلفی ازجمله پایتون، پی‌اچ‌پی، جاوا، سی شارپ، روبی و جاوا اسکریپت و غیره پشتیبانی می‌کند؛ ۳- اسکریپت برای اجرا در هیچ سیستم‌عامل دیگری به‌جز سیستم‌عاملی که در آن ایجاد شده است نیازی به تغییر ندارد؛ زیرا با بسیاری از سیستم‌عامل‌های دیگر سازگار است؛ ۴- قابلیت سازگاری مرورگر آن به شخص اجازه می‌دهد تا اسکریپت را در هر مرورگری اجرا کند، ازجمله مایکروسافت اج، گوگل کروم، فایرفاکس، سافاری، اپرا و بسیاری دیگر؛ ۵- کاربرپسند و نصب آن ساده است؛ ۶- کد به سرویس‌های وب تفسیر می‌شود و درایور راه دور آن را با استفاده از پرس‌وجوهای پروتکل

---

[1]. Henrys, K.

[2]. Jason Huggins

انتقال ابرمتن[1] دریافت می‌کند، بنابراین قبل از اجرا به هیچ سروری نیاز ندارد. سلنیوم بیش از سایر فنون خراش دادن[2] وب یا همان وب اسکرپینگ موردعلاقه است؛ زیرا امکان تعامل با صفحات وب پویا را فراهم می‌کند. سلنیوم در شرایطی که داده‌هایی که می‌خواهیم استخراج کنیم در پشت اشیای جاوا اسکریپت پنهان می‌شوند که باید روی آن‌ها کلیک کرد تا محتوا نمایان شود، به‌خوبی کار می‌کند (Teotia et al., 2023).

اما باوجود مزایای گفته‌شده، سلنیوم به‌ویژه در طراحی برنامه‌های آزمون خودکار نرم‌افزار معایبی را نیز داراست که ازجمله آن‌ها می‌توان به ۱- کندی؛ ۲- شکنندگی[3] (این مشکل زمانی ایجاد می‌گردد که تکامل وب‌سایت یا برنامه کاربردی و یا رفع ایراد آن، موجب ایجاد خطا در بخش‌های دیگر شده و عملکرد آن‌ها را تحت تأثیر قرار دهد که در صورت وقوع چنین مشکلی سلنیوم اغلب قادر به تشخیص آن و یا حتی یافتن برخی پیوندها و فیلدها نیست)؛ ۳- پوسته‌پوسته شدن[4] (این مشکل هنگامی رخ می‌دهد که اسکریپت‌های وب‌سایت و یا برنامه کاربردی بدون دلیل آشکاری تغییر می‌کنند، این مشکل موجب ایجاد خطا هنگام اجرای کدهای مربوط به کتابخانه سلنیوم می‌شود که گاه خطایابی آن دشوار و چالش‌برانگیز است)؛ ۴- دشواری در قابلیت نگهداری؛ ۵- نامتقارن بودن؛ ۶- زمان‌بر بودن؛ ۷- مرورگر کراس[5] (این مشکل هنگام استفاده از مرورگرهای متفاوت اتفاق می‌افتد، به‌عبارت‌دیگر هر یک از مرورگرها ممکن است صفحات وب را به‌صورت متفاوتی ارائه دهند که این مورد به سیستم‌عامل مورداستفاده نیز بستگی دارد، یعنی کدهای نوشته‌شده برای با کمک کتابخانه سلنیوم ممکن است هنگام استفاده از یک مرور به‌درستی کار کند ولی هنگام استفاده از مرورگر دیگر یا حتی سیستم‌عامل دیگر با خطا مواجه گردد)؛ ۸- تجزیه‌وتحلیل شکست؛ ۹- زیرساخت؛ ۱۰- مقیاس‌پذیری؛ ۱۱- ادعاپذیری[6] (این مشکل به این دلیل ایجاد می‌گردد که اغلب داده‌ها از منابع نامعتبر جمع‌آوری می‌گردد و هرلحظه امکان دارد که ظاهر و احساس صفحات وب، سطح دسترسی و یا نقشه‌های تعاملی موجود در صفحات وب تغییر کند، درنتیجه برخی ادعاها مانند این مسئله که سلنیوم قادر به بررسی

---

1. HTTP
2. Scraping
3. Brittleness
4. Flakiness
5. Cross-browser
6. Assertability

ویژگی‌های بصری صفحات وب است، با چالش مواجه می‌گردد)؛ ۱۲- مستندسازی؛ ۱۳- پشتیبانی کردن، اشاره نمود (Leotta et al., 2023).

روش‌شناسی پژوهش

کتابخانه سلنیوم به‌عنوان یکی از بهترین کتابخانه‌ها برای برداشت اطلاعات از بستر وب، مدتی است که موردتوجه پژوهشگران مختلف قرار گرفته است. بااین‌حال، باوجود مزایای مختلف دارای معایبی نیز است که ازجمله آن‌ها می‌توان به عدم وجود مدیریت زمان و چالش نامتقارن بودن در آن اشاره نمود که در ادامه ابتدا تعریف عملیاتی از این دو مورد را ارائه نموده و سپس روش پژوهش را بیان نموده و بعدازآن راه‌حلی مناسبی به‌منظور بهبود کتابخانه سلنیوم که مبتنی بر این دو مورد است را ارائه نموده‌ایم.

تعریف عملیاتی از مدیریت زمان: ازجمله مشکلات کتابخانه سلنیوم کندی و زمان‌بر بودن آن است، چراکه به‌منظور برداشت اطلاعات از بستر وب باید از طریق مرورگر تمام لایه‌های موردنیاز را بررسی و اطلاعات موردنظر را جمع‌آوری نماید؛ اما چگونه می‌توان این کندی و زمان‌بر بودن را اندازه‌گیری نمود و آن را بهبود بخشید؟ ما به‌منظور افزایش سرعت برداشت اطلاعات از بستر وب و یا همان کاهش زمان برداشت اطلاعات از بستر وب و کاهش مشکلات کندی و زمان‌بر بودن کتابخانه سلنیوم که به‌صورت کلی آن را مدیریت زمان در کتابخانه سلنیوم نام‌گذاری نموده‌ایم، از تابع sleep استفاده نموده‌ایم. به‌عبارت‌دیگر، در این پژوهش مشخص نموده‌ایم که ایجاد چه تغییراتی با استفاده از تابع sleep می‌تواند مشکل کندی و زمان‌بر بودن کتابخانه سلنیوم را کاهش دهد.

تعریف عملیاتی از نامتقارن بودن: یکی دیگر از مشکلات کتابخانه سلنیوم نامتقارن بودن آن است. این مشکل هنگامی‌که تعاملات ناهم‌زمان در یک صفحه وب وجود دارد، اتفاق می‌افتد. تعاملات ناهم‌زمان، به این دلیل رخ می‌دهند که یک صفحه وب اطلاعات خود را از سرویس‌دهنده‌های مختلفی دریافت کرده و آن‌ها را نمایش می‌دهد، درنتیجه ممکن است سرعت ارتباط بین یک صفحه وب با تمام سرویس‌دهنده‌های آن یکسان نباشد و یا حتی به دلایل مختلف و در طول زمان پیوند بین سرویس‌دهنده و وب‌سایت تغییر نماید و این مسئله موجب می‌گردد تا برخی اطلاعات به نسبت اطلاعات دیگر با تأخیر نمایش داده شوند و یا اصلاً سلنیوم قادر به تشخیص وجود آن‌ها نباشد. این مسئله، اسکریپت‌های نوشته‌شده برای کتابخانه سلنیوم را با مشکل مواجه می‌نماید، چراکه اسکریپت‌ها قادر به برداشت بخشی از

اطلاعات هستند ولی نمی‌توانند تشخیص دهند که اطلاعات دیگر هنوز بارگذاری نشده‌اند و یا پیوند ارتباطی آن‌ها با سرور تغییر نموده است. درنتیجه خروجی نهایی را تحت تأثیر قرار می‌دهند و موجب می‌شوند که اطلاعات در جای درست خود قرار نگیرند، به‌عنوان نمونه در مورد مطالعاتی ما قیمت یک محصول در مقابل نام محصولی دیگر ثبت گردد. ما به‌منظور برطرف نمودن این مشکل که نام آن را در عنوان مقاله افزایش دقت اطلاعات گذاشته‌ایم، از فهرست‌ها و طول آن‌ها استفاده نموده‌ایم.

ما در پژوهش خود ابتدا داده‌های کیفی را از طریق اسکریپت‌های نوشته‌شده با استفاده از کتابخانه سلنیوم از بستر وب جمع‌آوری نمودیم و سپس با محاسبه تعداد این داده‌ها و مدت‌زمان موردنیاز برای برداشت این اطلاعات از بستر وب، داده‌های کمی را محاسبه نمودیم. به‌عبارت‌دیگر داده‌های نخست ما کیفی و داده‌های دوم که اسکریپت‌های سلنیوم براساس آن‌ها بهبود یافت و پژوهش نیز براساس آن‌ها ارزیابی گردید، کمی است. به‌عبارت‌دیگر، داده‌های استفاده‌شده در این پژوهش هم شامل داده‌های کمی و هم شامل داده‌های کیفی هستند. درنتیجه داده‌های موجود در این پژوهش ترکیبی و پژوهش حاضر ازنظر هدف کاربردی و ازنظر روش تحلیلی است. روش جمع‌آوری اطلاعات هم به‌صورت مطالعه موردی بود که در آن از مکمل‌های دارویی سایت دیجی کالا استفاده شده است.

اولین موردی که آن را در کتابخانه سلنیوم موردبررسی قرار دادیم، عدم وجود امکان مدیریت زمان در آن است. مدیریت زمان درواقع اشاره به این موضوع دارد که مدت‌زمان انتظار برای برداشت اطلاعات از بستر وب در این کتابخانه مشخص نیست و بستگی بسیار زیادی به‌سرعت اینترنت دارد. همچنین این مشکل زمانی حاد می‌شود که سیستمی تماماً خودکار برای برداشت اطلاعات از بستر وب طراحی نماییم.

اما چگونه می‌توان بر این مشکل غلبه و آن را برطرف نمود؟

قبل از پرداختن به این مشکل، ابتدا نگاهی به ساختار برخی از صفحات وب می‌اندازیم. اکثر صفحات اصلی فروشگاهی در بستر وب، خلاصه‌ای از اطلاعات محصول از قبیل نام محصول، قیمت و میزان تخفیف را در صفحه اول نمایش می‌دهند که کاربر با کلیک بر روی محصول می‌تواند وارد صفحه مربوط به آن محصول شده و اطلاعات کامل‌تری از قبیل مشخصات، روش استفاده، دیدگاه کاربران و کارشناسان پیرامون محصول و ... را به دست آورد.

همچنین، طبق بررسی‌های انجام‌شده بر روی سایت‌های فروشگاهی مختلف در بستر وب، صفحه اصلی فروشگاه‌ها بر اساس تعداد محصول موجود در آن‌ها، معمولاً سه نوع هستند؛ یکی صفحاتی که فقط نیاز به پیمایش کردن دارند تا بتوان اطلاعات اولیه (نظیر نام محصول و قیمت و میزان تخفیف آن) همه محصولات را مشاهده نمود، دوم صفحاتی که هم نیاز به پیمایش دارند و هم نیاز به رفتن به صفحات بعد دارند تا بتوان اطلاعات اولیه تمام محصولات موردنظر را مشاهده نمود و سوم صفحاتی که نه نیاز به پیمایش دارند و نه نیاز به رفتن به صفحه بعد که معمولاً در صفحات از نوع سوم، اطلاعات و محصولات کمی وجود دارد.

کتابخانه سلنیوم به‌منظور برداشت اطلاعات از بستر وب از وب درایور استفاده می‌کند و برای هر مرورگر نیز وب درایور مناسب آن طراحی گردیده است. همچنین ازجمله دستورات مهم در کتابخانه سلنیوم، دستور get است. این دستور به‌منظور بارگذاری صفحه موردنظر با کمک وب درایور به کار گرفته می‌شود. برای نمونه به دستور ذیل که دستوری برای باز شدن یکی از صفحات دیجی کالا است، توجه نمایید:

```
driver = webdriver.chrome()
driver.get("""https://www.digikala.com/search/nutritional-supplement""")
```

اما اگر فقط همین دستور را اجرا نماییم و سپس دستورات مربوط به برداشت اطلاعاتی از قبیل نام کالا، قیمت کالا و ... از صفحه موردنظر را قرار دهیم، به‌احتمال‌زیاد با خطا مواجه خواهیم شد، زیرا احتمالاً یا صفحه اصلاً بارگذاری نشده است و یا بخشی از آن بارگذاری نشده است. برای رفع این مشکل تابع sleep در کتابخانه سلنیوم طراحی گردیده است. تابع sleep که به‌صورت ذیل نوشته می‌شود، موجب ایجاد زمان انتظاری که توسط کاربر تعریف شده، می‌شود و احتمال خطا را کاهش می‌دهد. برای نمونه، در کد ذیل، برنامه پس از رفتن به صفحه موردنظر، ۲ ثانیه صبر می‌کند تا صفحه بارگذاری شود:

```
Time.sleep(2)
```

اما چه میزان زمان برای بارگذاری این صفحه که برای نمونه ما در تابع بالا آن را ۲ ثانیه در نظر گرفته‌ایم، موردنیاز است؟ برای رفع این مشکل ما از راه‌حلی استفاده نموده‌ایم که شامل سه مرحله مختلف است و فرضیه‌های آن به شرح ذیل است:

۱. ما در پژوهش‌های خود ساعت ثابتی را برای برداشت اطلاعات در نظر گرفته‌ایم چراکه هدف ما ساخت سیستمی خودکار است که مثلاً روز پانزدهم هرماه ساعت ۸ صبح

اطلاعات موردنظر را برداشت و داده‌های موجود را به‌روزرسانی نماید. به همین منظور ساعت به‌روزرسانی اطلاعات جدول ساعت ۸ صبح است.

۲. فرض بر این است که زمان انتظار برای باز شدن هر صفحه جدید، زمان انتظار برای نمایان شدن محصول بعد از هر پیمایش و برداشت هر یک از اطلاعات مربوط به هر محصولات در شرایط با سرعت اینترنت مناسب، نیم ثانیه است.

۳. همچنین میزان پیمایش را نیز می‌دانیم و یا اصلاً نیاز به پیمایش نداریم.

۴. در صفحاتی که بررسی می‌نماییم حداقل یک کالا وجود دارد.

۵. بیشترین تعداد کالای موجود در صفحه ۲۰ است.

مرحله ۱) طبق بررسی‌هایی که به‌صورت دستی انجام شد، ابتدا متغیر t1 را به شرح ذیل تعریف می‌کنیم:

```
t1 = 0.5
```

که در آن t1 زمان موردنیاز برای باز شدن صفحه اصلی در حالت عادی است.

مرحله ۲) از حلقه while و try، except خارج از آن، استفاده می‌نماییم. به این منظور که اگر بعد از ۳ ثانیه صفحه باز نشدد، یا به‌عبارت‌دیگر بعد از ۶ بار تلاش با زمان‌های sleep متفاوت صفحه و کالاهای موردنظر باز نشده و نمایان نشدند، خطای در دسترس نبودن سایت و یا کندی اینترنت چاپ می‌گردد؛ به‌عبارت‌دیگر، در بدنه اصلی try، except دیگری قرار می‌دهیم، این بخش به این منظور قرار دارد که اگر صفحه در زمان t1 باز نشد و محصولات نمایان نشدند، نیم ثانیه به آن اضافه می‌شود و مجدد موردبررسی قرار گیرد که این کار تا ۵ بار ادامه پیدا می‌کند و سپس t1 به‌دست‌آمده به‌عنوان یک متغیر نگهداری می‌شود و برای صفحات بعد مورداستفاده قرار می‌گیرد.

```
t1 = 0.5
try:
    while t1 <= 3:
        try:
            driver.get("""سایت موردمطالعه""") # کد باز شدن صفحه
            time.sleep(t1) # انتظار برای باز شدن صفحه
            # کد مربوط به اسکرول صفحه
            time.sleep(t1) # انتظار برای نمایش اطلاعات بعد اسکرول
```

```
مابقی بخش‌های کد مربوط به برداشت اطلاعات هر محصول است #
        except:
            t1 += 0.5
except:
    print('The internet speed is very slow or the intended site is down')
```

که در آن برنامه با زمان t1=0.5 ابتدا تلاش می‌کند تا اطلاعات صفحه را به‌صورت کامل نمایش دهد و اگر موفق نشد، نیم ثانیه به t1 اضافه می‌نماید و این تکرار تا ۶ بار ادامه می‌یابد. درصورتی‌که صفحه به‌صورت کامل نمایش داده شود، از مقدار جدید t1 برای صفحات بعد استفاده می‌نماییم. همچنین از این راه‌حل می‌توان برای صفحات مربوط به محصول نیز استفاده نمود، به‌طوری‌که بعد از کلیک بر روی صفحه محصول اگر صفحه قابل‌نمایش نباشد و یا خطا مواجه شویم، می‌توانیم صفحه را با تابع refresh() به‌جای تابع get() مجدد به‌روزرسانی نموده و زمان تابع sleep() را هم بر اساس راه‌حل بالا افزایش دهیم تا صفحه محصول نیز به‌صورت کامل بارگذاری شود.

مرحله ۳) در صورت باز شدن صفحه وارد مرحله سوم می‌شویم که این مرحله مربوط به برداشت اطلاعات اولیه محصولات است؛ اما چه طور مانع بروز چالش نامتقارن بودن شویم؟ به‌عبارت‌دیگر چگونه اطمینان حاصل می‌شود که اطلاعات مربوط به تمام محصولات به‌صورت کامل بارگذاری شده و اطلاعات درست را برداشت خواهیم کرد.

در این مرحله با توجه به وجود پنج نوع اطلاعات مختلف (نظیر نام کالا، قیمت، میزان تخفیف، قیمت بعد از تخفیف، نوع تخفیف) از هر محصول، ابتدا پنج فهرست مختلف و خالی که نشان‌دهنده هرکدام از این پنج نوع است را به شکل ذیل تعریف می‌نماییم:

Product_name =[] که فهرست مربوط به نام محصول است.

Product_ price =[] که فهرست مربوط به قیمت محصول است.

Product_off =[] که فهرست مربوط به میزان تخفیف محصول است.

Product_price2 =[] که فهرست مربوط به قیمت محصول بعد از تخفیف است.

Product_off_type =[] که فهرست مربوط به نوع تخفیف است.

سپس با کمک دستورهای مربوط به برداشت اطلاعات، اطلاعات مربوط به هر محصول را برداشته و در فهرست مربوط به آن قرار می‌دهیم. به‌عبارت‌دیگر سلنیوم پس از خواندن اطلاعات موردنظر مانند نام کالا، آن را در متغیری ذخیره نموده و سپس آن را به فهرست

می‌افزاید. پس از چند با برداشت اطلاعات از بستر وب ممکن است اطلاعات وب‌سایت تغییر نماید. به‌عبارت‌دیگر، کدی که ما در روز اول نوشتیم و به‌خوبی هم کار می‌کرد، ممکن است طی هفته‌های آینده به‌خوبی کار نکند و اسکریپت نوشته‌شده را با خطا مواجه نماید. همان‌طور که گفته شد ما در پژوهش‌های خود از برداشت نام محصول، قیمت، میزان تخفیف و نوع تخفیف از بستر وب استفاده نموده‌ایم و متوجه شدیم که بعد چند هفته تعداد آیتم موجود در فهرست‌ها باهم برابر نیست. به‌عبارت‌دیگر تعداد نام داروی برداشت‌شده از سایت دیجی کالا با تعداد نوع تخفیف برداشت‌شده برای محصول یکسان نیستند. به‌عنوان نمونه هنگامی‌که پس از اجرای کد و برداشت اطلاعات از بستر وب، فهرست مربوط به نام محصول شامل ۲۶۷۰ تعداد محصول مختلف، فهرست مربوط به قیمت شامل ۲۴۸۰ تعداد قیمت و فهرست مربوط به نوع تخفیف شامل ۱۷۸۰ تعداد نوع تخفیف باشد، ما با خطا مواجه هستیم، چراکه تعداد اطلاعات موجود در فهرست‌ها باید باهم یکسان باشند. شاید سؤال پیش بیاید که تمام محصولات دارای تخفیف نیستند و درنتیجه تعداد اقلام موجود در فهرست تخفیف‌ها با تعداد اقلام موجود در نام کالا یکسان نیست ولی این مورد را ما از ابتدای نوشتن اسکریپت در نظر گرفته‌ایم. ولی باوجود در نظر گرفتن تمام این موارد بازهم ممکن است مواردی پیش بیاید که نمی‌توان آن را پیش‌بینی کرد. برای نمونه، در طول سال اغلب فروشگاه‌ها تخفیف‌های مختلفی را ارائه می‌دهند، مثل تخفیف‌های ویژه و یا شگفت‌انگیز ولی برخی از روزها نوع این تخفیف‌ها تغییر می‌کند، مثلاً بلک فرایدی (جمعه سیاه) که یک روز خاص در سال است، اغلب فروشگاه‌ها تخفیف‌های بیشتری را ارائه می‌دهند که اگر ما آن را در اسکریپت خود از قبل در نظر نگرفته باشیم، ممکن است کد با خطا مواجه گردد و یا تعداد مقادیر موجود در فهرست‌ها باهم برابر نباشند و این عامل مانع از ایجاد خروجی اکسل یا sql و درنتیجه مانع از به‌روزرسانی پایگاه داده و برنامه اصلی می‌گردد. ما در این مقاله راه‌حلی را ارائه داده‌ایم که با کمک آن می‌توان از وقوع چنین مشکلات پیش‌بینی‌نشده‌ای جلوگیری نمود.

اسکریپت استفاده‌شده برای جلوگیری از رد شدن آیتم و عدم ثبت آن در فهرست:

```
check1 = [[product_prices], [product_off], [product _prices2], [Product_off_type] ]

for j in range(0,4):
        if len(product_name) < check1[len[j]]:
```

```
                product_name.append('error')
        elif len(product_name) > check1[len[j]]:
                check[j].append('error')
```

طبق بررسی‌های انجام‌شده، ممکن است پس از برداشت اطلاعات از بستر وب متوجه شویم که تعداد آیتم‌های موجود در فهرست‌های تعریف‌شده، مثلاً فهرست نام کالا، فهرست قیمت و ... باهم یکسان نیستند. درنتیجه کد بالا که در انتهای حلقه for مربوط به محصولات قرار دارد و بعد از افزودن مقدار به فهرست، بررسی می‌کند که آیا تعداد آیتم‌های موجود در فهرست‌ها باهم برابر است یا خیر، مانع از رخ دادن چنین مشکلی می‌گردد که این کد با استفاده از ایجاد فهرستی به نام چک به این کار کمک می‌کند. ما در اسکریپت اصلی ابتدا کدهای مربوط به برداشت اطلاعات مربوط به یک محصول خاص را قرار داده‌ایم و بعد از کدهای اصلی، کد بالا را قرار دادیم. به‌عبارت‌دیگر، برنامه بعد از برداشت اطلاعات یک کالا و افزودن آن‌ها به فهرست‌های اصلی از قبل تعریف‌شده، تعداد آیتم فهرست‌های موجود را با کمک تابع len محاسبه و آن‌ها را در فهرست چک قرار می‌دهد. سپس با کمک حلقه for طول هر فهرست را با بقیه فهرست‌ها مقایسه و در صورت کم بودن تعداد آیتم‌های موجود در یک فهرست کلمه "error" را به آن می‌افزاید.

به‌منظور ارزیابی راه‌حل‌های ارائه‌شده در این پژوهش، ما اطلاعات مربوط داروهای مکمل سایت دیجی کالا که تقریباً ۳۰۰۰ داروی مختلف بودند را ۱۳ بار در ۱۳ تاریخ مختلف، از اول مهرماه ۱۴۰۲ تا ۲۵ اسفندماه ۱۴۰۲ موردبررسی قرار داده‌ایم. در این بررسی، کدهای نوشته‌شده به‌منظور برداشت اطلاعات از سایت گفته‌شده، بار اول با راه‌حل پیشنهادشده و بار دوم بدون راه‌حل پیشنهادشده، اجرا شدند و در هر اجرا هم‌زمان برداشت اطلاعات و هم تعداد اطلاعات مربوط به هر فهرست یا همان ستون ثبت و با یکدیگر مقایسه شدند و میزان خطا و درصد آن نیز براساس تفاضل زمان برداشت اطلاعات در اجرای اول و دوم و تفاضل تعداد اطلاعات برداشت‌شده برای هر ستون در اجرای اول و دوم محاسبه گردید.

درنهایت نیز پس از اعمال دو اسکریپت بالا و بارگذاری اطلاعات در فایل موردنظر، احتمالاً ما با تعدادی سلول یا فیلد که حاوی کلمه error هستند مواجه خواهیم بود که باید آن‌ها را به‌صورت دستی بررسی و در صورت نیاز یا کد موردنظر را اصلاح نماییم و یا

اطلاعات را به‌صورت دستی اصلاح نماییم که این بستگی به هزینه، دشواری و زمان موردنظر برای اصلاح این موارد دارد.

یافته‌ها

پس از پیاده‌سازی و استفاده از راه‌حل ارائه‌شده، جدول ۱ حاصل گردید؛ که نشان می‌دهد دقت و صحت اطلاعات برداشت‌شده، نسبت به عدم استفاده از راه‌حل پیشنهادی، افزایش یافته و زمان برداشت اطلاعات نیز بهبود یافته است:

جدول ۱. تعداد اطلاعات برداشت‌شده از دیجی کالا و زمان موردنیاز برای برداشت آن‌ها

ردیف	تاریخ برداشت اطلاعات	هنگام عدم استفاده از راه‌حل پیشنهادی						زمانی که از راه‌حل پیشنهادشده، استفاده می‌شود					
		نام محصول	قیمت	تخفیف	قیمت بعد از تخفیف	نوع تخفیف	زمان برداشت اطلاعات به دقیقه	نام محصول	قیمت	تخفیف	قیمت بعد از تخفیف	نوع تخفیف	زمان برداشت اطلاعات به دقیقه
۱	۰۱-۰۷-۱۴۰۲	۲۶۷۰	۲۶۶۲	۲۶۷۰	۲۶۷۰	۲۶۶۳	۲۶۷	۲۶۷۰	۲۶۷۰	۲۶۷۰	۲۶۷۰	۲۶۷۰	۱۰۷
۲	۱۵-۰۷-۱۴۰۲	۲۵۴۲	۲۵۳۷	۲۵۴۲	۲۵۴۲	۲۴۹۳	۲۵۴	۲۵۴۲	۲۵۴۲	۲۵۴۲	۲۵۴۲	۲۵۴۲	۱۰۲
۳	۰۱-۰۸-۱۴۰۲	۲۸۹۸	۲۸۹۸	۲۸۹۸	۲۸۹۸	۲۸۹۸	۲۹۰	۲۸۹۸	۲۸۹۸	۲۸۹۸	۲۸۹۸	۲۸۹۸	۱۱۶
۴	۱۵-۰۸-۱۴۰۲	۳۰۴۲	۳۰۴۰	۳۰۴۲	۳۰۴۲	۲۹۴۳	۳۰۴	۳۰۴۲	۳۰۴۲	۳۰۴۲	۳۰۴۲	۳۰۴۲	۱۲۲
۵	۰۱-۰۹-۱۴۰۲	۲۸۶۰	۲۸۵۵	۲۸۶۰	۲۸۶۰	۲۵۲۴	۲۸۶	۲۸۶۰	۲۸۶۰	۲۸۶۰	۲۸۶۰	۲۸۶۰	۱۹۱
۶	۱۵-۰۹-۱۴۰۲	۳۱۲۳	۳۱۲۱	۳۱۲۳	۳۱۲۳	۲۸۲۳	۳۱۲	۳۱۲۳	۳۱۲۳	۳۱۲۳	۳۱۲۳	۳۱۲۳	۱۵۶
۷	۰۱-۱۰-۱۴۰۲	۳۰۶۹	۳۰۷۰	۳۰۵۶	۳۰۵۶	۲۹۴۳	۳۰۷	۳۰۷۰	۳۰۷۰	۳۰۷۰	۳۰۷۰	۳۰۷۰	۱۵۴
۸	۱۵-۱۰-۱۴۰۲	۲۹۲۶	۲۹۲۴	۲۹۲۶	۲۹۲۶	۲۸۵۶	۲۹۳	۲۹۲۶	۲۹۲۶	۲۹۲۶	۲۹۲۶	۲۹۲۶	۱۶۶

۹	۰۱-۱۱-۱۴۰۲	۲۵۶۷	۲۵۶۷	۲۵۶۷	۲۵۰٤	۲۵۷	۲۵۶۷	۲۵۶۷	۲۵۶۷	۲۵۶۷	۲۵۶۷	۱۰۳
۱۰	۱۵-۱۱-۱۴۰۲	۲۵۶۱	۲۵۵۸	۲۵۶۱	۲۵۶۱	۲۵۶	۲۵۶۱	۲۵۶۱	۲۵۶۱	۲۵۶۱	۲۵۶۱	۱٤۵
۱۱	۰۱-۱۲-۱۴۰۲	۲۷۲۶	۲۷۲۶	۲۷۱۸	۲۶٤۲	۲۷۳	۲۷۲۶	۲۷۲۶	۲۷۲۶	۲۷۲۶	۲۷۲۶	۱۳۶
۱۲	۱۵-۱۲-۱۴۰۲	۳۱۱۰	۳۱۱۰	۳۱۱۰	۲۸۶۲	۳۱۱	۳۱۰٤	۳۱۱۰	۳۱۱۰	۳۱۱۰	۳۱۱۰	۲۰۷
۱۳	۲۵-۱۲-۱۴۰۲	۳۱۲٤	۳۰۸۹	۳۱۲٤	۲۸۶۰	۳۱۲	۳۱۲٤	۳۱۲٤	۳۱۲٤	۳۱۲٤	۳۱۲٤	۲۶۰
مجموع		۳۷۲۱۸	۳۷۱۵۷	۳۷۱۹۷	۳۵۵۷۲	۳۷۲۲	۳۷۱۹۹	۳۷۲۱۹	۳۷۲۱۹	۳۷۲۱۹	۳۷۲۱۹	۱۹۶٤

جدول ۱ نشان می‌دهد که برنامه نوشته‌شده به‌منظور برداشت اطلاعات از بستر وب، هنگام استفاده از راه‌حل پیشنهادی و هنگام عدم استفاده از آن به تفکیک تاریخ، برای هر یک از فهرست‌ها یا همان ستون‌های نام محصول، قیمت، تخفیف، قیمت بعد از تخفیف و نوع تخفیف، چند برداشت موفقیت‌آمیز داشته است. به‌عبارت‌دیگر، برنامه نوشته‌شده در تاریخ ۰۱-۰۷-۱۴۰۲ هنگام عدم استفاده از راه‌حل پیشنهادی، موفق شده تا ۲۶۷۰ نام محصول مختلف را برای ستون نام محصول و ۲۶۶۲ قیمت مختلف را برای ستون یا همان فهرست قیمت از سایت دیجی کالا بردارد. این در حالی است که هنگام استفاده از راه‌حل پیشنهادی مقادیر مربوط به ستون‌ها یا همان فهرست‌های نام محصول، قیمت و ... عددی برابر داشته و اطلاعات نام محصول، قیمت و ... ۲۶۷۰ داروی مکمل مختلف از بستر وب به‌صورت موفقیت‌آمیز برداشت شده است. همچنین دو ستون «زمان برداشت اطلاعات به دقیقه» مدت‌زمانی که طول می‌کشد تا برنامه نوشته‌شده اطلاعات مربوط به تمام داروهای مکمل را از سایت دیجی کالا بردارد، نشان می‌دهد. این ستون‌ها نیز به تفکیک هنگام استفاده و عدم استفاده از راه‌حل پیشنهادی و تاریخ در جدول ۱ نشان داده شده‌اند. طبق بررسی‌های انجام‌شده به‌صورت دستی، عدد عنوان‌شده در ستون نام محصول هنگام استفاده از راه‌حل پیشنهادی، نشان‌دهنده تعداد واقعی و اصلی داروهای مکمل موجود در تاریخ‌های ذکرشده در سایت دیجی کالا بودند.

بر اساس جدول ۱ عدم استفاده از راه‌حل پیشنهادشده، موجب می‌گردد تا تعداد اطلاعات موجود در هر فهرست باهم یکسان نباشد و درنهایت در خروجی پایگاه داده نه‌تنها اطلاعات کامل نباشد، بلکه اطلاعات در جای درست خود نیز قرار نگیرند، برای نمونه تخفیف ارائه‌شده برای یک محصول در مقابل نام محصول دیگر قرار گیرد. همچنین زمان مربوط به تابع sleep() نیز برای هر پیمایش و هر صفحه به‌صورت جداگانه عدد ثابت ۳ ثانیه در نظر گرفته شده است، درحالی‌که در راه‌حل پیشنهادشده این مقدار بر اساس ترافیک موجود در شبکه و سرعت اینترنت از نیم تا ۳ ثانیه متغیر است. اطلاعات به‌دست‌آمده پس از برداشت اطلاعات از بستر وب نشان می‌دهد که راه‌حل پیشنهادشده، هم دقت اطلاعات را افزایش داده و هم‌زمان برداشت اطلاعات از بستر وب را کاهش می‌دهد. همچنین در جدول ۲ تفاضل اطلاعات موجود در جدول شماره یک را نشان می‌دهد. برای نمونه همان‌طور که در جدول ۲ قابل‌مشاهده است، هنگام عدم استفاده از راه‌حل پیشنهادی در طول تقریباً ۶ ماه، درمجموع ۱۶۴۷ سطر اطلاعات از ستون نوع تخفیف خالی بوده و در صفحه وب بر اساس جدول یک ۳۵۵۷۲ سطر اطلاعات برای ستون نوع تخفیف یافت شده است، درحالی‌که در اصل ۳۷۲۱۹ سطر از اطلاعات یا همان داروی مکمل وجود داشته است. بااین‌حال ستون نام محصول ثبات بیشتری داشته و در طول ۶ ماه، فقط یک مورد از آن توسط خزنده وب یا همان کتابخانه سلنیوم، در هنگام عدم استفاده از راه‌حل پیشنهادی، یافت نشده است. همچنین زمان برداشت اطلاعات از بستر وب در هنگام استفاده از راه‌حل پیشنهادی ۱۷۵۸ دقیقه نسبت به عدم استفاده از آن، کمتر بوده است.

جدول ۲. قدر مطلق تفاضل هنگام استفاده از راه‌حل پیشنهادی و هنگام عدم استفاده از آن

ردیف	تاریخ برداشت اطلاعات	قدر مطلق تفاضل هنگام استفاده از راه‌حل پیشنهادی و هنگام عدم استفاده از آن					
		نام محصول	قیمت	تخفیف	قیمت بعد از تخفیف	نوع تخفیف	زمان برداشت اطلاعات به دقیقه
۱	۱۴۰۲-۰۷-۰۱	۰	۸	۰	۰	۷	۱۶۰
۲	۱۴۰۲-۰۷-۱۵	۰	۵	۰	۰	۴۹	۱۵۳
۳	۱۴۰۲-۰۸-۰۱	۰	۰	۰	۰	۰	۱۷۴

٤	١٤٠٢-٠٨-١٥	٠	٢	٠	٠	٩٩	١٨٣
٥	١٤٠٢-٠٩-٠١	٠	٥	٠	٠	٣٣٦	٩٥
٦	١٤٠٢-٠٩-١٥	٠	٢	٠	٠	٣٠٠	١٥٦
٧	١٤٠٢-١٠-٠١	١	٠	١٤	١٤	١٢٧	١٥٤
٨	١٤٠٢-١٠-١٥	٠	٢	٠	٠	٧٠	١٢٧
٩	١٤٠٢-١١-٠١	٠	٠	٠	٠	٦٣	١٥٤
١٠	١٤٠٢-١١-١٥	٠	٣	٠	٠	٠	١١١
١١	١٤٠٢-١٢-٠١	٠	٠	٨	٠	٨٤	١٣٦
١٢	١٤٠٢-١٢-١٥	٠	٠	٠	٦	٢٤٨	١٠٤
١٣	١٤٠٢-١٢-٢٥	٠	٣٥	٠	٠-	٢٦٤	٥٢
مجموع		١	٦٢	٢٢	٢٠	١٦٤٧	١٧٥٨

بحث و نتیجه‌گیری

در این مقاله ما چالش‌های کند بودن، زمان‌بر بودن و نامتقارن بودن کتابخانه سلنیوم را موردمطالعه و ارزیابی قراردادیم. مطالعات ما با استفاده از زبان برنامه‌نویسی پایتون و برداشت اطلاعات از سایت دیجی کالا در ۱۳ تاریخ مختلف از ۰۱-۰۷-۱۴۰۲ تا ۲۵-۱۲-۱۴۰۲ انجام شد. مطالعات نشان می‌دهد که استفاده از راه‌حل بررسی فهرست و یکسان بودن طول فهرست در پایان برداشت هر محصول از بستر وب بسیار مهم است، به‌طوری‌که عدم استفاده از آن در ۱۲ مورد از ۱۳ مورد برداشت اطلاعات از بستر وب، ما را با خطا مواجه نمود. همچنین استفاده از یک مقدار ثابت برای تابع sleep() زمان برداشت اطلاعات را نسبت به زمان استفاده از مقدار متغیر برای آن به‌طور چشم‌گیری افزایش می‌دهد. در کل یافته‌ها نشان می‌دهد که استفاده از راه‌حل پیشنهادی در هنگام استفاده از کتابخانه سلنیوم به‌منظور برداشت اطلاعات از بستر وب، کمک شایانی به افزایش دقت و صحت اطلاعات نموده و زمان برداشت کامل اطلاعات از بستر وب را نیز بهبود می‌دهد.

طبق بررسی‌های انجام‌شده و همان‌طور که در جدول ۲ نیز آمده است، عدم استفاده از کد مربوط به مدیریت زمان موجب می‌گردد تا در طول تقریباً ۶ ماه ۱۷۵۶ دقیقه زمان یا تقریباً ۲۹ ساعت زمان بیشتری صرف برداشت اطلاعات از بستر وب گردد. به‌عبارت‌دیگر همان‌طور که در جدول ۱ آمده است، در صورت استفاده از راه‌حل پیشنهادی برای مدیریت زمان، درمجموع ۱۹۶۴ دقیقه زمان برای ۱۳ برداشت مختلف اطلاعات از سایت دیجی کالا

در طول تقریباً ۶ ماه موردنیاز است، این درحالی است که عدم استفاده از راه‌حل پیشنهادی و استفاده از یک مقدار ثابت برای تابع ()sleep این زمان را به ۳۷۲۲ دقیقه افزایش می‌دهد. همچنین عدم استفاده از راه‌حل پیشنهادی برای مشکل نامتقارن بودن، موجب می‌گردد تا درمجموع و در طول تقریباً ۶ ماه و ۱۳ برداشت مختلف، یک آیتم از اطلاعات مربوط به ستون محصول، ۶۲ آیتم از اطلاعات مربوط به ستون قیمت، ۲۲ آیتم از اطلاعات مربوط به ستون تخفیف، ۲۰ آیتم مربوط به ستون قیمت بعد از تخفیف و ۱۶۴۷ آیتم مربوط به ستون نوع تخفیف کمتر از میزانی که باید از بستر وب برداشت گردد، برداشت شود. درحالی‌که هنگام استفاده از راه‌حل پیشنهادی، مقادیر تمام ستون‌ها باهم یکسان بودند، این بدین معناست که اطلاعات در جای صحیح خود قرار گرفته‌اند و درصورتی‌که اطلاعات بخشی از سایت هنوز نمایش داده نشده و یا ارتباط بخشی از سایت تغییر یافته باشد، به‌جای داده مربوط به آن آیتم، مقدار error درج گردیده است. درج کلمه error نه‌تنها موجب می‌گردد که اطلاعات در جای درست خود قرار گیرند، بلکه با توجه به بررسی‌های انجام‌شده اغلب کلمه error ثبت‌شده به دلیل تغییر ارتباط داخل وب‌سایت بوده است، درنتیجه می‌توان اسکریپت نوشته‌شده را با استفاده از آن‌ها اصلاح نمود و یا توسعه داد.

پیشنهادها

در این مقاله ما راه‌حلی را به‌منظور بهبود عملکرد کتابخانه سلنیوم ارائه نمودیم که به‌موجب آن می‌توان زمان برداشت اطلاعات از بستر وب و چالش نامتقارن بودن را بهبود داد. بااین‌حال، ما در مطالعات خود میزان و زمان پیمایش را نادیده گرفته‌ایم که خود یکی چالش‌های مهم در کند بودن کتابخانه سلنیوم است. بخصوص در صفحات فروشگاهی که بعد از پیمایش، صفحه مقداری به سمت بالا و پایین هدایت‌شده و برداشت اطلاعات و کلیک روی محصولات را با خطا مواجه می‌نماید.

همچنین ما در پژوهش خود فقط پیرامون چالش‌های کندی سلنیوم، زمان‌بر بودن آن و نامتقارن بودن سلنیوم مطالعه نموده و پیشنهادهایی را به‌منظور بهبود آن‌ها ارائه نمودیم. بااین‌حال، کتابخانه سلنیوم دارای چالش‌های دیگری نیز است که نیازمند مطالعه و بررسی بیشتر است. این چالش‌ها عبارت‌اند از ۱- شکنندگی؛ ۲- پوسته‌پوسته شدن؛ ۳- دشواری در

قابلیت نگهداری[۱]؛ ۴- مرورگر کراس[۲]؛ ۵- تجزیه‌وتحلیل شکست[۳]؛ ۶- زیرساخت[۴]؛ ۷- مقیاس‌پذیری[۵]؛ ۸- ادعاپذیری[۶]؛ ۹- مستندسازی[۷]؛ ۱۰- پشتیبانی کردن[۸] که در مقاله لواتا و همکاران (۲۰۲۳) به‌تفصیل پیرامون آن‌ها صحبت شده است و می‌توان به‌منظور بهبود کتابخانه سلنیوم آن‌ها را مطالعه و راه‌های مناسبی را جهت برطرف نمودن آن‌ها پیشنهاد نمود. شایان‌ذکر است که بررسی اکثر این چالش‌ها نیازمند مورد مطالعاتی متفاوت با رویکردی متفاوت است که خارج از محدوده این مقاله است.

ORCID

Farnaz Taghizadeh Kourayem http://orcid.org/0000-0003-2676-2495
Mohammadreza Kabaranzad Ghadim http://orcid.org/0000-0002-4372-5226
Seyed Abdollah Amin Mousavi http://orcid.org/0009-0005-3052-5910

References

Boeing, G., & Waddell, P. (2017). New insights into rental housing markets across the United States: Web scraping and analyzing craigslist rental listings. *Journal of Planning Education and Research*, *37*(4), 457-476.

Henrys, K. (2021). Importance of web scraping in e-commerce and e-marketing. Available at SSRN 3769593.

Krotov, V., & Tennyson, M. (2018). Scraping Financial Data from the Web Using the R Language. *Journal of Emerging Technologies in Accounting*, *15*(1), 169-181.

Krotov, V., Johnson, L., & Silva, L. (2020). Tutorial: Legality and ethics of web scraping.

Leotta, M., García, B., Ricca, F., & Whitehead, J. (2023, April). Challenges of end-to-end testing with selenium WebDriver and how to face them: A survey. In *2023 IEEE Conference on Software Testing, Verification and Validation* (ICST) (pp. 339-350). IEEE.

Neumann, M., Steinberg, J., & Schaer, P. (2017). Web-Scraping for non-programmers: Introducing OXPath for digital library metadata harvesting. *Code4Lib Journal*, *38*.

---

[1]. Maintainability
[2].Cross-browser
[3]. Failure analysis
[4]. Infrastructure
[5]. Scalability
[6]. Assertability
[7]. Documentation
[8]. Support

Suganthi, V., & Varun, M. M. (2024). Automation Using Selenium. *International Journal Of Multidisciplinary Research In Science, Engineering And Technology*, e-ISSN:2582-7219.

Teotia, H., Shishodia, G., Tyagi, E., Prakash, A., & Avasthi, S. (2023, April). Instagram Analysis and Activity Automation: Using Python and Selenium Automation Tools. In *2023 International Conference on Computational Intelligence, Communication Technology and Networking* (CICTN) (pp. 522-526). IEEE.

Yuan, S. (2023). Design and Visualization of Python Web Scraping Based on Third-Party Libraries and Selenium Tools. *Academic Journal of Computing & Information Science*, *6*(9), 25-31.

استناد به این مقاله: تقی‌زاده کورایم، فرناز، کاباران‌زاد قدیم، محمدرضا و موسوی، سید عبداله امین. (۱۴۰۳). سازوکاری برای مدیریت زمان و افزایش دقت اطلاعات هنگام استفاده از کتابخانه سلنیوم. *فصلنامه بازیابی دانش و نظام‌های معنایی*، *۱۱* (۴۱)، ۱۹۹-۲۲۵. DOI: 10.22054/jks.2024.80235.1660