18. Adeolu O. Essential Guide to Python for All Levels Collection: Forging Ahead in Tech and Programming. 2024. 314 p.
19. Nocedal J., Wright S. Numerical Optimization. Springer Series in Operations Research and Financial Engineering. 2006. 664 p. DOI: https://doi.org/10.1007/978-0-387-40065-5.

Стаття надійшла 15.10.2024
Стаття прийнята 28.10.2024

УДК 004.62: 658.5 **doi: 10.31498/2225-6733.49.1.2024.321179**

© Сергієнко А.В.[1], Балалаєва О.Ю.[2], Гаркуша І.М.[3], Платонов Д.М.[4]

РОЗРОБКА ТА ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПАРСИНГУ САЙТІВ ІЗ ВИКОРИСТАННЯМ ФРЕЙМВОРКУ SELENIDE

Стаття присвячена дослідженню методів автоматизації збору даних із вебсайтів за допомогою технологій парсингу. У роботі описано основні переваги парсингу порівняно з ручним збором даних, наведено класифікацію існуючих парсерів, їх можливості, обмеження та застосування в реальних проєктах. Проведено детальний аналіз популярних комерційних та безкоштовних парсерів, таких як Import.io, Webhose.io, Dexi.io, Scraperhub, ParseHub, Visual Scraper, Spinn3r, 80legs, Scraper, OutWit Hub, з метою визначення їх переваг та недоліків у різних сценаріях використання. Особливу увагу приділено порівнянню фреймворків Selenium та Selenide, що широко застосовуються для автоматизації взаємодії з веббраузерами. Зроблено висновок про доцільність використання фреймворку Selenide завдяки його спрощеному синтаксису, можливостям роботи з динамічним контентом та підтримці інтелектуального очікування. У статті представлено розробку власного парсера на базі Selenide, орієнтованого на потреби малих і середніх підприємств із обмеженим бюджетом. Система побудована на сучасному технологічному стеку, що включає Java 11, Python, PostgreSQL, Angular 12, Docker, Gradle, Kafka, Node.js. Детально описано архітектуру програми, взаємодію між модулями, а також реляційну модель бази даних для зберігання отриманих даних. Запропонований підхід дозволяє налаштовувати парсер для роботи з різними типами сайтів, забезпечує високу швидкість збору та обробки інформації, а також гнучкість у налаштуванні параметрів вибірки. Створений інструмент надає можливість використовувати технології контейнеризації для спрощення розгортання та підтримки додатка. Результати роботи можуть бути використані для реалізації ефективних інформаційно-пошукових систем та автоматизації рутинних процесів збору даних, що особливо актуально для компаній, які прагнуть оптимізувати свої бізнес-процеси та зменшити витрати.

Ключові слова: *парсинг, парсер, сайт, вебдодаток, інформаційно-пошукова система, дані, Selenium, Selenide.*

[1] *канд. техн. наук, доцент, ДВНЗ «Приазовський державний технічний університет», м. Дніпро/Маріуполь, ORCID: 0000-0003-1328-2572, sergienko_a_v@pstu.edu*
[2] *канд. техн. наук, доцент, ДВНЗ «Приазовський державний технічний університет», м. Дніпро/Маріуполь, ORCID: 0000-0003-1461-4399, balalaeva_e_u@pstu.edu*
[3] *канд. техн. наук, доцент, НТУ «Дніпровська політехніка», м. Дніпро, ORCID: 0000-0003-1190-1501, garkusha.i.m@nmu.one*
[4] *магістр, ДВНЗ «Приазовський державний технічний університет», м. Дніпро/Маріуполь, platonov_d_m@pstu.edu*

p-ISSN: 2225-6733; e-ISSN: 2519-271X

A. Serhiienko, O. Balalaieva, I. Garkusha, D. Platonov. Development and research of the efficiency of a website parsing information system using the Selenide framework. *The article is devoted to the study of methods for automating data collection from websites using parsing technologies. The paper describes the main advantages of parsing compared to manual data collection, provides a classification of existing parsers, their capabilities, limitations and application in real projects. A detailed analysis of popular commercial and free parsers, such as Import.io, Webhose.io, Dexi.io, Scraperhub, ParseHub, Visual Scraper, Spinn3r, 80legs, Scraper, OutWit Hub, is carried out in order to determine their advantages and disadvantages in various usage scenarios. Particular attention is paid to the comparison of the Selenide and Selenide frameworks, which are widely used to automate interaction with web browsers. The conclusion is made about the feasibility of using the Selenide framework due to its simplified syntax, capabilities for working with dynamic content and support for intelligent waiting. The article presents the development of a custom parser based on Selenide, focused on the needs of small and medium-sized enterprises with a limited budget. The system is built on a modern technology stack, including Java 11, Python, PostgreSQL, Angular 12, Docker, Gradle, Kafka, Node.js. The program architecture, interaction between modules, and the relational database model for storing the obtained data are described in detail. The proposed approach allows you to configure the parser to work with different types of sites, provides high speed of information collection and processing, as well as flexibility in configuring sampling parameters. The created tool provides the opportunity to use containerization technologies to simplify the deployment and support of the application. The results of the work can be used to implement effective information search systems and automate routine data collection processes, which is especially important for companies that seek to optimize their business processes and reduce costs.*

Keywords: *web parsing, web parser, web site, web application, information search system, data, Selenium, Selenide.*

Постановка проблеми. У сучасному світі швидкий доступ до структурованої інформації є ключовою вимогою для ефективного функціонування бізнесу. Вебсайти виступають основним джерелом актуальних даних, які можуть використовуватися для маркетингових досліджень, аналізу конкурентів або оптимізації внутрішніх процесів, проте ручний збір даних вже давно є неефективним через значні витрати часу та високу ймовірність помилок. Це обумовлює зростання попиту на автоматизовані інформаційно-пошукові системи парсингу, які дозволяють отримувати великий обсяг необхідних даних у зручному форматі з мінімальним втручанням людини. Доцільність використання парсерів обумовлена швидкістю їх роботи, можливістю складання вибірки за багатьма параметрами та налаштування парсингу з певною періодичністю, а також не лише відсутністю помилок у звітах, а й рекомендаціями щодо виправлення помилок на сайті. На сьогоднішній день існує багато сервісів для парсингу, які можуть бути як хмарними, так і коробковими версіями. Водночас слід зазначити, що існуючі інструменти мають низку обмежень, пов'язаних із складністю налаштування, недостатньою гнучкістю або високою вартістю, що створює труднощі для малих і середніх підприємств. Тому актуальною задачею є розробка власного парсера, для чого необхідно проаналізувати сферу застосування парсингу, обрати необхідний функціонал, а також провести дослідження ефективності роботи створеного парсера.

Аналіз останніх досліджень і публікацій. Import.io пропонує розробникові легко формувати власні пакети даних, для чого необхідно тільки імпортувати інформацію з певної вебсторінки й експортувати її в CSV. Це дозволяє збирати інформацію з тисячі вебсторінок дуже швидко, при цьому не написавши ні рядка коду, а також створювати тисячі API згідно з заданими вимогами. Тариф Starter призначений для простих випадків використання, таких як вилучення даних зі статичних сайтів, та включає тільки базові функції. Даний тариф містить 5 000 запитів на місяць та вартує від 199 доларів на місяць. Тариф Standard дозволяє працювати з більш складнішими завданнями, такими як робота з динамічними сайтами, капчами або аутентифікацією, містить 20 000 запитів та коштує від 599 доларів на місяць. Разом з веб інструментом доступні безкоштовні застосування для Windows, Mac OS X і Linux для створення екстракторів і пошукових роботів. Import.io підходить для швидкого збору даних, але може бути недостатньо потужним

для складних або масштабних проєктів. Даний інструмент може мати труднощі з обробкою сайтів із динамічним контентом, а зміна структури сайту може зламати налаштування імпорту.

Webhose.io [1] забезпечує прямий доступ в реальному часі до структурованих даних, отриманих у результаті парсингу тисяч онлайн джерел. Цей парсер здатний збирати вебдані на більш ніж 240 мовах і зберігати результати в різних форматах, у тому числі в XML, JSON і RSS. Webhose.io представляє собою вебзастосування для браузеру, що використовує власну технологію парсингу даних, яка дозволяє обробляти величезні об'єми інформації з численних джерел з єдиним API. Webhose пропонує безкоштовний тарифний план за обробку 1 000 запитів в місяць і 50 доларів за преміальний план, що покриває 5 000 запитів в місяць. Webhose.io робить фокус на текстовому контенті, тому не дуже підходить для вилучення мультимедіа (зображень чи відео).

Dexi.io [2] (попередня назва CloudScraper) здатний парсити інформацію з будь якого вебсайту і не вимагає завантаження додаткових застосувань, як і Webhose, а редактор самостійно встановлює своїх пошукових роботів і отримує дані в режимі реального часу. Користувач може зберегти зібрані дані в хмарі, наприклад, Google Drive і Box.net, або експортувати дані у форматах CSV або JSON. CloudScraper забезпечує анонімний доступ до даних, пропонуючи ряд проксі-серверів, які допомагають приховати ідентифікаційні дані користувача. CloudScraper зберігає дані на своїх серверах впродовж 2 тижнів, потім архівуючи їх. Сервіс пропонує 20 годин роботи безкоштовно, після чого він коштуватиме 29 доларів в місяць. Dexi.io підходить для автоматизації веб-скрейпінгу без глибоких технічних знань, але для великих або специфічних проєктів можуть знадобитися додаткові налаштування чи ресурси.

Scrapinghub (зараз Zyte) [3] представляє собою хмарний інструмент парсингу даних, який допомагає вибирати і збирати необхідні дані для будь-яких цілей. Scrapinghub використовує Crawlera, розумний проксі-ротатор, оснащений механізмами, здатними обходити захисту від ботів. Сервіс здатний справлятися з великим обсягом інформації і захищеними від роботів сайтами. Scrapinghub перетворює вебсторінки в організований контент. Базовий безкоштовний пакет дає доступ до одного пошукового робота (обробка до 1 Гб даних, далі – 9 доларів в місяць), преміальний пакет надає чотирьох паралельних пошукових ботів. Scrapinghub варто використовувати для професійного вебскрейпінгу, але даний сервіс може бути складним і затратним для простих задач чи новачків.

ParseHub [2] може парсити один або багато сайтів з підтримкою JavaScript, AJAX, сеансів, cookie і редиректів. Сервіс використовує технологію самонавчання і здатний розпізнати складні документи в мережі, після чого генерує вихідний файл у тому форматі, який потрібний користувачеві. ParseHub існує окремо від вебзастосування в якості програми робочого столу для Windows, Mac OS X і Linux. Програма дає безкоштовно п'ять пробних пошукових проектів. Тарифний план Преміум за 89 доларів дозволяє 20 проєктів і обробку 10 000 вебсторінок за проєкт. ParseHub добре підходить для початківців і невеликих проєктів, але може виявитися недостатньо потужним для масштабного або складного скрейпінгу.

VisualScraper [4] є програмним засобом для парсингу великих об'ємів інформації з мережі. Даний сервіс видобуває дані з декількох вебсторінок і синтезує результати в режимі реального часу, при цьому дані можна експортувати в формати CSV, XML, JSON і SQL. Користуватися і управляти веб-даними допомагає простий інтерфейс типу point and click. VisualScraper пропонує пакет з обробкою більше 100 000 сторінок з мінімальною вартістю 49 доларів в місяць. Є безкоштовне застосування, схоже на Parsehub, доступне для Windows із можливістю використання додаткових платних функцій.

Spinn3r [1] дозволяє парсити дані з блогів, новинних стрічок, новинних каналів RSS і Atom, соціальних мереж. Spinn3r має «оновлюваний» API, який робить 95% роботи по індексації. Це припускає вдосконалений захист від спаму і підвищений рівень безпеки даних. Spinn3r індексує контент, як Google, і зберігає витягнуті дані у файлах формату JSON. Інструмент постійно сканує мережу і знаходить оновлення потрібної інформації з безлічі джерел, користувач завжди має оновлювану в реальному часі інформацію. Консоль адміністрування дозволяє управляти процесом дослідження; є повнотекстовий пошук. Spinn3r робить фокус на медіаконтенті, він оптимізований для збору даних із новинних сайтів, блогів, соціальних медіа та RSS-стрічок, при цьому менш ефективний для збору інших типів даних.

p-ISSN: 2225-6733; e-ISSN: 2519-271X

80legs [5] представляє собою гнучкий вебінструмент парсингу сайтів, який можна достатньо точно підлаштувати під потреби користувача, при цьому сервіс є дуже потужним, він справляється з дуже великими обсягами даних і має функцію негайного витягання. Клієнтами 80legs є такі гіганти як MailChimp і PayPal. Опція «Datafiniti» дозволяє знаходити дані якнайшвидше, що забезпечує високоефективну пошукову мережу. Сервіс пропонує безкоштовний пакет – 10 000 посилань за сесію, який можна оновити до пакету INTRO за 29 доларів в місяць, що збільшує кількість посилань до 100 000 за сесію. 80legs підходить для масштабних, ресурсомістких завдань, але вимагає технічної підготовки та не є оптимальним для невеликих або простих проєктів.

Scraper [1] є розширенням для Chrome з обмеженими функціями парсинга даних, але воно корисне для онлайнових досліджень і експортування даних в Google Spreadsheets. Цей інструмент призначений як для новачків, такі для експертів, які можуть скопіювати дані у буфер обміну або сховище у вигляді електронних таблиць, використовуючи OAuth. Scraper – це безкоштовний інструмент, що працює прямо у браузері й автоматично генерує XPaths для визначення URL, які треба перевірити. Сервіс досить простий: він не має повної автоматизації або пошукових ботів, як Import або Webhose, але це можна розглядати як перевагу для новачків, оскільки його не доведеться довго настроювати, щоб отримати потрібний результат. Scraper доцільно використовувати для простого та швидкого вилучення даних, але він не підходить для складних або масштабних проєктів.

OutWit Hub [6] – це доповнення Firefox з десятками функцій видобування даних, що може автоматично переглядати сторінки і зберігати інформацію у необхідному користувачеві форматі. Це один з найпростіших безкоштовних вебінструментів по збору даних, що не вимагають спеціальних знань в написанні кодів. OutWit Hub підходить для поміркованих завдань зі скрейпінгу, але може бути обмежений при роботі з великими або складними сайтами.

Аналіз існуючих готових сервісів показав, що доцільною є реалізація за допомогою фреймморку власного парсера, який включав би найчастіше затребуваний функціонал. Програмувати парсери самостійно можна за допомогою спеціальних бібліотек, що існують для мов програмування Python, JavaScript, Java. Бібліотеки на Python надають безліч ефективних і швидких функцій для парсингу [7]. Багато з цих інструментів можна підключити до готового застосування у форматі API для створення краулеров, що настроюються, а відкритий код мають наступні проєкти: BeautifulSoup, Selenium, Lxml. Для JavaScript теж існують готові бібліотеки для парсингу із зручними функціональними API: Cheerio, Osmosis, Apify SDK. У Java реалізовані різні інструменти і бібліотеки, а також зовнішні API, які можна використати для парсингу: Jsoup, Jaunt, HTMLUnit.

Огляд вищеперелічених бібліотек та фреймворків показав, що найбільш перспективною є розробка парсера за допомогою фреймворків Selenium та Selenide на базі мови програмування Java, при цьому останнім часом в проєктах частіше використовують інструмент Selenide замість Selenium Web Driver. Для остаточного вибору засобу розробки було проведено порівняльний аналіз роботи з фреймворками Selenium та Selenide [8].

Selenium є портативним фреймворком для автоматизації вебдрайверів, що надає інструменти для тестування вебзастосунків. Крім того, Selenium Testing Framework поставляється з предметно орієнтованою мовою (Selenese), що дозволяє писати тести на декількох мовах програмування: Groovy, C#, Java, PHP, Perl, Python, Scala і Ruby. Перевага таких тестів полягає в тому, що їх можна запускати в сучасних веббраузерах, і це дозволяє проводити тестування на різних операційних системах, таких як Windows, macOS і Linux. Серед можливостей даного фрейворку слід зазначити крос-браузерну автоматизацію та автоматизацію мобільних браузерів і нативних застосунків. Він пропонує низькорівневі API для детального управління браузером та підходить для створення інструментів автоматизації, таких як FluentLenium, Thucydides, Watir-webdriver. Selenium – велику спільноту та безліч ресурсів для навчання, є основою для багатьох інших інструментів автоматизації, але його використання вимагає детального знання роботи WebDriver та тайм-менеджменту. Selenium підходить для загальної автоматизації браузерів і тестування специфічних сценаріїв, але він не забезпечує вбудованих механізмів для роботи з динамічним контентом. Selenium надає широкі можливості для створення тестових рішень, але вимагає більше зусиль для управління тестами.

p-ISSN: 2225-6733; e-ISSN: 2519-271X

Selenide побудований на основі Selenium та спеціалізується на спрощенні автоматизації UI-тестів. Даний фреймворк інтегрується з Selenium WebDriver, має простий синтаксис для взаємодії з елементами сторінки та орієнтований на написання лаконічного коду. Він може вирішувати проблеми динамічного контенту та таймаутів за допомогою інтелектуального очікування, а також забезпечує високоабстраговані API. Selenide обирають, коли потрібні швидкі, стабільні та легко підтримувані UI-тести, адже даний фреймворк спрощує процес тестування завдяки вбудованим рішенням для поширених проблем (таймаути, Ajax).

За результатами проведеного аналізу ефективності парсингу сайтів було зроблено вибір на користь Selenid, оскільки спеціально розроблений для тестування вебзастосувань і широко відомий як інструмент автоматизації веббраузеру, а також забезпечує підтримку крос-браузерної автоматизації, автоматизації нативних застосувань, автоматизації мобільних браузерів тощо.

Метою роботи є дослідження автоматизації отримання даних із сайтів, які є різними за характером, за допомогою парсера, що створений за базі фреймворку Selenid та працює за модифікованою технологією.

Виклад основного матеріалу. У даній роботі було розроблено систему, що отримує дані з сайту за допомогою парсингу. Для реалізації проєкту використовується наступний стек технологій: СУБД PostgreSQL, фреймворк Angular 12, а також Docker, Docker-compose, Gradle, Selenid, Node js, Npm, Kafka; мови програмування Java 11 та Python. Angular — один з найпопулярніших і сучасних фреймворків (бібліотек) JavaScript, що розвивається Google. За допомогою нього можна швидко створювати складні й динамічні інтерфейси на сайтах. Docker – програмне забезпечення для автоматизації розгортання і управління застосуваннями в середовищах з підтримкою контейнеризації, контейнеризатор застосувань.

Схему взаємодії програмних модулів додатку наведено на рисунку 1.

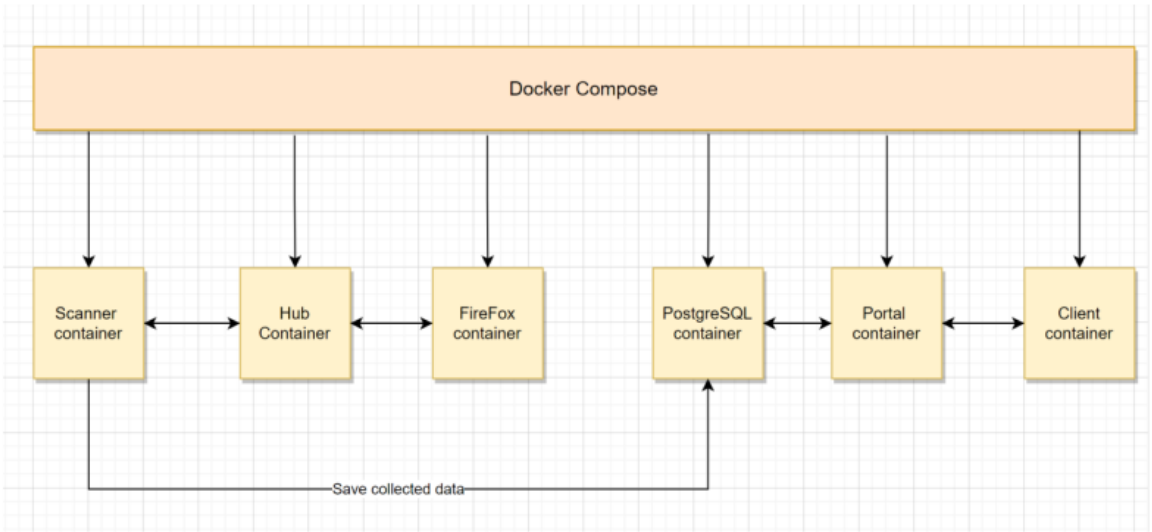

Рис. 1 – Схема взаємодії програмних модулів

Реляційна модель даних наведена на рисунку 2.

Процес виконання додатку передбачає наступні кроки:

− запускається консоль, здійснюється перехід в директорію проекту, виконується команда запуску контейнерів

− сканер-контейнер підключається до hub-контейнеру, а hub-контейнер – до firefox;

− сканер зберігає отримані дані в базі даних;

− клієнт-контейнер робить запит до порталу-контейнеру, який отримує збережену в базі даних інформацію та виводить її в клієнтську частину додатку.

p-ISSN: 2225-6733; e-ISSN: 2519-271X

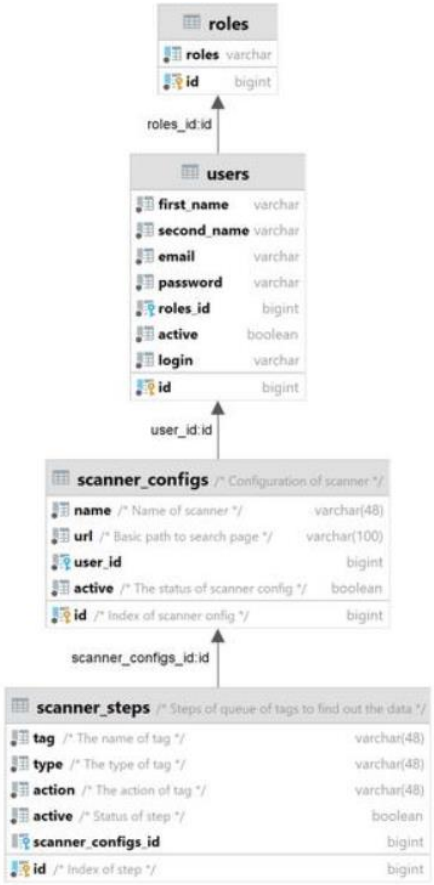

Рис. 2 – Реляційна схема бази даних інформаційно-пошукової системи парсингу сайтів

Програма сканера працює в контейнері докера, незалежно від програми порталу, а знайдена інформація зберігається в БД PostgreSQL (рисунок 3).

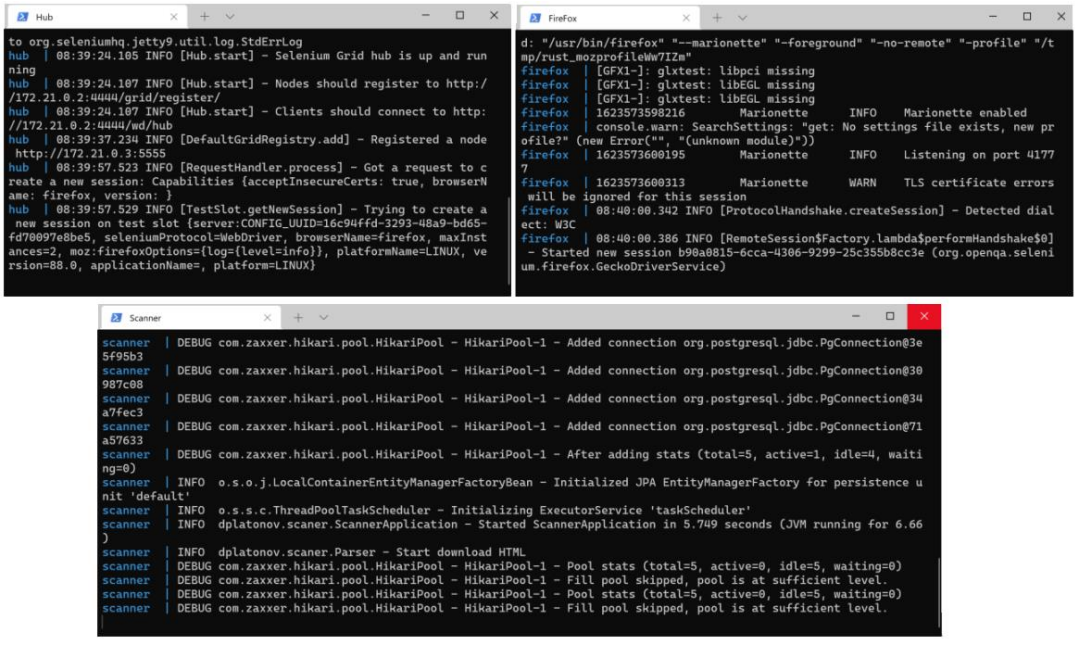

Рис. 3 – Консоль запуску та роботи частин сканеру: Hub, FireFox, Scanner

p-ISSN: 2225-6733; e-ISSN: 2519-271X

Додаток Portal працює у контейнері докера, незалежно від програми сканера. Головна сторінка містить таблицю зі сканерами, кожному з яких користувачем задаються умови пошуку та параметри роботи сканеру (рисунок 4).

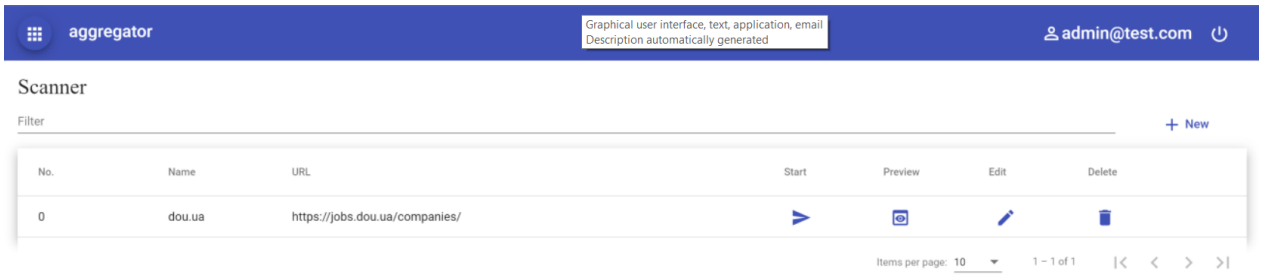

Рис. 4 – Головна сторінка системи

Для створення нового сканеру авторизований користувач задає ім’я сканеру та базову URL-адресу для пошуку і ланцюгу подій для доступу до потрібних даних. Для створення ланцюга подій (рисунок 5) необхідно зазначити назву тегу, його клас, а також дію, яку необхідно буде виконати (наприклад, натискання кнопки).

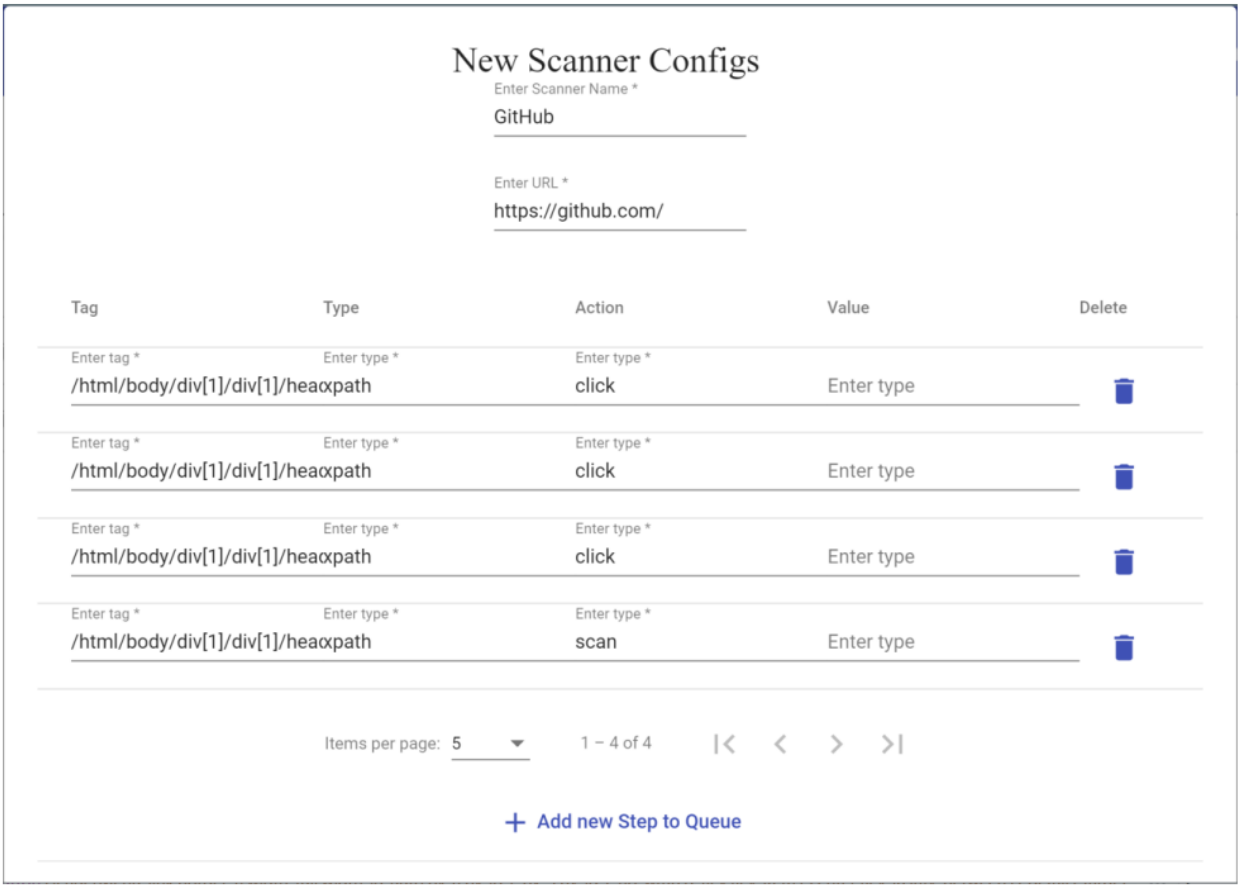

Рис. 5 – Створення ланцюгу подій

З наведеного на рисунку 4 прикладу налаштувань видно, що сканер повинний буде перейти за URL: https://jobs.dou.ua/companies/ та знайти на HTML сторінці ланцюг тегів, що задані в чіткій послідовності із типом class: l-items; l-company; cn a; city; descr;. Останній тег має містити текст, який шукатиме парсер.

p-ISSN: 2225-6733; e-ISSN: 2519-271X

Для подальшого аналізу ефективності роботи сканера із обраними налаштуваннями пошуку та налаштуваннями самого парсера зручною опцією є експорт отриманих результатів у файл у форматах csv та xls (рисунок 6).

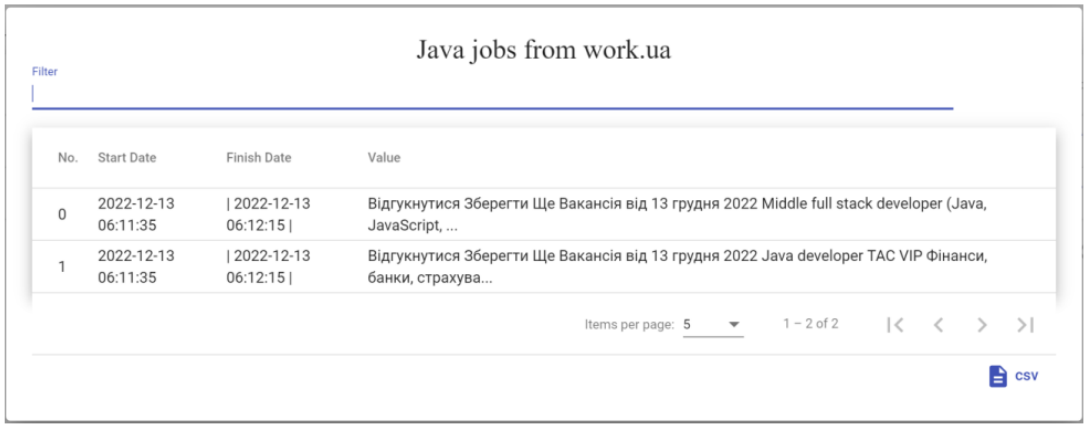

Рис. 6 – Експорт отриманих сканером даних у окремий файл

У рамках даної роботи було проведено дослідження парсингу даних з різних сайтів з використанням розробленого парсера Aggregator.

Для прикладу було поставлено задачу отримати дані по вакансіях з сайтів www.work.ua, www.jobs.dou.ua. Під час дослідження складності видобутку даних розробленим авторами парсером, аналізували кількість кроків, яка необхідна для виконання поставленого завдання для кожного з обраних сайтів. Для цього було створено два сканери з різними налаштуваннями за тегами та за переліком подій (рисунки 7-8).

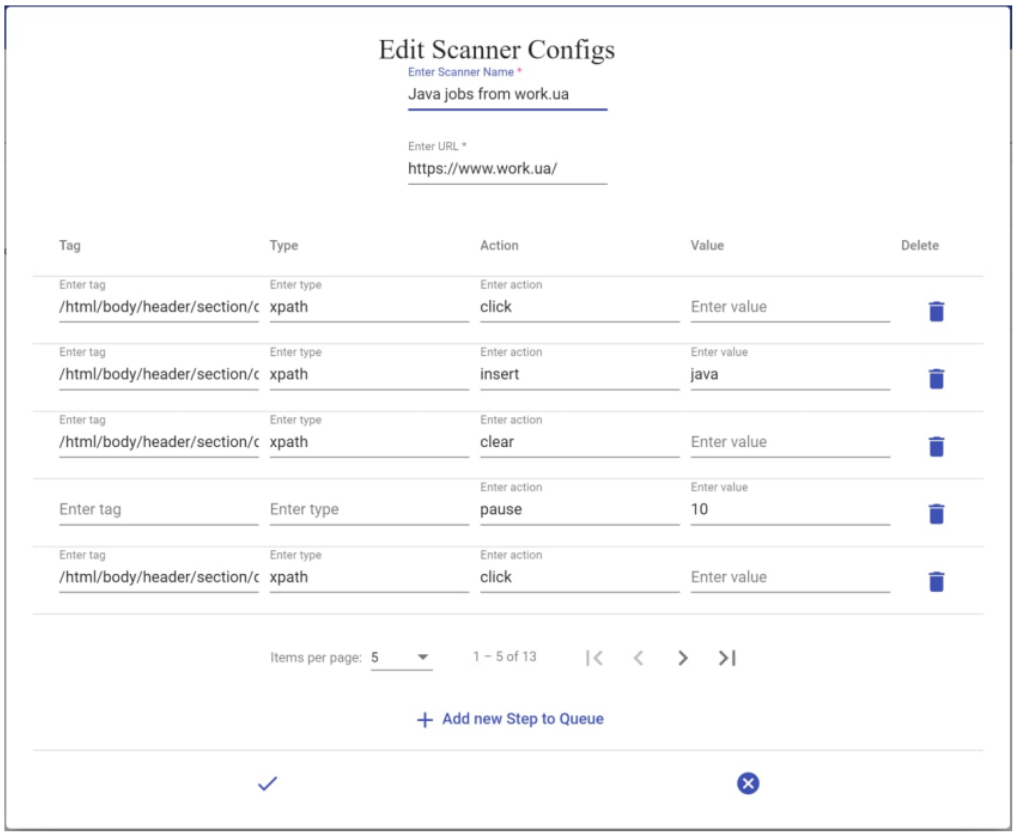

Рис. 7 – Налаштування сканера по www.work.ua

p-ISSN: 2225-6733; e-ISSN: 2519-271X

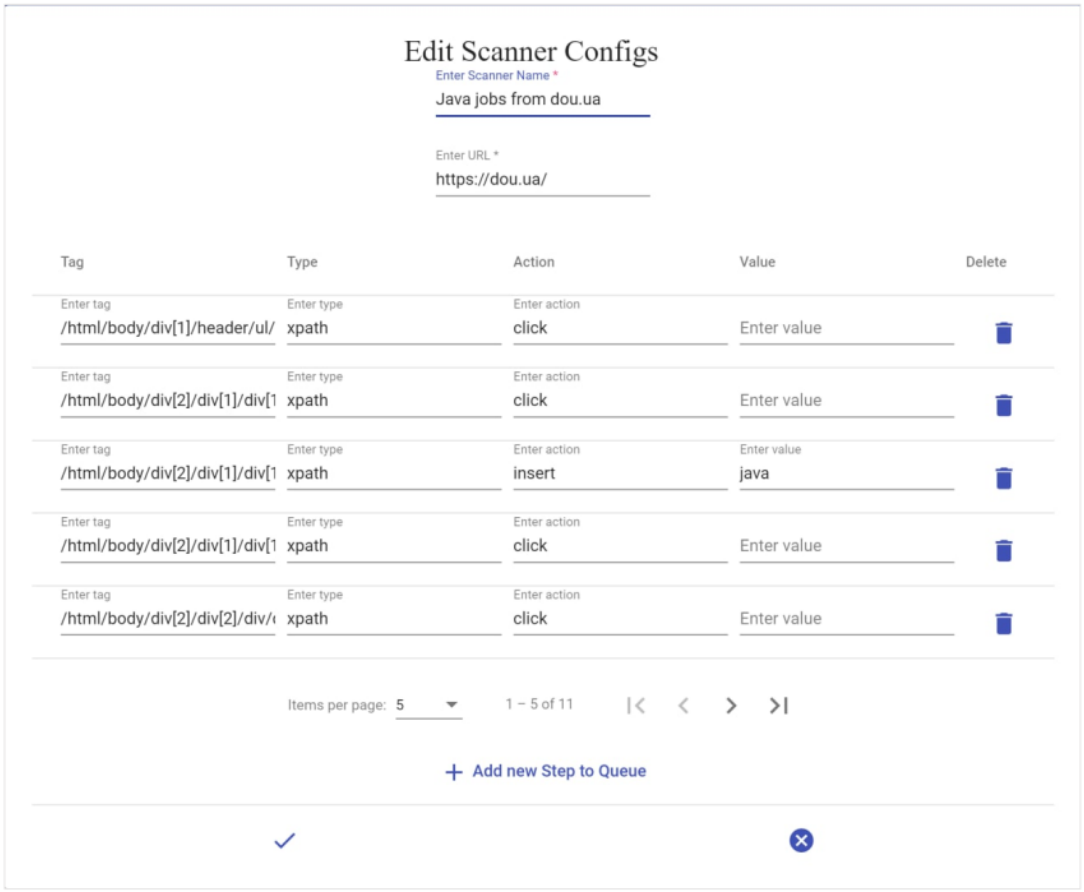

Рис. 8 – Налаштування сканера по www.jobs.dou.ua

Іншим параметром складності роботи алгоритму є кількість кліків, що здійснюється: чим більше кліків, тим складніше виконати видобуток даних. Кожний клік – це новий html-запит, який збільшує час виконання роботи парсера, тому що зажить від швидкості з'єднання, яке складно прогнозувати в теперішніх умовах.

Сформовані в результаті роботи сканерів файли, що містять видобуті з сайтів дані, наведені на рисунках 9-10.

	A	B	C	D	E	F	G	H	I	J
1	Crated date	Start time	End time	Name	Url	Tags		Created by	st modify date	Modify by
2	11/22/2022 11:39PM	11/22/2022 1:00AM	11/22/2022 1:05AM	Java jobs from work.ua	https://www.work.ua/	<class="form-control jobseeker-search">insert="java";<class="btn btn-search">click;<class="card card-hover card-visited wordwrap job-link">click,repeat,<class="pagination hidden-xs",class="no-style">click;<class="card wordwrap">scan;	Вакансія від 22 листопада 2022 Middle full stack developer (Java, JavaScript, angular 2+) 110 000 грн Люкс Асciст IT Дистанційна робота Повна зайнятість. Досвід роботи від 1 року. Вакансію зараз переглядає 1 шукач. Відгукніться на вакансію, щоб бути серед перших! Опис вакансії Компанія «Люкс-Асciст» шукає fullstack розробника для проектів для міжнародної страхової компанії, з якою співпрацюємо вже більше 4-х років Кількість проектів зросла, тому потрібен ще один фахівець. Робота в дружній злагодженій команді Обов'язково потрібний вільний усний рівень володіння англійською мовою. Вимоги: •Java •JavaScript (angular 2+) •Postgres SQL (не обов'язково, але буде плюсом) Умові праці: •повний робочий день •віддалена робота •оплачувані відпустки та лікарняні •оплата праці 90000–110000 грн/міс Чекаємо на ваше резюме			

Рис. 9 – Вихідний файл даних з парсингу сайту www.work.ua

p-ISSN: 2225-6733; e-ISSN: 2519-271X

	A	B	C	D	E	F	G	H	I	J
1	Crated date	Start time	End time	Name	Url	Tags		Created by	modify date	Modify by
2	11/22/2022 11:39PM	11/22/2022 1:00AM	11/22/2022 1:10AM	Java jobs from dou.ua	https://jobs.dou.ua/	<href="https://jobs.dou.ua/">click;<class="job">insert="Java";<class="btn-search">click;<class="l-vacancy __hot">click,repeat;<class="b-vacancy">scan;<class="l-vacancy">click,repeat,<class="more-btn">click;<class="b-vacancy">scan;	Micro Focus Всі вакансії компанії Micro Focus Світова IT корпорація зі штатом понад 13000 співробітників, що розробляє широкий спектр Enterprise software у сегментах DevOps, Security, Hybrid IT, BigData analytic. •Всі вакансії / Java / Київ 17 листопада 2022 Middle Java developer Київ, віддалено Job Description We are Ukraine R&D site of Software development company MicroFocus. More than 40,000 customers worldwide depend on solutions developed by more than 12,000 Micro Focus employees in 48 countries. Welcome to the team of professionals to work in friendly, family-style, innovative and stable environment. Must Haves •3+ years of hands-on experience working in core Java and advanced Java software development. •Strong Programming Skills in designing and implementation of multi-tier applications using Java, JDBC, REST API. •Strong analytical and problem-solving skills •Strong understanding of data structures, algorithms, OOPs concepts •Experience in working with Apache Tomcat application servers. •Strong experience in all the phases of software development life cycle including requirements gathering, analysis, design, implementation, deployment and support. •Experience in Java design patterns. •Experience in Industry standard Relational databases particularly Oracle, SQL Server and Postgres. •Experience in software testing, Junit testing, regression testing, defect tracking and management •Experience with Search Technologies particularly Apache Lucene •Hands-on experience with build tools and technologies, like Gradle, Jenkins, Ant, etc.. •Experience with secure development. •Familiar with Source Control versioning, branching •Familiar with Agile methodologies. •Excellent written and verbal communication skills, and problem-solving skills. Technical Skills Java Technologies: Core Java, JDBC, Spring Framework, Swing Search Technologies: Apache Lucene Web Technologies: RESTFull API Database Technologies: Oracle, SQL Server and Postgres Operating Systems: Windows, Linux Source Control: StarTeam, GIT Build Tools: Gradle, Ant, Continuous builds, Jenkins We offer •Possibility to work with highly qualified colleagues and possibility to work on interesting and challenging tasks and projects. •Opportunity to work in company that has strong position on the market and contribute to development of highly competitive			

Рис. 10 – Вихідний файл даних з парсингу сайту www.jobs.dou.ua

Проаналізовано результати виконання парсингу обох сайтів. Час виконання сканування сайтів склав 5 хвилин для www.work.ua та 10 хвилин для www.jobs.dou.ua. Загальна кількість вакансій на www.work.ua в категорії «IT, комп'ютери, інтернет» – 4220, на www.jobs.dou.ua – 4150.

Для виведення 14 вакансій по ключовому слову «java» з сайту www.work.ua для сканера потрібно було виконати 17 ітерацій, 2 з яких – циклічні ітерації з повтором по 6 і поверненням на одну сторінку, а також 3 теги. Усього сканер виконав 63 дій.

Для виведення 14 вакансій по ключовому слову «java» з сайту www.jobs.dou.ua для сканера потрібно було виконати 8 ітерацій, 1 з яких – циклічна з повтором 13 разів і поверненням на одну сторінку, а також 3 теги. Усього сканер виконав 59 дій.

Для порівняння з відкритих джерел було взято дані про швидкість роботи двох парсерів, розроблених на C# та на ANTLR відповідно, а саме результати з тестувань двох проєктів: CoreFX-1.0.0-rc2 та Roslyn-1.1.1 із однаковими параметрами (лексер 12%; парсер – 88%). Завдання з обробки 7329 файлів із загальною кількість рядків 2 286 274 та загальною кількістю символів 91 132 116 парсер Roslyn виконав за 4 секунди, а парсер ANTLR – за 25 секунди. Завдання з обробки 6527 файлів із загальною кількість рядків 1 967 672 та загальною кількістю символів 74 319 082 парсер Roslyn виконав за 3 секунди, а парсер ANTLR – за 16 секунд (рисунок 11).

Сканування парсером Aggregator, розробленим в рамках даної роботи, проводиться по заголовках та по тегах. У проведеному дослідженні швидкість підключення є нестабільним показником, який за власними замірами склав 1 сек. для сайту www.work.ua та до 1,5 сек. для сайту www.jobs.dou.ua. Для сканування сайту www.work.ua час на підключення склав: 63·1 = 63 сек. Враховуючи загальній час 5 хв. = 300 сек., сумарно на обробку було витрачено 300 – 63= 237 сек., при цьому на обробку однієї операції – 237/63 = 3,76 сек. Для сканування сайту www.jobs.dou.ua час на підключення склав 59·1,5 = 88,5 сек. Враховуючи загальній час 10 хв. = 600 сек., сумарно на обробку витрачено 600 – 88,5= 511,5 сек., при цьому на обробку однієї операції – 511,5/59 = 8,67 сек. (рисунки 11-12).

p-ISSN: 2225-6733; e-ISSN: 2519-271X

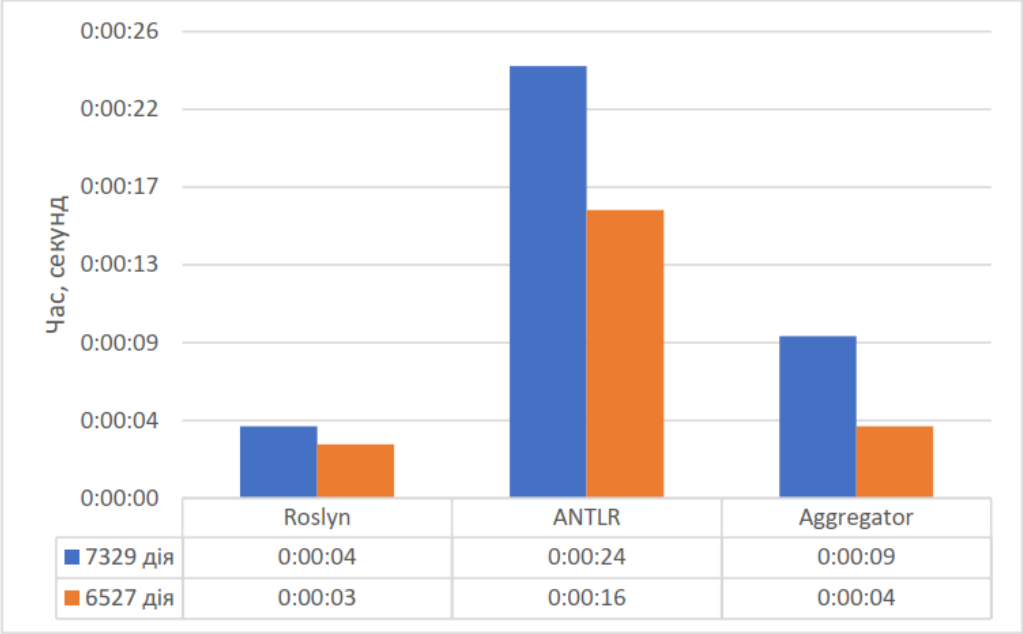

Рис. 11 – Порівняння роботи розробленого парсера та інших відомих парсерів

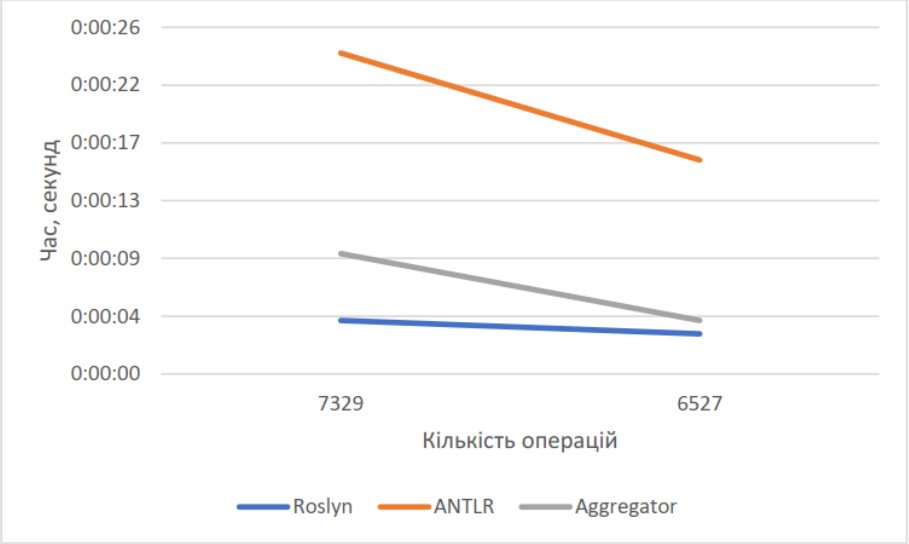

Рис. 12 – Залежність часу роботи парсера від кількості операцій

Результати дослідження показали, що розроблений парсер Aggregator працює на ряду з розглянутими парсерами і має прямопропорційну залежність часу виконання від кількості проведених операцій. Хоча Aggregator не може конкурувати з такими гігантами, як Roslyn, проте його все ж можна використати для парсингу, оскільки його швидкість досить висока і складає від 3,23 до 8,67 секунд (у середньому 5,95 секунди), що лише в 1,98 разів повільніше за корпоративно розроблені парсери. Зважаючи на нестабільне інтернет-підключення, отриманий результат є прийнятим, що доводить ефективність роботи створеного парсера для поставлених умов.

Висновки

Таким чином, розроблено інформаційну систему парсингу сайтів. Вебдодаток розміщений на хостингу: https://github.com/BearClumsy/aggregator та є доступним в будь-який час доби через браузер. У роботі докладно проаналізовано 10 існуючих парсерів, виділено їхні переваги та недоліки, що дозволило поставити задачу для розробки власного парсера.

Основними функціональними можливостями розробленого вебдодатку є: додавання та редагування налаштувань сканерів, редагування налаштувань сканеру, отримання інформації поза заданими критеріями пошуку, формування звітів та експортування отриманих даних.

Для розробки використано наступний стек технологій: СУБД Postgres SQL, мови програмування Java 11 та Python, фреймворки Angular 12, Node js, Npm, а також засоби з автоматизації роботи із проєктом Docker, Docker-compose, Gradle, Selenid.

Розроблений парсер Aggregator працює на ряду з розглянутими парсерами і має прямопропорційну залежність часу виконання від кількості проведених операцій. Швидкість його роботи складає від 3,23 до 8,67 секунд, що в середньому становить 5,95 секунд, що лише в 1,98 разів повільніше за відомі корпоративні додатки для парсингу статичного тексту. Зважаючи на те, що запропонований парсер використовується для сканування вебсторінок, а умови використання передбачали нестабільне інтернет-підключення, отриманий результат є прийнятим, а робота створеного парсера є ефективною.

Перелік використаних джерел:

1. Ratra R., Gulia P. Big Data tools and techniques: a roadmap for predictive analytics. *International Journal of Engineering and Advanced Technology (IJEAT)*. 2009. Vol. 9. Iss. 2. Pp. 4986-4992. DOI: https://doi.org/10.35940/ijeat.B2360.129219.
2. Tomar R.S. A Study on Web Scraping. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*. 2020. Vol. 8. Iss. 6. Pp. 1820-1824. DOI: https://doi.org/10.15662/IJAREEIE.2019.0806020.
3. Ateeq W. M. B., Al-Khalifa H. S. Intelligent framework for detecting predatory publishing venues. *IEEE Access*. 2023. Vol. 11. Pp. 20582-20618. DOI: https://doi.org/10.1109/ACCESS.2023.3250256.
4. EasySpider: EasySpider: A No-Code Visual System for Crawling the Web / Wang N., Feng W., Yin J., Ng S.-K. *WWW* '23 *Companion* : Companion Proceedings of the ACM Web Conference 2023, Austin, TX, USA, 30 April - 4 May 2023. Pp. 192-195. DOI: https://doi.org/10.1145/3543873.3587345.
5. A classification framework for data marketplaces / Stahl F., Schomm F., Vossen G., Vomfell L. *Vietnam Journal of Computer Science*. 2016. Vol. 3. Pp. 137-143. DOI: https://doi.org/10.1007/s40595-016-0064-2.
6. Kirichenko L., Radivilova T., Carlsson A. Detecting cyber threats through social network analysis: short survey. *SocioEconomic Challenges*. 2017. Vol. 1. Iss. 1. Pp. 20-34. DOI: https://doi.org/10.21272/sec.2017.1-03.
7. Exploring Web Scraping with Python / Sasi A., Deep A., Kumar K., Birla V. *Machine Intelligence and Smart Systems* : Proceedings of the I International Conference, Gwalior, India, 24-25 September 2020. Pp. 287-296. DOI: https://doi.org/10.1007/978-981-33-4893-6_26.
8. Selenide. Concise UI tests in Java. URL: https://selenide.org/documentation/selenide-vs-selenium.html (дата звернення: 28.08.2024).

References:

1. R. Ratra, and P. Gulia, «Big Data tools and techniques: a roadmap for predictive analytics», *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 9, iss. 2, pp. 4986-4992, 2009. **doi: 10.35940/ijeat.B2360.129219**.
2. R.S. Tomar, «A Study on Web Scraping», *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 8, iss. 6, pp. 1820-1824, 2020. **doi: 10.15662/IJAREEIE.2019.0806020**.
3. W.M.B. Ateeq, H.S. Al-Khalifa, «Intelligent framework for detecting predatory publishing venues», *IEEE Access*, vol. 11, pp. 20582-20618, 2023. **doi: 10.1109/ACCESS.2023.3250256**.
4. N. Wang, W. Feng, J. Yin, and S.-K. Ng, «EasySpider: EasySpider: A No-Code Visual System for Crawling the Web», in Proc. ACM Web Conference 2023, USA, 2023, pp. 192-195. **doi: 10.1145/3543873.3587345**.
5. F. Stahl, F. Schomm, G. Vossen, and L. Vomfell, «A classification framework for data marketplaces», *Vietnam Journal of Computer Science*, vol. 3, p. 137-143, 2016. **doi: 10.1007/s40595-016-0064-2**.

6. L. Kirichenko, T. Radivilova, and A. Carlsson, «Detecting cyber threats through social network analysis: short survey», *SocioEconomic Challenges*, vol. 1, iss. 1, pp. 20-34, 2017. **doi: 10.21272/sec.2017.1-03**.
7. A. Sasi, A. Deep, K. Kumar, and V. Birla, «Exploring Web Scraping with Python», in Proc. I Int. Conf. «Machine Intelligence and Smart Systems», Gwalior, India, 2020, pp. 287-296. **doi: 10.1007/978-981-33-4893-6_26**.
8. Selenide. Concise UI tests in Java. [Online]. Available: https://selenide.org/documentation/selenide-vs-selenium.html. Accessed on: August 28, 2024.

Стаття надійшла 15.09.2024
Стаття прийнята 10.10.2024

УДК 004.4:004.94 **doi: 10.31498/2225-6733.49.1.2024.321181**

© П'ятикоп О.Є.[1], Гніденко В.А.[2], Воротнікова З.Є.[3]

МОБІЛЬНИЙ ЗАСТОСУНОК АНАЛІЗУ ТА ПРОГНОЗУВАННЯ ОСОБИСТИХ ВИТРАТ

У сучасному світі все частіше використовуються електронні платіжні системи, мобільні додатки для платежів і цифрові банківські послуги, створюючи попит на більш точні та зручні інструменти для управління особистими витратами. Це дослідження представляє розробку інтелектуальної системи для аналізу та прогнозування особистих витрат, інтегрованої в кросплатформний мобільний додаток. Дослідження включає аналіз останніх досліджень та публікацій, які підтвердили, що впровадження аналітичних та інтелектуальних систем управління витратами значно підвищує ефективність фінансового контролю та дозволяє своєчасно приймати обґрунтовані рішення. Огляд літератури також показав, що інтеграція таких систем з мобільними додатками є актуальною, затребуваною та забезпечує зручність і доступність в управлінні фінансами. Для розробки мобільного додатку для аналізу та прогнозування особистих витрат на основі історичних даних користувачів були вивчені такі методи прогнозування, як експоненційно зважене ковзне середнє (EWMA) і лінійна регресія. Оцінка цих методів продемонструвала, що EWMA постійно досягає вищої точності, виміряної WAPE та MAPE. За результатами цього створено прототип мобільного додатку, який використовує історичні дані транзакцій, отримані через відкритий API від Monobank, класифікує витрати за категоріями та застосовує вибраний метод для надання користувачам персоналізованих прогнозів. Щоб забезпечити кросплатформну функціональність, мобільний додаток було розроблено з використанням технологій Dart і Flutter. Розроблений прототип має інтуїтивно зрозумілий інтерфейс із графічною візуалізацією результатів аналізу та прогнозу у вигляді діаграм та графіків. Запропонований підхід сприяє раціоналізації витрат, підвищує ефективність фінансового планування та підвищує фінансову грамотність користувачів.

Ключові слова: *аналіз витрат, прогнозування, мобільний додаток, персональні фінанси, експоненційне згладжування, EWMA, лінійна регресія, WAPE, MAPE, Flutter.*

[1] *канд. техн. наук, доцент, ДВНЗ «Приазовський державний технічний університет», м. Дніпро, ORCID: 0000-0002-7731-3051, piatykop_o_ye@pstu.edu*
[2] *магістр, ДВНЗ «Приазовський державний технічний університет», м. Дніпро*
[3] *канд. техн. наук, доцент, ДВНЗ «Приазовський державний технічний університет», м. Дніпро, ORCID: 0000-0002-0746-5981, vorotnikova_z_j@pstu.edu*