Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики
(інститут)

Факультет інформаційних технологій
(факультет)

Кафедра Програмного забезпечення комп'ютерних систем
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня

бакалавра
(назва освітньо-кваліфікаційного рівня)

студента *Крамаренко Нікіти Васильовича*
(ПІБ)

академічної групи *122-19-3*
(шифр)

спеціальності *122 Комп'ютерні науки*
(код і назва спеціальності)

освітньої програми *Комп'ютерні науки*
(назва освітньої програми)

на тему: *Розробка інформаційної технології аналізу даних з відкритих джерел*

Керівники	**Прізвище, ініціали**	**Оцінка за шкалою**		**Підпис**
		рейтинговою	**інституційною**	
кваліфікаційної роботи	*проф. Лактіонов І.С.*	85	добре	
розділів:				
спеціальний	*проф. Лактіонов І.С.*	85	добре	
економічний	*проф. Вагонова О.Г*			
Рецензент				
Нормоконтролер	*доц. Гуліна І.Г.*			

Дніпро
2023

Міністерство освіти і науки України
НТУ «Дніпровська політехніка»

ЗАТВЕРДЖЕНО:
завідувач кафедри
програмного забезпечення комп'ютерних систем
(повна назва)

_____ М.О. Алексєєв
(підпис) (прізвище, ініціали)

« » _____ 2023 року

ЗАВДАННЯ
на кваліфікаційну роботу

бакалавра
(назва освітньо-кваліфікаційного рівня)

студента *122-19-3* *Крамаренко Н.В.*
(група) (прізвище та ініціали)

тема кваліфікаційної роботи *Розробка інформаційної технології аналізу даних з відкритих джерел*

затверджена наказом ректора НТУ «ДП»
від 16.05.2023 № 350-с

Розділ	Зміст виконання	Термін виконання
Спеціальний	***На основі матеріалів проєктно-технологічної практики та інших науково-технічних джерел провести аналіз стану рішення проблеми та постановку задачі. Обґрунтувати вибір та здійснити реалізацію методів вирішення проблеми***	13.05.2023 р.
Економічний	***Провести розрахунок трудомісткості розробки програмного забезпечення, витрат на створення ПЗ й тривалості його розробки***	27.05.2023 р.

Завдання видав _____ *проф. Лактіонов І.С.*
(підпис) (посада, прізвище, ініціали)

Завдання прийняв до виконання _____ *Крамаренко Н.В.*
(підпис) (прізвище, ініціали)

Дата видачі завдання: *14.01.2023 р*

Термін подання кваліфікаційної роботи до ЕК: *12.06.2023 р*

РЕФЕРАТ

Пояснювальна записка: 56 с., 13 рис., 0 табл., 4 дод., 20 джерел.

Об'єкт розробки: інформаційна система аналізу даних з відкритих джерел.

Мета кваліфікаційної роботи: створення інформаційної системи аналізу даних з відкритих джерел, яка дозволяє інтернет-магазину отримувати дані про ціни, наявний асортимент та відгуки клієнтів для надання бізнесу можливостей ефективного аналізу конкурентів і ухвалення обґрунтованих рішень щодо стратегічного позиціонування себе на ринку і відповідного коригування цінової та маркетингової стратегії.

У вступі розглядається аналіз та сучасний стан проблеми, конкретизується мета кваліфікаційної роботи та галузь її застосування, наведено обґрунтування актуальності теми та уточнюється постановка завдання. У першому розділі проаналізовано предметну галузь, визначено актуальність завдання та призначення розробки, сформульовано постановку завдання, зазначено вимоги до програмної реалізації, технологій та програмних засобів. У другому розділі проаналізовані наявні рішення, обрано платформи для розробки, виконано проектування і розробка програми, описана робота програми, алгоритм і структура її функціонування, а також виклик та завантаження програми, визначено вхідні і вихідні дані, охарактеризовано склад параметрів технічних засобів.

В економічному розділі визначено трудомісткість розроблення програмного забезпечення, проведений підрахунок вартості робот по створенню програми та розраховано час на його створення.

Практичне значення полягає у створенні програмного забезпечення, що надає можливість моніторингу цін, відстежуючи ціни конкурентів у режимі реального часу і дозволяючи магазину відповідно коригувати свої ціни забезпечуючи конкурентне ціноутворення та автоматизованого збору інформації про товари, наприклад, специфікацій і оцінок клієнтів, що дозволяє магазину підтримувати актуальний і всеосяжний каталог товарів.

Актуальність даної інформаційної системи визначається тим, що велика кількість інформації у сучасному цифровому середовищі надає широкі можливості для бізнесу, проте ручне збирання та аналіз такої величезної кількості даних вимагає багато часу і ресурсів. Використовуючи сучасні технології отримання та аналізу даних з відкритих джерел, інтернет-магазини можуть автоматизувати процес збору даних, значно скоротивши зусилля, необхідні для збору цінної інформації. Така ефективність дає конкурентну перевагу, дозволяючи підприємствам більше зосередитися на аналізі даних і виробленні стратегії.

Ключові слова: ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ, ВИЛУЧЕННЯ ДАНИХ, АНАЛІЗ ДАНИХ, ВЕБ-СКРЕПІНГ, ВІДКРИТІ ДЖЕРЕЛА, ЕЛЕКТРОННА КОМЕРЦІЯ, МОНІТОРИНГ ЦІН, ДОСЛІДЖЕННЯ РИНКУ, АНАЛІЗ КОНКУРЕНТІВ, МАРКЕТИНГОВА АНАЛІТИКА, ПРИЙНЯТТЯ РІШЕНЬ НА ОСНОВІ ДАНИХ

ABSTRACT

Explanatory note: 56 pages., 13 fig., 0 table, 4 appendix, 20 sources.

Object of development: information technology for analyzing data from open sources.

The purpose of qualifying work: creation of an information system for analyzing data from open sources that allows an online store to obtain data on prices, available assortment, and customer feedback to enable businesses to effectively analyze competitors and make informed decisions on strategic market positioning and adjust pricing and marketing strategies accordingly.

The introduction considers the analysis and the current state of the problem, specifies the purpose of the qualification work and the field of its application, provides a justification for the relevance of the topic and clarifies the problem.

In the first section the subject branch is analyzed, the urgency of the task and purpose of development are defined, the statement of the task is formulated, requirements to software realization, technologies and software are specified.

The second section analyzes the existing solutions, selects platforms for development, design and development of the program, describes the program, algorithm and structure of its operation, as well as calling and loading the program, determines the input and output data, describes the parameters of hardware.

In the economic section, the complexity of software development is determined, the cost of work and the time on creation of the program is calculated.

The practical value lies in the creation of software that provides the ability to monitor prices, tracking competitors' prices in real time and allowing the store to adjust its prices accordingly, ensuring competitive pricing and automated collection of product information, such as specifications and customer ratings, allowing the store to maintain an up-to-date and comprehensive product catalogue.

The relevance of the information system is determined by the fact that the abundance of information in the modern digital environment provides ample opportunities for business, but manual collection and analysis of such a huge amount of data requires a lot of time and resources. By using modern technologies for obtaining and analyzing data from open sources, online retailers can automate the data collection process, significantly reducing the effort required to collect valuable information. This efficiency provides a competitive advantage, allowing businesses to focus more on data analysis and strategy development.

Keywords: INFORMATION TECHNOLOGY, DATA EXTRACTION, DATA ANALYSIS, WEB SCRAPING, OPEN SOURCES, E-COMMERCE, PRICE MONITORING, MARKET RESEARCH, COMPETITOR ANALYSIS, MARKETING ANALYTICS, DATA-DRIVEN DECISION MAKING

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ПЗ – програмне забезпечення

API - Application Programming Interface

CAPTCHA - Completely Automated Public Turing test to tell Computers and Humans Apart

CSS - Cascading Style Sheets

CSV - Comma-Separated Values

EDN - Extensible Data Notation

HTML - HyperText Markup Language

HTTP - HyperText Transfer Protocol

HTTPS - HyperText Transfer Protocol Secure

IP - Internet Protocol

JDK - Java Development Kit

JSON - JavaScript Object Notation

JVM - Java Virtual Machine

SKU - Stock Keeping Unit

SSL - Secure Sockets Layer

TLS - Transport Layer Security

URL - Uniform Resource Locator

WWW - World Wide Web

XML - Extensible Markup Language

ЗМІСТ

ВСТУП

У сфері збору даних комп'ютерні науковці використовують свої знання веб-технологій та мов програмування, щоб програмно отримувати дані з веб-сайтів та відкритих джерел. Вони орієнтуються в складних структурах веб-сайтів за допомогою алгоритмів, розуміють основні мови розмітки, такі як HTML і XML, і використовують API або розробляють власні скрипти для вилучення необхідних даних.

Веб-технології та протоколи складають основу систем веб-скрепінгу, і комп'ютерні вчені володіють глибоким розумінням цих технологій. Вони використовують такі протоколи, як HTTP і SSL/TLS, для безпечного зв'язку з веб-сервісами, взаємодії з API для отримання даних, а також для автентифікації та передачі даних. Володіння веб-технологіями дозволяє комп'ютерним науковцям ефективно орієнтуватися в складнощах веб-скрепінгу та забезпечувати безперешкодну інтеграцію з відкритими джерелами даних [1].

Сфера застосування такої системи є широкою та універсальною. Її можна застосовувати в різних сферах і галузях, включаючи електронну комерцію, маркетингові дослідження, конкурентний аналіз, оптимізацію ціноутворення, моніторинг тенденцій, аналіз настроїв і багато іншого. Для інтернет-магазину система може використовуватися для вилучення інформації про товари, відгуки клієнтів, дані про ціни та інформацію про конкурентів з різних джерел. Ці дані можна аналізувати для виявлення ринкових тенденцій, моніторингу динаміки цін, оптимізації товарних пропозицій та покращення загальної бізнес-стратегії [2].

Актуальність збору даних з відкритих джерел зумовлена величезною цінністю, яку вони приносять бізнесу в сучасну цифрову епоху. В умовах експоненціального зростання обсягів даних в Інтернеті організаціям потрібні ефективні та автоматизовані методи вилучення, аналізу та використання цієї інформації на свою користь.

Веб-скрепінг дозволяє компаніям отримати конкурентну перевагу. На висококонкурентному ринку, щоб залишатися попереду, потрібно глибоке

розуміння дій конкурентів. Веб-скрейпінг дозволяє організаціям відстежувати цінову політику конкурентів, їхні товарні пропозиції, рекламні акції та відгуки клієнтів. Ці знання дають можливість компаніям вдосконалювати власні стратегії, диференціювати свої пропозиції та оперативно реагувати на ринкові зміни [3].

Крім того, веб-скрепінг полегшує дослідження ринку та бізнес-аналітику. Витягуючи дані з різних джерел, компанії можуть збирати інформацію про вподобання клієнтів, галузеві тенденції та сегменти ринку, що розвиваються. Ці знання дозволяють організаціям виявляти нові можливості, адаптувати свої продукти або послуги до потреб клієнтів та оптимізувати свої маркетингові зусилля.

Конкретне завдання кваліфікаційної роботи полягає в розробці програмного рішення для інтернет-магазину, яке автоматизує вилучення релевантних даних з відкритих джерел в Інтернеті. Система повинна бути призначена для навігації по веб-сайтах, отримання конкретної інформації, такої як інформація про товари, ціни, відгуки клієнтів і дані про конкурентів, а також для зберігання зібраних даних у структурованому форматі для подальшого аналізу та використання.

РОЗДІЛ 1
АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1. Загальні відомості з предметної галузі

Предметну область, яка охоплює збір і аналіз даних, можна загалом назвати "видобуванням веб-даних" або "інтелектуальним аналізом веб-даних". Ці терміни охоплюють процеси і методи, пов'язані з видобуванням, збором і аналізом даних з різних онлайнових джерел, насамперед веб-сайтів. Веб-скрепінг - це конкретний акт вилучення даних з веб-сторінок, тоді як аналіз даних включає обробку, інтерпретацію та отримання інсайдів на основі вилучених даних. Поєднання веб-скрепінгу та аналізу даних дозволяє компаніям і дослідникам отримувати цінну інформацію, приймати обґрунтовані рішення та використовувати її в різних сферах, таких як дослідження ринку, конкурентний аналіз, моніторинг тенденцій тощо.

У сфері веб-скрепінгу Інтернет слугує сховищем даних. HTTP, протокол, який використовується для зв'язку між веб-клієнтами і серверами, дозволяє обмінюватися запитами і відповідями, необхідними для веб-скрепінгу. HTTP базується на моделі клієнт-сервер, де клієнт (наприклад, веб-браузер) надсилає запит до сервера, а сервер відповідає запитуваними даними. Зв'язок між клієнтом і сервером відбувається за допомогою HTTP-повідомлень, які складаються із запиту від клієнта і відповідної відповіді від сервера [4]. URL-адреси слугують адресами для пошуку певних ресурсів в Інтернеті. Вони використовуються для пошуку та доступу до різних ресурсів, таких як веб-сторінки, зображення, файли тощо. URL складається з декількох компонентів, які надають інформацію про місцезнаходження ресурсу та способи його отримання [5]. Роблячи HTTP-запити до цільових URL-адрес, веб-скрепери отримують HTML-вміст веб-сторінок. який потім можна розбирати і аналізувати для вилучення потрібних даних. HTML - це мова розмітки, яка використовується для структурування вмісту веб-сторінок. Вона містить набір елементів і тегів, які визначають різні частини веб-

сторінки, такі як заголовки, абзаци, посилання, зображення, таблиці та форми. HTML відповідає за структуру і семантику контенту, дозволяючи браузерам та іншому програмному забезпеченню інтерпретувати і представляти його користувачам [6]. Приклад HTML-сторінки можна побачити на рис. 1.1.

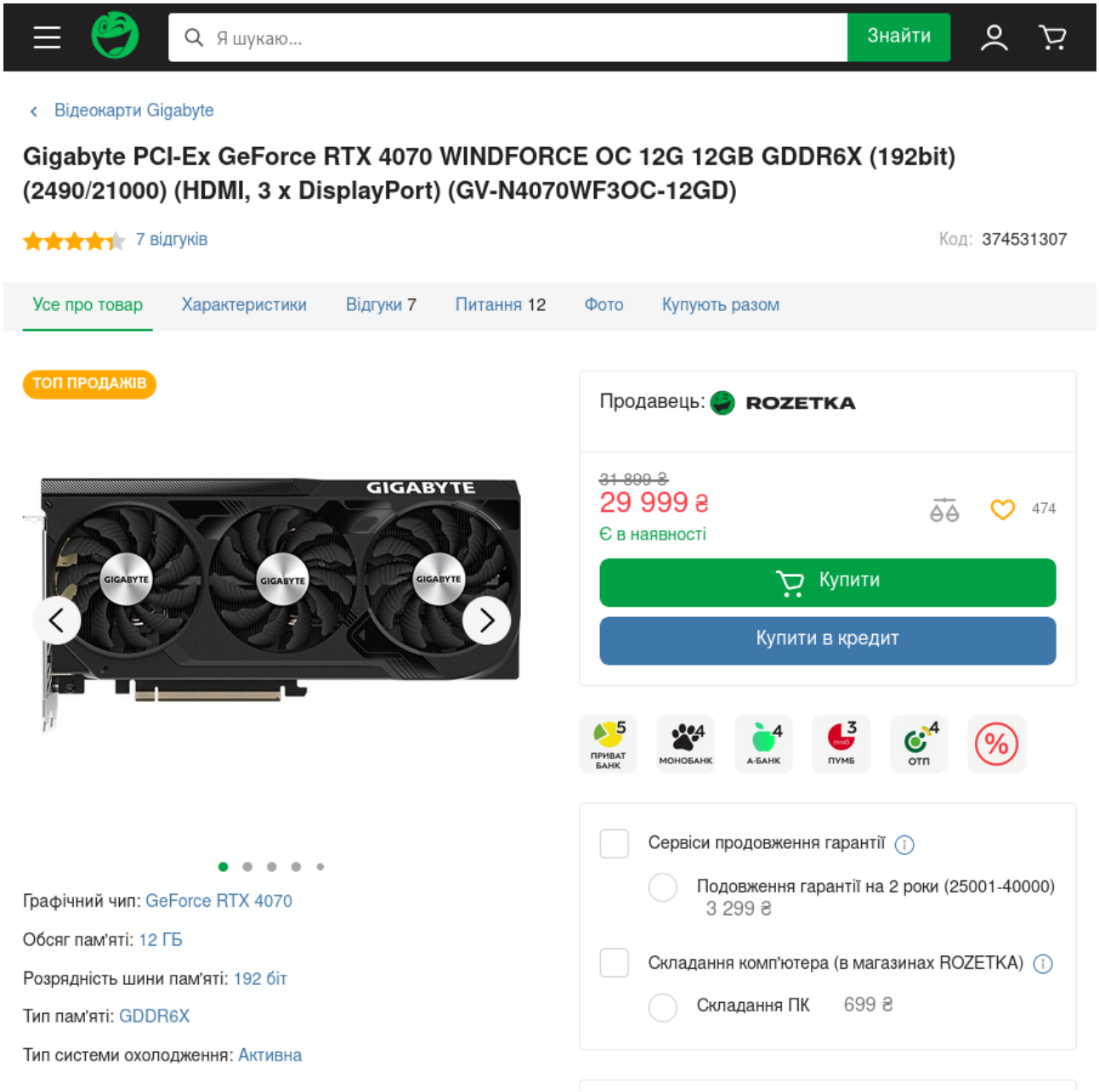

Рис. 1.1. Приклад HTML-сторінки товару з назвою, вартістю та характеристиками товару, що відображається за допомогою веб-браузера, який відправляє HTTP-запит на певну URL-адресу, що відповідає місцезнаходженню сторінки з товаром на сервері інтернет-магазину і отримуючи цю сторінку виводить її користувачеві

Вилучення даних з Інтернету передбачає вилучення даних з цих веб-сайтів і веб-сторінок, як правило, в автоматизованому режимі за допомогою програмних інструментів або скриптів. Інтернет є джерелом даних, які вилучаються, і слугує середовищем, за допомогою якого інформація поширюється та обмінюється в Інтернеті [7].

У сфері збору та аналізу даних поточний стан проблеми свідчить про значний прогрес у технологіях і методологіях. Існує кілька аналогів та інструментів, які допомагають у вирішенні завдань веб-скрепінгу, починаючи від бібліотек з відкритим вихідним кодом і закінчуючи комерційними програмними рішеннями. Ці інструменти надають розробникам можливість витягувати дані з різних онлайн-джерел, роблячи процес більш ефективним і доступним [8].

Незважаючи на значний прогрес у вирішенні завдань веб-скрепінгу, проблеми та технічні протиріччя все ще існують. Для запобігання несанкціонованому доступу та вилученню даних веб-сайти використовують такі заходи, як CAPTCHA, механізми антискрепінгу та динамічне відображення контенту. Подолання цих технічних перешкод вимагає розробки складних методів і стратегій скрепінгу, таких як ротація IP-адрес, управління сесіями та інтелектуальні алгоритми синтаксичного аналізу [9].

Існують прогалини в знаннях у галузі веб-скрепінгу, особливо в таких сферах, як обробка складних структур веб-сторінок, ефективне вилучення даних зі сторінок, відрендерених на JavaScript, і робота з нестандартними форматами даних.

Крім того, існують нездійснені вимоги до виробів і розробок у цій галузі. Забезпечення точності та надійності вилучених даних, обробка великомасштабних операцій вилучення та створення систем, здатних адаптуватися до змін у структурі веб-сайтів, є постійними викликами. З організаційної точки зору, питання, пов'язані з етичними практиками вилучення даних, конфіденційністю даних і дотриманням правових норм, залишаються критично важливими. Організаціям необхідно орієнтуватися в етичних і

правових межах веб-скрепінгу, щоб забезпечити відповідальну і законну практику збору даних.

Розглянемо кілька продуктів, присутніх на ринку, які використовують можливості збору та аналізу даних з відкритих джерел як свою основну послугу.

ScrapingAnt спеціалізується на наданні масштабованих і надійних послуг веб-скрепінгу. Вони вирішують такі проблеми, як рендеринг JavaScript і CAPTCHA, які є поширеними перешкодами у веб-скрепінгу. Їхній сервіс використовує пул IP-адрес для забезпечення успішного вилучення даних і уникнення блокування IP-адрес. Як і будь-яка служба веб-скрепінгу, ScrapingAnt може зіткнутися з випадковими обмеженнями або проблемами при вилученні певних веб-сайтів або обробці заходів протидії вилученню. Доступність і продуктивність їхнього пулу IP-адрес може змінюватися залежно від використання та попиту.

DataForSEO пропонує спеціалізований постачальник SEO-даних на основі API. Їхній API дозволяє користувачам отримати доступ до широкого спектру даних пошукових систем, рейтингу ключових слів та іншої інформації, пов'язаної з SEO, для аналізу та звітності. Вони надають вичерпну документацію та мають дружній підхід до розробників. Хоча DataForSEO фокусується на даних, пов'язаних з SEO, їхні послуги можуть бути більш обмеженими з точки зору загального веб-скрепінгу порівняно з іншими провайдерами. Користувачам, які шукають більш широкі можливості вилучення веб-даних, можливо, доведеться розглянути додаткові інструменти або провайдерів.

Zyte пропонує комплексну платформу для веб-скрепінгу з розширеними можливостями, такими як вилучення даних за допомогою штучного інтелекту. Їхній продукт AutoExtract автоматизує процес вилучення даних, полегшуючи його для користувачів без глибоких технічних знань. Вони мають хорошу репутацію в галузі та пропонують надійні та масштабовані рішення. Ціни на послуги Zyte можуть бути відносно вищими, ніж у деяких інших постачальників. Як і у випадку з будь-яким інструментом автоматизованого вилучення, точність

вилучення даних значною мірою залежить від структури та складності цільових веб-сайтів.

Таким чином, ці компанії надають рішення та послуги для вилучення та аналізу даних. Вони допомагають клієнтам збирати дані з різних онлайн-джерел, обробляти та очищати їх і надавати у зручному форматі для подальшого аналізу або використання.

1.2 Призначення розробки та галузь застосування

Оскільки основне призначення системи полягає в зборі та аналізі даних конкурентів, система має назву "Compis", що походить від Competitive Intelligence System.

Система збору та аналізу даних для інтернет-магазинів включає в себе кілька ключових компонентів. По-перше, збір даних - це процес збору релевантної інформації з різних джерел, таких як веб-сайти, API та бази даних. Веб-скрепінг, метод, який автоматизує вилучення даних з веб-сайтів, часто використовується для збору інформації про товари, ціни, відгуки клієнтів та інші релевантні дані.

Поява системи для інтернет-магазину зумовлена зростаючою потребою у прийнятті рішень на основі даних та підтримці конкурентоспроможності на цифровому ринку. Оскільки інтернет-магазини продовжують розширювати свою присутність і асортимент, наявність релевантних і актуальних даних стає вирішальним фактором успіху. Система має на меті задовольнити цю потребу, надаючи надійне та ефективне рішення для вилучення та аналізу даних.

Система, що розробляється, може бути використана в різних сферах діяльності інтернет-магазину. Одним з важливих застосувань є моніторинг цін. Шляхом сканування сайтів конкурентів та відстеження цінових даних інтернет-магазин може бути в курсі ринкових тенденцій, визначати цінові стратегії конкурентів та відповідно коригувати власну цінову політику. Це допомагає підтримувати конкурентоспроможність і максимізувати прибуток.

Ще одна сфера, де система може бути корисною, - це аналіз конкурентів, що дозволяє інтернет-магазину збирати дані про товари конкурентів, ціни, рекламні заходи та відгуки клієнтів. Цей аналіз дає цінну інформацію про сильні та слабкі сторони конкурентів, що дозволяє магазину визначити сфери диференціації та сформулювати ефективні конкурентні стратегії.

Загалом, розробка системи задовольняє потребу в прийнятті рішень на основі даних, ефективному зборі даних та конкурентній розвідці в індустрії інтернет-магазинів. Автоматизуючи вилучення та аналіз даних, система дає можливість магазину приймати обґрунтовані рішення, оптимізувати операції та отримати конкурентну перевагу на динамічному онлайн-ринку.

1.3. Підстава для розробки

В кінці навчання, студент виконує кваліфікаційну роботу (проект). Тема роботи узгоджується з керівником проекту, випускаючою кафедрою.

Підставою для розробки кваліфікаційної роботи на тему «Розробка інформаційної технології аналізу даних з відкритих джерел» є наказ по Національному технічному університету «Дніпровська політехніка» від 16.05.2023р. № 350-с

1.4. Постановка завдання

По-перше, потрібно ідентифікувати та вибрати відповідні веб-сайти конкурентів для моніторингу, потім визначитися з цільовими сторінками і розділами, з яких витягуватимуть дані. Це передбачає визначення URL-адрес і розуміння структури веб-сайтів.

Після цього треба впровадити механізми синтаксичного аналізу даних для вилучення та перетворення відповідної інформації з отриманого HTML-контенту.

Це передбачає визначення відповідних тегів HTML та селекторів CSS для точного пошуку та вилучення потрібних елементів даних. CSS-селектори є стандартним способом націлювання на елементи на HTML-сторінці. Вони надають можливість вибирати елементи на основі їхніх атрибутів, властивостей, зв'язків або позиції в структурі документа [10]. Приклад отриманного CSS-селектора можна побачити на рис 1.2.

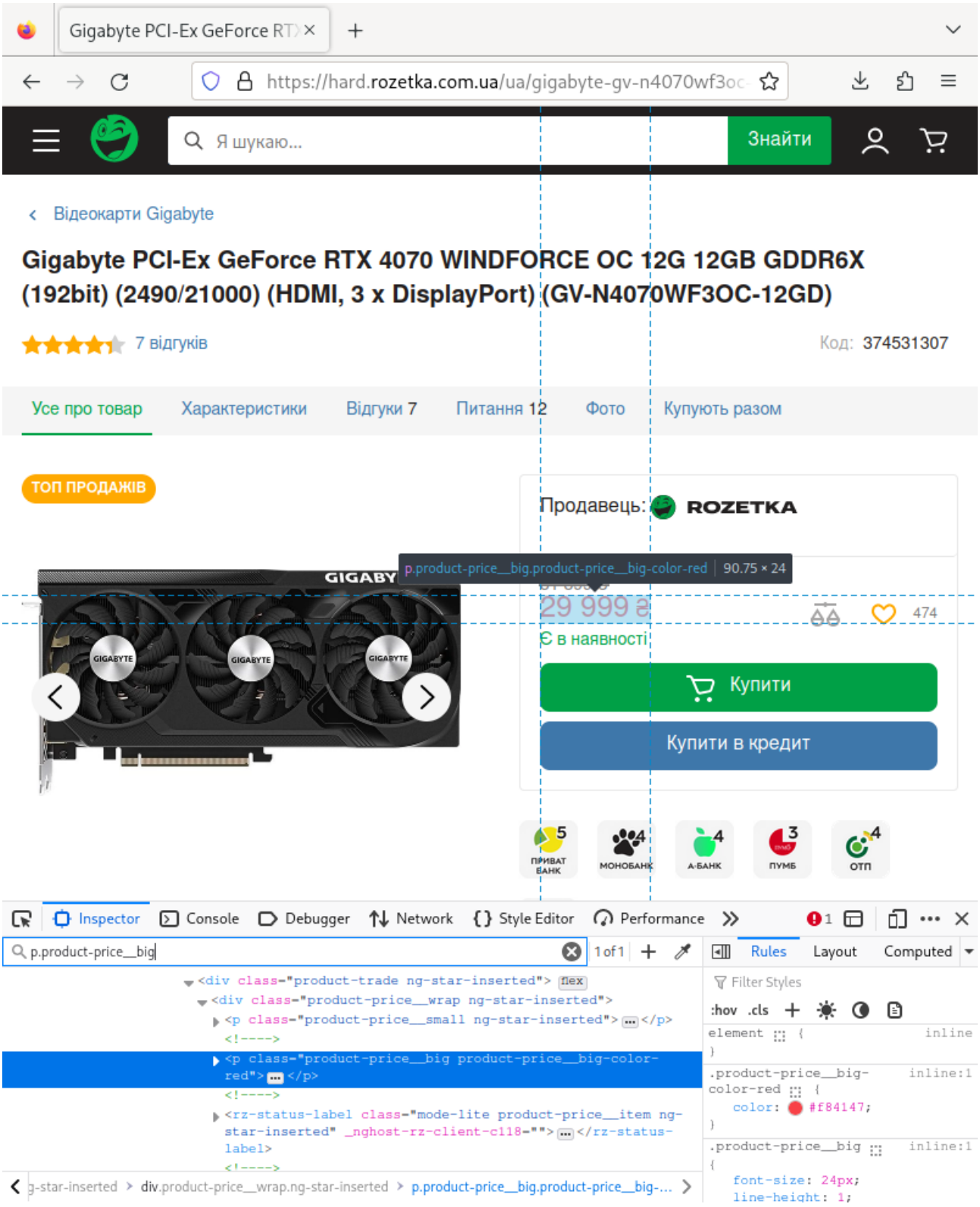

Рис. 1.2. Приклад використання інструментів розробника, присутніх у сучасних веб-браузерах, які надають можливість в інтерактивному режимі виділити певний елемент на сторінці та отримати CSS-селектор для цього елемента

Також, може потребуватися виконання завдання з очищення та перевірки даних, щоб забезпечити точність і надійність зібраних даних. Це передбачає видалення небажаних символів, нормалізацію форматів, обробку відсутніх або суперечливих даних і перевірку цілісності видобутої інформації.

Нарешті, вилучені дані потрібно зберегти та організовувати для подальшого аналізу. Треба розробити методи організації та структурування даних для ефективного зберігання та управління зібраною інформацією для подальшого аналізу. Це може включати використання баз даних, таких як PostgreSQL або MongoDB, або форматів файлів, таких як CSV або JSON.

1.5. Вимоги до програми або програмного виробу

1.5.1. Вимоги до функціональних характеристик

Для отримання інформації про товар у підтримуваних системою магазинах достатньо мати інформацію лише про SKU товару, який є унікальним буквено-цифровим ідентифікатором, який присвоюється конкретному товару або позиції в управлінні запасами. Ці ідентифікатори використовуються ритейлерами для відстеження та управління запасами. Кожен SKU відповідає певному варіанту товару, наприклад, розміру, кольору або стилю [11]. При інтеграції програмного компонента в наявну систему інтернет-магазину, користувач, який знає цей код і використовує систему, може знайти за ним таку інформацію, як вартість, деталі товару, найпопулярніші та найновіші товари тощо, серед магазинів конкурентів для проведення подальшого аналізу. Система поверне дані користувачеві в структурованому форматі даних, який може бути підданий програмній обробці для подальшого аналізу. Користувач також може отримати інформацію про товар у конкретному магазині, вказавши магазин разом із SKU, або URL на сам товар у магазині. Приклад SKU товару можна побачити на рис 1.3.

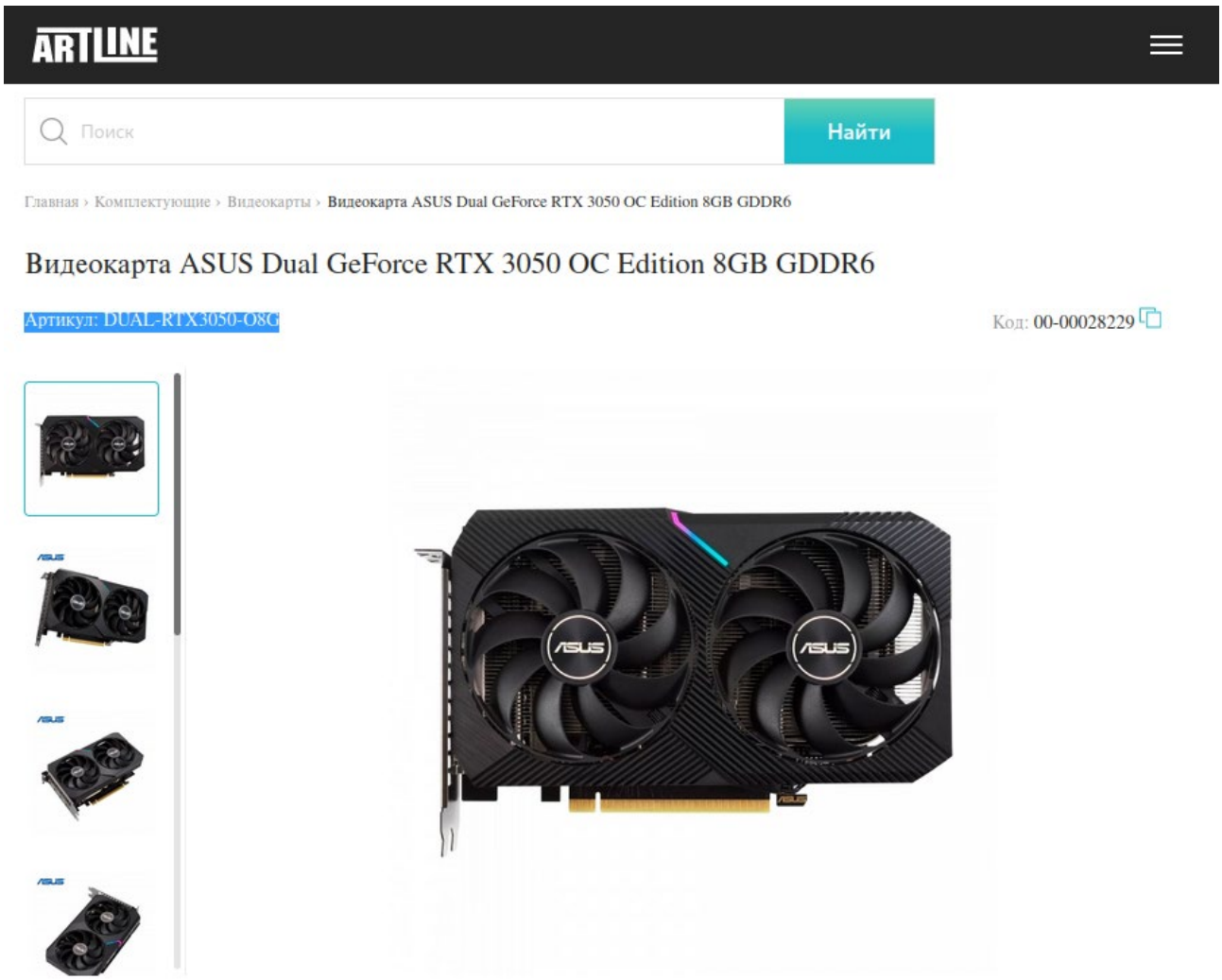

Рис. 1.3. Приклад SKU на сторінці товару. В україномовному середовищі SKU часто називають артикулами товару

1.5.2. Вимоги до інформаційної безпеки

При розробці системи вимоги до інформаційної безпеки включають використання захищених протоколів зв'язку, таких як HTTPS, і впровадження TLS або SSL шифрування для встановлення захищених каналів зв'язку між системою та веб-сайтами, з яких вилучаються дані. HTTPS - це захищений протокол зв'язку, який використовується для захисту цілісності та конфіденційності даних, що передаються через Інтернет [12]. Він поєднує протокол передачі гіпертексту HTTP з можливостями шифрування TLS, створюючи безпечний і зашифрований канал між клієнтом (зазвичай веб-браузером) і сервером. При використанні HTTPS дані, що передаються між

клієнтом і сервером, шифруються і не можуть бути легко перехоплені або підроблені зловмисниками. TLS забезпечує автентифікацію ідентичності сервера, гарантуючи, що клієнт зв'язується з потрібним сервером. Він також встановлює безпечний сеанс із симетричними алгоритмами шифрування, що дозволяє клієнту і серверу безпечно обмінюватися даними. Це гарантує, що конфіденційні дані, які передаються мережею, зашифровані та захищені від несанкціонованого доступу.

Також, важливими є надійна автентифікація та контроль доступу, безпечні механізми зберігання, регулярне оновлення безпеки, дотримання правил конфіденційності, моніторинг безпеки та можливості реагування на інциденти, а також програми інформування та навчання співробітників. Задовольняючи ці вимоги, система може захистити конфіденційні дані, запобігти несанкціонованому доступу та зменшити ризики безпеки, щоб забезпечити цілісність і конфіденційність інформаційних активів інтернет-магазину.

1.5.3. Вимоги до складу та параметрів технічних засобів

Сервер повинен мати достатню обчислювальну потужність для ефективного виконання завдань системи. Багатоядерний процесор може забезпечити необхідні обчислювальні можливості. Для обробки та зберігання даних потрібен достатній обсяг пам'яті, який вимірюється в гігабайтах або терабайтах. Для зберігання вилучених даних підходять твердотілі накопичувачі (SSD) або жорсткі диски великої ємності (HDD).

Швидке та надійне підключення до Інтернету необхідне для доступу та вилучення даних з різних онлайн-джерел. Стабільне інтернет-з'єднання необхідне для доступу до цільових веб-сайтів, завантаження веб-сторінок і отримання необхідних даних. Високошвидкісне широкосмугове з'єднання, бажано з низькою затримкою, забезпечує своєчасне отримання даних і скорочує час вилучення.

Для зберігання вилучених даних важлива ємність сховища. Точні вимоги до сховища залежать від очікуваного обсягу даних і періоду зберігання. Наприклад, можуть бути використані масштабовані рішення для зберігання даних, такі як мережеві сховища (NAS) або хмарні сервіси, які можуть задовольнити зростаючі потреби інтернет-магазину в даних. Заходи резервування, такі як конфігурації RAID або реплікація даних, допомагають запобігти втраті даних у разі апаратних збоїв.

1.5.4. Вимоги до інформаційної та програмної сумісності

Програмне забезпечення має бути розроблене мовою програмування Clojure. Clojure працює на віртуальній машині Java (JVM), тому необхідно забезпечити сумісність з версією Java, визначеною Clojure. Це включає сумісність з версією Java Development Kit (JDK), версією JVM та будь-якими специфічними функціями чи бібліотеками Java, що використовуються Clojure або самою програмою.

В якості інструменту збірки та системи управління залежностями використовуються інструменти Clojure CLI, які мають бути встановлені для їх використання.

Для передавання даних між програмними компонентами, реалізованими на Clojure використовується формат даних EDN, для зберігання і передавання даних у зовнішнє середовище використовується формат JSON. Необхідно бути впевненим, що доступні необхідні бібліотеки або функції для розбору, обробки та генерації даних у цих форматах.

Програма повинна бути сумісною з цільовими платформами, на яких вона буде розгорнута, такими як різні операційні системи (Linux, macOS, Windows) та апаратні архітектури (x86, ARM). Це передбачає врахування специфічних для платформи залежностей, бібліотек та API, на які покладається програма.

РОЗДІЛ 2.
ПРОЕКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Функціональне призначення інформаційної системи

Функціональне призначення системи полягає у вилученні релевантних даних, таких як ціни на товари, описи, специфікації, наявність та іншу відповідну інформацію з різних джерел конкурентів.

З точки зору замовника, така система має кілька переваг. По-перше, вона дозволяє інтернет-магазину пропонувати конкурентні ціни. Відстежуючи ціни конкурентів, магазин може скоригувати власну цінову стратегію, щоб забезпечити конкурентоспроможні ціни для клієнтів. Це дає покупцям перевагу доступу до товарів за конкурентними цінами, що потенційно заощаджує їхні гроші. По-друге, система допомагає інтернет-магазину підтримувати точну та актуальну інформацію про товари. Отримуючи інформацію про товари з веб-сайтів конкурентів, магазин може переконатися, що його власний каталог товарів є вичерпним, повним і відображає найсвіжішу інформацію. Це покращує досвід покупців, надаючи їм точні та достовірні дані про товари.

2.2 Опис застосованих математичних методів

Під час розроблення програмного забезпечення в межах кваліфікаційної роботи математичні методи не використовуються.

2.3 Опис використаних технологій та мов програмування

Clojure - це сучасна мова програмування з динамічною типізацією, функціональна і призначена для роботи на віртуальній машині Java (JVM). Це діалект мови Lisp, яка є сімейством мов програмування, відомих своїм унікальним синтаксисом та потужними абстракціями [13].

Clojure та функціональне програмування загалом добре підходять для створення систем подібного роду з кількох ключових причин [14]:

- Функціональне програмування сприяє незмінності, що означає, що дані не можуть бути змінені після створення. У контексті веб-скрепінгу незмінність гарантує, що вилучені дані залишаються послідовними і незмінними протягом усього процесу вилучення. Це допомагає уникнути пошкодження даних і полегшує міркування про стан даних.

- Вилучення даних з Інтернету часто передбачає вилучення та перетворення даних з різних джерел. Функціональне програмування надає потужні інструменти для роботи з даними, такі як функції вищого порядку та незмінні структури даних. Це дозволяє легко та ефективно виконувати операції з перетворення даних, такі як фільтрація, картування та зменшення, що спрощує обробку вилучених даних.

- Багато завдань веб-скрепінгу передбачають отримання даних з декількох джерел одночасно. Функціональні мови програмування, такі як Clojure, надають вбудовану підтримку паралелізму. Незмінні структури даних і чисті функції Clojure полегшують роботу з паралельними завданнями, не турбуючись про спільні змінні стани або умови перегонів. Це дозволяє створювати ефективні та масштабовані операції веб-скрепінгу.

- Функціональне програмування заохочує використання чистих функцій і компонування. Системи веб-скрепінгу часто потребують модульних і багаторазових компонентів для обробки різних аспектів процесу скрапінгу, таких як отримання веб-сторінок, синтаксичний аналіз HTML і вилучення даних. Функціональне програмування дозволяє створювати складні функції, які можна легко комбінувати і повторно використовувати, що призводить до більш зручного для підтримки і модульного коду.

- Вилучення веб-даних може бути пов'язане з великою кількістю помилок через мінливість і непередбачуваність веб-джерел даних. Функціональне програмування надає потужні механізми обробки помилок, такі як незмінність і

монади, які можна використовувати для контрольованої обробки і поширення помилок. Це допомагає створювати надійні та відмовостійкі веб-скрепери.

- Функціональні мови програмування, включаючи Clojure, часто мають стислий і виразний синтаксис. Це дозволяє розробникам писати зрозумілий і читабельний код, який відображає суть логіки веб-скрепінгу. Зосередженість функціонального програмування на чистих функціях і перетвореннях даних призводить до створення коду, який легше розуміти, підтримувати і налагоджувати.

Jsoup - це бібліотека Java, яка надає зручний спосіб вилучення та маніпулювання даними з HTML-документів. Вона зазвичай використовується для веб-скрепінгу та синтаксичного аналізу HTML-контенту в додатках на Java. Jsoup надає простий та інтуїтивно зрозумілий API, який дозволяє розробникам переміщатися та маніпулювати елементами HTML, витягувати дані та виконувати різні операції на веб-сторінках [15].

Clojure, будучи мовою, яка працює на віртуальній машині Java (JVM), може легко інтегруватися з бібліотеками Java, такими як Jsoup. Це дозволяє Clojure розробникам використовувати функціональність Jsoup у своїх Clojure проектах. Clojure надає потужний механізм взаємодії під назвою Java Interop, який дозволяє викликати методи Java та отримувати доступ до класів Java безпосередньо з коду Clojure [16].

2.4 Опис структури інформаційної системи та алгоритмів її функціонування

Система, реалізована у вигляді бібліотеки, якою користуються переважно програмісти, орієнтована на отримання інформації про вартість продукції та деталізацію товарів на основі заданого SKU. Структура системи базується на наданні зручного API для легкої інтеграції в існуючі системи інтернет-магазину.

Компонент збору даних відповідає за веб-скрепінг і збір даних з веб-сайтів конкурентів. Це передбачає створення HTTP-запитів, обробку автентифікації та

розбір HTML-відповідей для вилучення необхідної інформації. Цей компонент використовує алгоритми для навігації по веб-сторінках, пошуку релевантної інформації та вилучення необхідних даних. Система використовує такі методи, як синтаксичний аналіз HTML та використання селекторів CSS, щоб ідентифікувати та витягувати певні елементи з веб-сторінок.

За допомогою компонента збереження даних, зібрані дані можуть бути збережені в структурований формат обміну даними JSON для їх організації та подальшого аналізу.

2.5 Обґрунтування та організація вхідних та вихідних даних програми

Дані передаються та отримуються через API програми за допомогою виклику певних функцій.

Система потребує URL-адреси або доменні імена веб-сайтів конкурентів як вхідні дані. Це дозволяє програмі орієнтуватися на конкретні джерела для вилучення даних. Для отримання інформації про конкретні товари з веб-сайтів конкурентів програмі потрібен SKU. SKU слугують унікальними ідентифікаторами продуктів і допомагають звузити процес вилучення даних.

Надання URL-адрес веб-сайтів конкурентів та SKU товарів у якості вхідних даних дозволяє програмі ефективно отримувати цільову інформацію. Вказавши конкурентів і продукти, що цікавлять, система може зосередитися на вилученні релевантних даних, зменшуючи непотрібну обробку і покращуючи загальну продуктивність.

Програма повертає детальну інформацію про товари, отриману з веб-сайтів конкурентів. Вона може включати такі атрибути, як назва продукту, опис, ціна, наявність та інші відповідні характеристики. Залежно від типу запитуваної інформації, кінцевий структурований формат даних відрізняється. наприклад, якщо користувач хоче отримати інформацію про вартість товару в різних інтернет-магазинах, знаючи SKU товару, на виході він отримає хеш-таблицю, в якій в якості ключів буде URL на сам товар у певному магазині, а в якості значень

- отримана вартість товару. В іншому прикладі, якщо користувач запитає характеристики товару в певному магазині, то вони також будуть повернуті у вигляді хеш-таблиці, але в якості ключів і значень будуть відповідні значення характеристик товару.

Для персистентності і подальшого аналізу вихідні дані можуть бути збережені у форматі JSON. JSON - це простий формат обміну даними, який широко використовується у веб-додатках. Він забезпечує простий і зрозумілий людині синтаксис для структурування даних у парах ключ-значення. JSON розроблений таким чином, щоб його було легко розуміти і писати, а також легко аналізувати і генерувати машинами. Він базується на підмножині мови програмування JavaScript, але його можна використовувати з будь-якою мовою програмування. JSON підтримує різні типи даних, включаючи рядки, числа, булеві функції, масиви та об'єкти, що дозволяє створювати гнучкі та ієрархічні структури даних. Він став стандартом де-факто для обміну даними у веб-аплікаторах і широко підтримується мовами програмування та фреймворками. [17]. Цей формат дозволить забезпечити легку інтеграцію з існуючими системами та полегшити аналіз даних.

2.6 Опис роботи інформаційної системи

2.6.1 Використані технічні засоби

Для забезпечення безперебійної роботи системи слід враховувати певні мінімальні вимоги до апаратного забезпечення.

Апаратне забезпечення системи повинно включати потужний процесор з декількома ядрами і пристойною тактовою частотою для ефективного виконання завдань.

Достатня кількість оперативної пам'яті (RAM) має вирішальне значення для обробки великих обсягів даних і паралельних завдань. Рекомендується мати мінімум 4 ГБ оперативної пам'яті, але буде корисним мати більше пам'яті.

Для зберігання вилучених даних потрібен достатній обсяг пам'яті. Твердотільні накопичувачі (SSD) є кращими для швидшого читання/запису та покращеної загальної продуктивності. Обсяг необхідного сховища залежить від очікуваного обсягу даних.

Надійне і високошвидкісне підключення до Інтернету має важливе значення для доступу і вилучення даних з веб-сайтів.

Треба переконатися, що мережеве з'єднання має стабільну і достатню пропускну здатність для ефективної обробки даних.

У деяких випадках можуть знадобитися проксі-сервіси, щоб впоратися з блокуванням IP-адрес або зберегти анонімність під час вилучення даних. Треба переконатися, що апаратне забезпечення може інтегрувати проксі-сервери та керувати ними, якщо це необхідно.

2.6.2 Використані програмні засоби

Розроблена система побудована за допомогою мови програмування Clojure і працює на JVM, використовуючи низку програмних інструментів для ефективного виконання своїх завдань. Ці інструменти включають операційну систему, середовище розробки та різноманітні бібліотеки.

По-перше, система покладається на віртуальну машину Java (JVM), оскільки Clojure - це мова на основі JVM. Це означає, що JVM повинна бути встановлена в системі, де буде працювати система. Система може бути розгорнута на різних операційних системах, які підтримує JVM. Наприклад, таких як Linux або Windows, залежно від уподобань та інфраструктури хостинг-сервера.

Окрім JVM, система покладається на екосистему Clojure, включаючи компілятор Clojure та стандартну бібліотеку Clojure. Вони надають фундаментальні мовні можливості та утиліти, необхідні для розробки та виконання системи.

Також, одним з ключових інструментів є Clojure CLI, який надає інструменти управління залежностями, специфічним для проектів Clojure. Він допомагає керувати залежностями проекту, вказуючи їх у централізованому конфігураційному файлі. Clojure CLI гарантує, що бібліотеки та пакети, необхідні для роботи системи, належним чином управляються та включаються до проекту.

Для полегшення функціональності веб-скрепінгу система використовує бібліотеку Jsoup, яка є популярною бібліотекою розбору та маніпулювання HTML в екосистемі Java та Clojure.

Ці програмні інструменти разом утворюють технічний стек для роботи системи. Вони дозволяють системі функціонувати в середовищі JVM, використовувати можливості мови Clojure та спеціалізовані бібліотеки, такі як Jsoup, для ефективного виконання операцій системи.

2.6.3 Виклик та завантаження програми

Виклик і завантаження програми, реалізованої у вигляді бібліотеки, передбачає певні кроки для її інтеграції в існуючу систему та використання її функціональних можливостей у кодовій базі.

Першим кроком є імпорт бібліотеки до програми. Для цього потрібно вказати бібліотеку як залежність у конфігураційному файлі програми, наприклад, у файлі project.clj для Leiningen або у файлі deps.edn для інструментів Clojure CLI. Після цього інструмент збірки завантажить бібліотеку і зробить її доступною для використання.

Після того, як бібліотеку імпортовано, її потрібно ініціалізувати у програмі. Це робиться за допомогою завантаження відповідного простору імен, який містить функції та ресурси бібліотеки. У Clojure для цього використовуються форми require та use, де вказуються простори імен, які потрібно завантажити.

Після ініціалізації бібліотеки її функції можна викликати у програмі. Це передбачає виклик функцій, наданих бібліотекою, передачу необхідних

аргументів і перехоплення значень, що повертаються. Програма може використовувати ці функції для виконання певних завдань.

2.6.4 Опис інтерфейсу користувача

Оскільки систему реалізовано у вигляді бібліотеки мови програмування, вона надає користувачеві програмний інтерфейс, з яким ми можемо взаємодіяти за допомогою виклику функцій, що співвідносяться з функціоналом інформаційної системи.

Для подальшої демонстрації програмного інтерфейсу системи ми будемо використовувати середовище розроблення, яке буде організовано таким чином, що у верхньому вікні будуть зображені викликані функції, а в нижньому - результат виклику цих функцій.

Перш за все, для того, щоб отримати інформацію про певний товар, можна скористатися URL-адресою на цей товар у певному інтернет-магазині (рис 2.1). Інформація про товар містить SKU товару, його назву, вартість та опис.

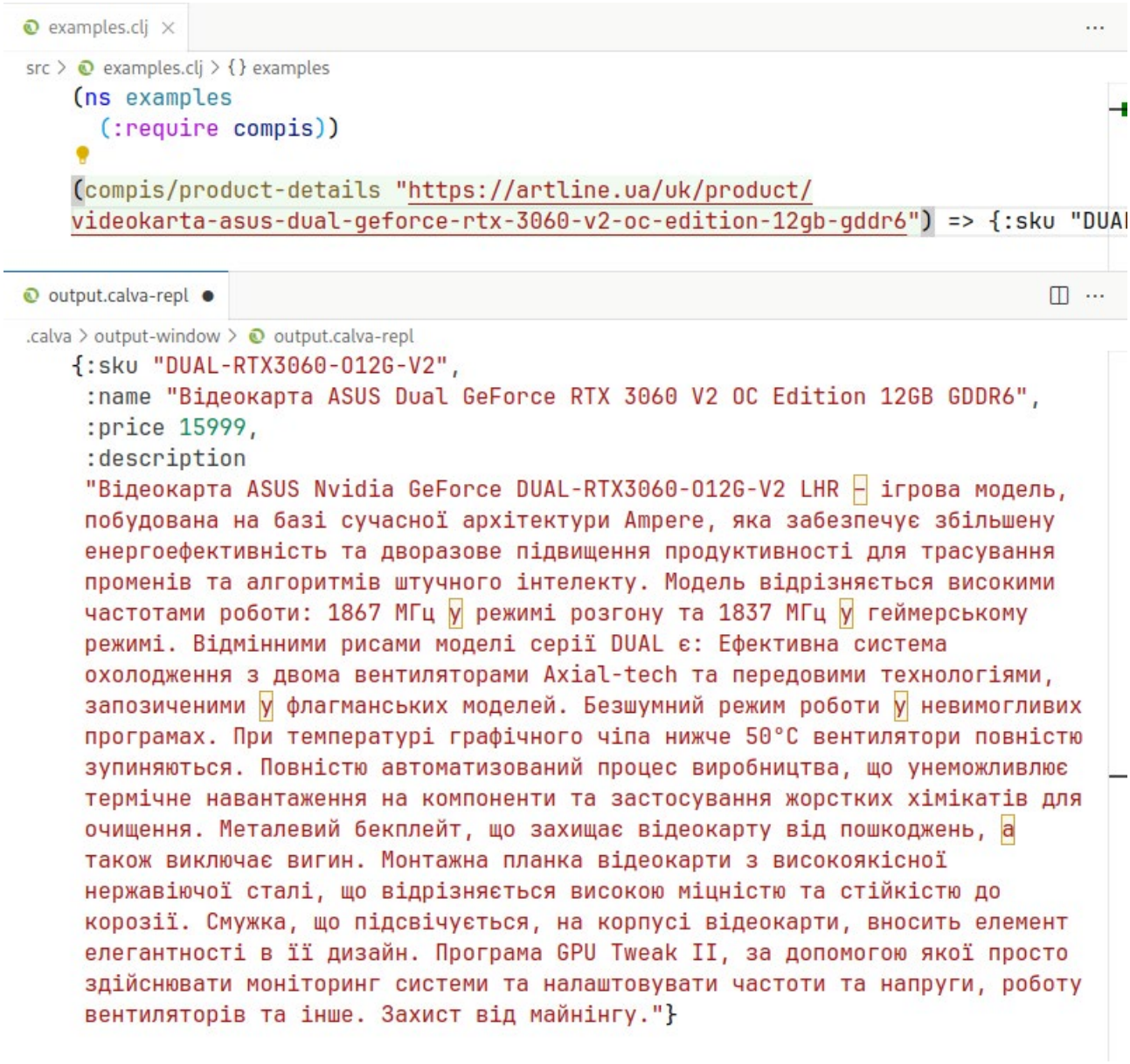

Рис. 2.1 Робимо виклик функції product-details передавши URL-адресу на товар в магазині в якості аргументу і в результаті отримаємо інформацію про товар у форматі EDN, який Clojure використовує внутрішньо для представлення даних

Якщо ми не знаємо URL товару, але нам відомий SKU товару, то ми можемо скористатися ним для отримання URL на товар у певному магазині. Для цього нам достатньо вказати магазин, у якому ми хочемо отримати URL на товар і сам SKU (рис 2.2).

Рис. 2.2 Робимо виклик функції product-url передавши магазин і SKU товару в якості аргументів і в результаті отримаємо рядок, що містить URL на товар у магазині.

Також, ми можемо отримати URL-адресу на товар у всіх підтримуваних інтернет-магазинах одразу вказавши лише один SKU (рис 2.3).

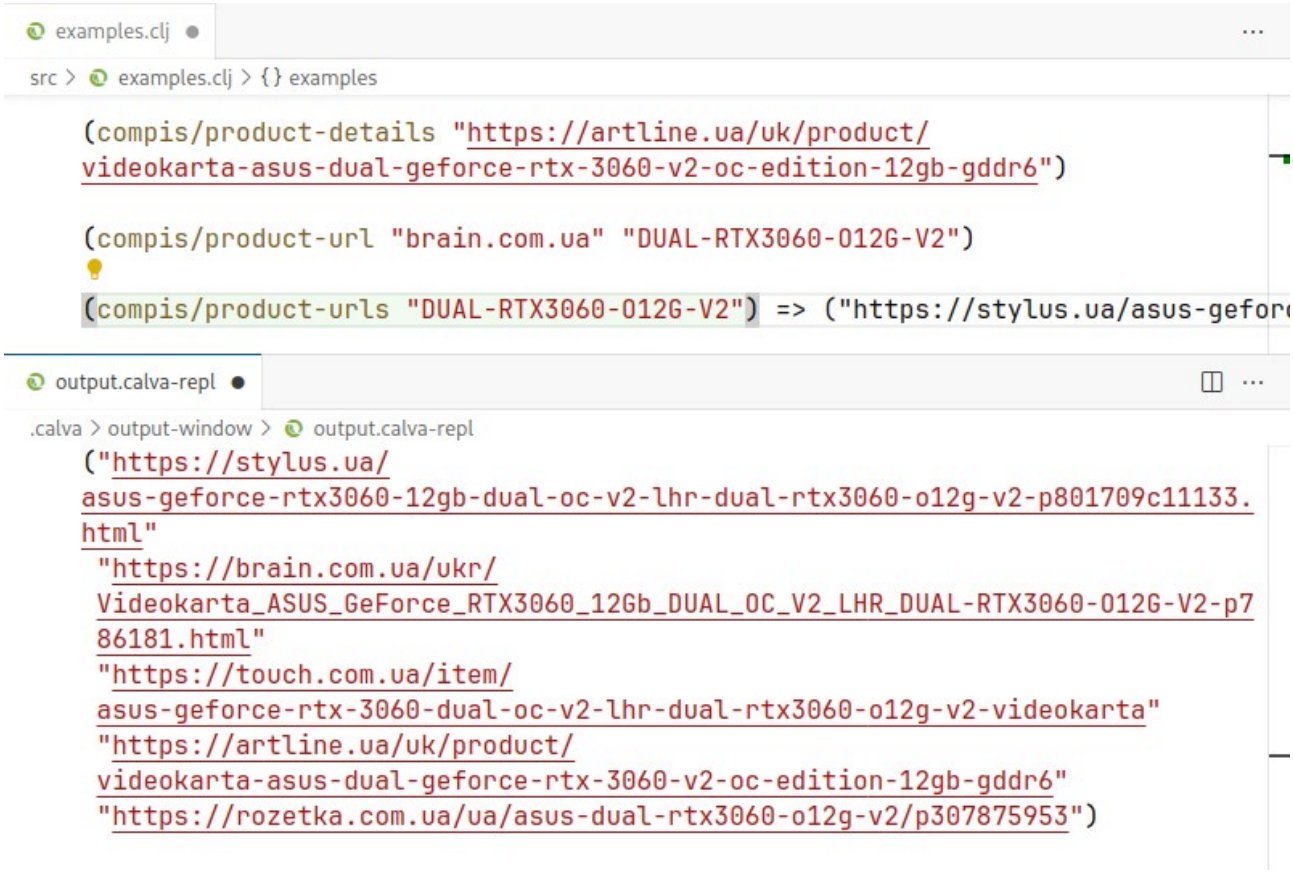

Рис. 2.3 Робимо виклик функції product-urls передавши SKU товару в якості аргументу і в результаті отримаємо послідовність рядків, які являють собою URL-адреси на товар у підтримуваних інтернет-магазинах

Це може бути корисним, якщо ми хочемо отримати інформацію про певний товар одразу у всіх магазинах, які підтримуються нашою системою. Наприклад, нам може бути цікава вартість товару в усіх підтримуваних інтернет-магазинах для моніторингу цін. Для цього реалізовано більш високорівневу функцію, що використовує у своїй реалзіації попередню функцію і дає змогу отримати вартість на певний товар у всіх підтримуваних інтернет-магазинах одразу (Рис 2.4).

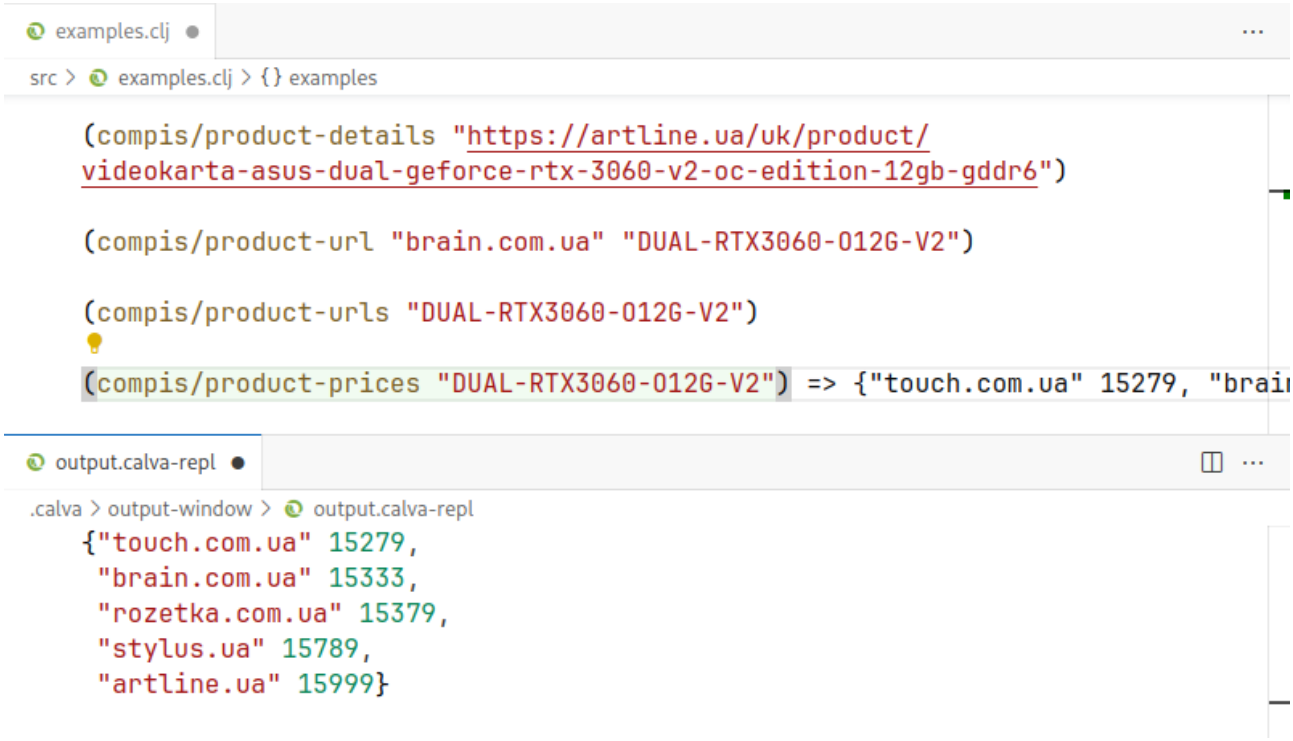

Рис. 2.4 Робимо виклик функції product-prices передавши SKU товару в якості аргументу і в результаті отримаємо структуру даних, яка зберігає пари ключ-значення, де в якості ключів виступають магазини, а в якості значень - отримана вартість товару у відповідних магазинах

Також, система дозволяє одразу отримати найкращу вартість серед підтримуваних магазинів (Рис 2.5).

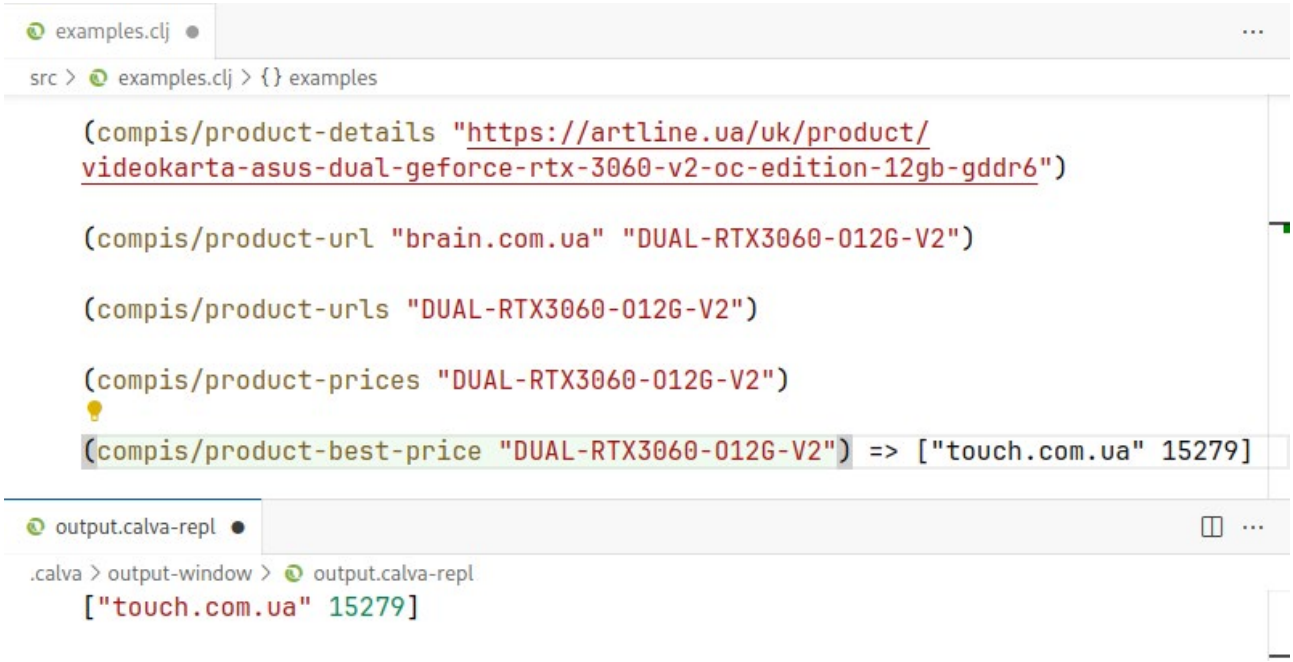

Рис. 2.5 Робимо виклик функції product-best-price передавши SKU товару в якості аргументу і в результаті отримаємо масив із двома елементами, включно з інтернет-магазином і вартістю товару

Система дає змогу зберігати зміни цін на товари у форматі зберігання даних JSON (Рис 2.6, 2.7, 2.8). За замовчуванням, дані будуть збережені у файл prices.json у директорії resources щодо кореневої директорії проєкту, але ми, також, можемо вказати кінцевий файл, у який будуть збережені дані, додавши опціональний другий аргумент. Формат, у якому зберігаються дані, містить використання часових штампів у форматі ISO 8601. ISO 8601 - це міжнародний стандарт для представлення дат, часу та часових інтервалів. Він визначає формат для позначення дати і часу, який широко визнаний і використовується в різних галузях і сферах застосування [18]. У нашому випадку часові штампи відповідають часу, коли було проведено операцію отримання цін.

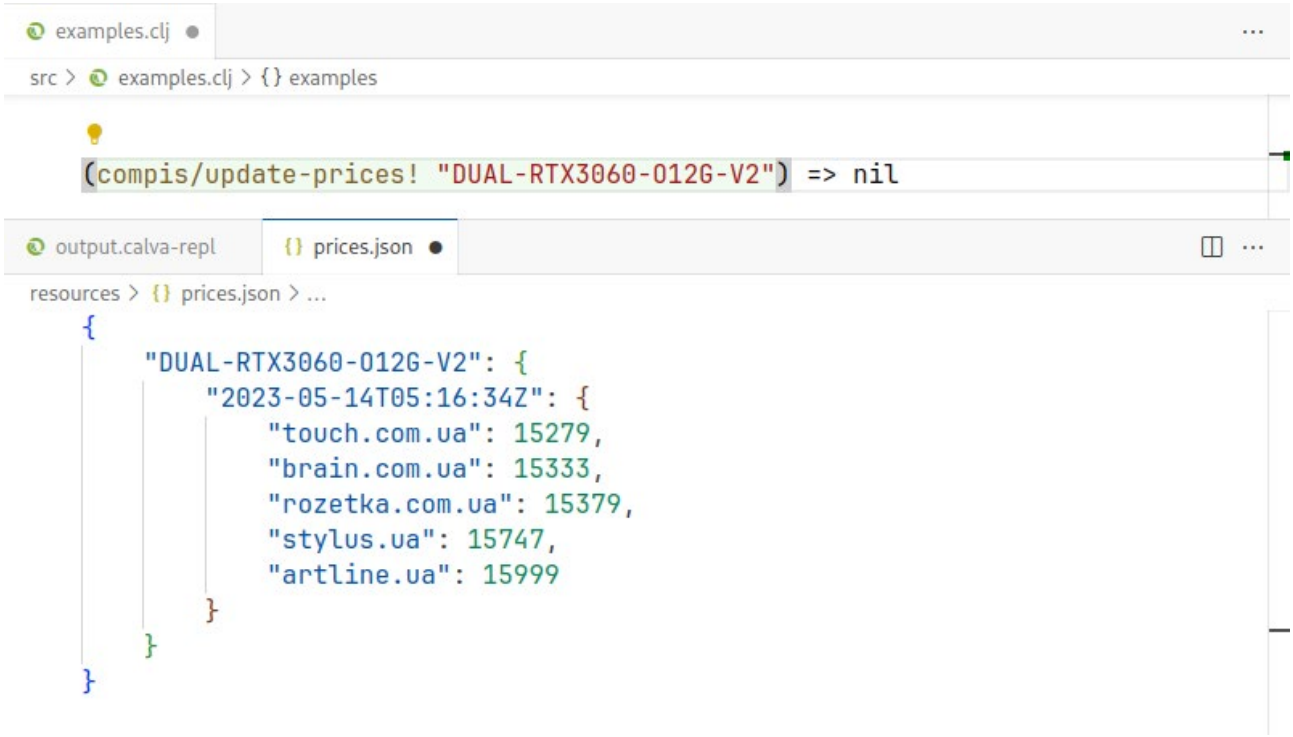

Рис. 2.6 Робимо виклик функції update-prices! передавши SKU товару в якості аргументу і в результаті зберігаємо інформацію про вартість товару в різних інтернет-магазинах у певний момент за часу у файл prices.json

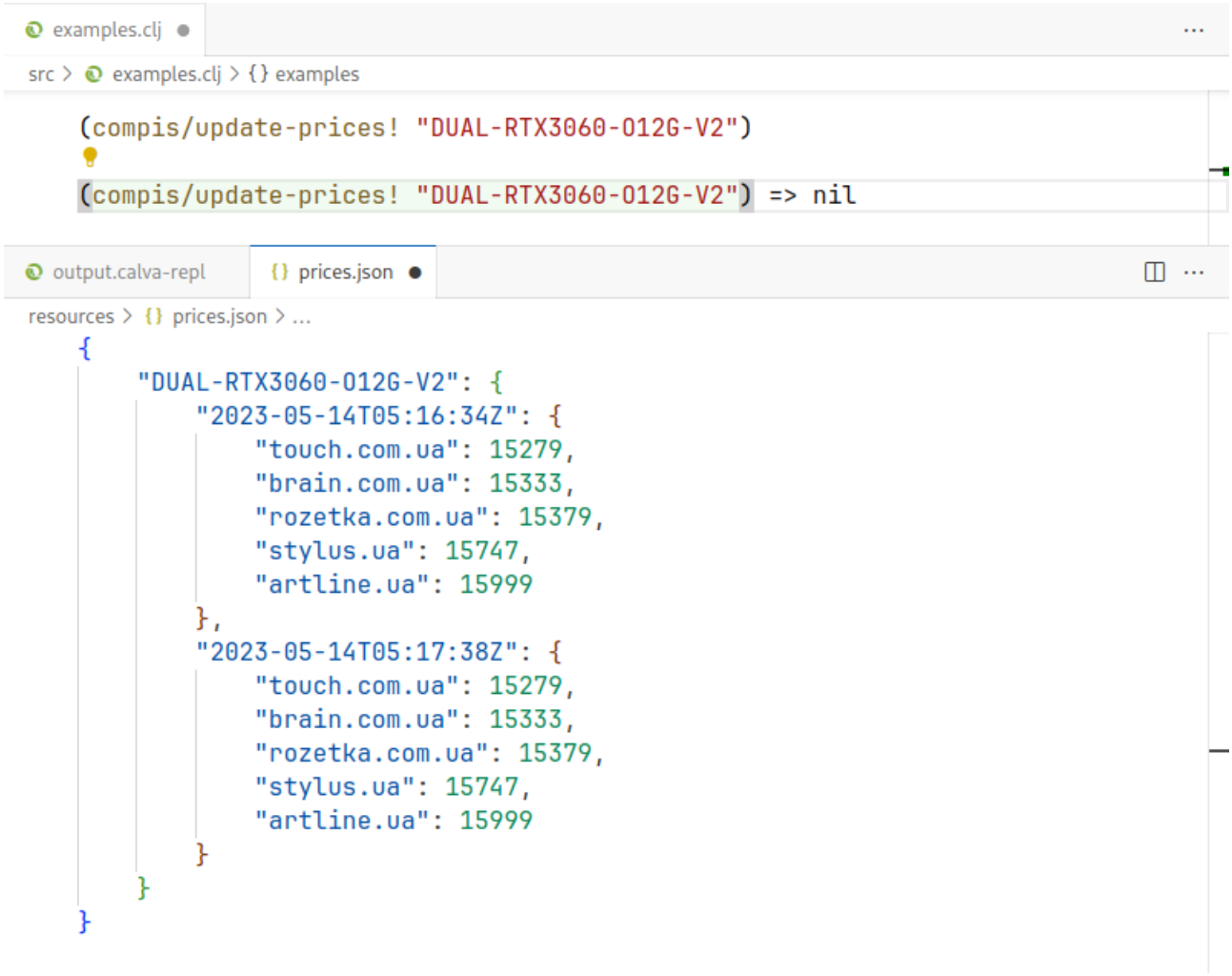

Рис. 2.7 Якщо ми викличемо попередню функцію ще раз, дані про вартість товару будуть збережені під новим тимчасовим штампом, що відповідає часу після попереднього

examples.clj

src > examples.clj > {} examples

```
(compis/update-prices! "DUAL-RTX3060-O12G-V2")

(compis/update-prices! "DUAL-RTX3060-O12G-V2")

(compis/update-prices! "GV-N4070WF3OC-12GD") => nil
```

output.calva-repl | prices.json

resources > {} prices.json > ...


```
{
    "DUAL-RTX3060-O12G-V2": {
        "2023-05-14T05:16:34Z": {
            "touch.com.ua": 15279,
            "brain.com.ua": 15333,
            "rozetka.com.ua": 15379,
            "stylus.ua": 15747,
            "artline.ua": 15999
        },
        "2023-05-14T05:17:38Z": {
            "touch.com.ua": 15279,
            "brain.com.ua": 15333,
            "rozetka.com.ua": 15379,
            "stylus.ua": 15747,
            "artline.ua": 15999
        }
    },
    "GV-N4070WF3OC-12GD": {
        "2023-06-08T16:05:05Z":{
            "touch.com.ua": 28559,
            "stylus.ua": 28815,
            "brain.com.ua": 29999,
            "artline.ua": 29999,
            "hard.rozetka.com.ua": 29999
        }
    }
}
```


Рис. 2.8 Для товару з іншим SKU нові дані будуть збережені під відповідним SKU товару

Для магазинів, які підтримують цей функціонал, ми можемо отримати технічні характеристики товару (Рис 2.9).

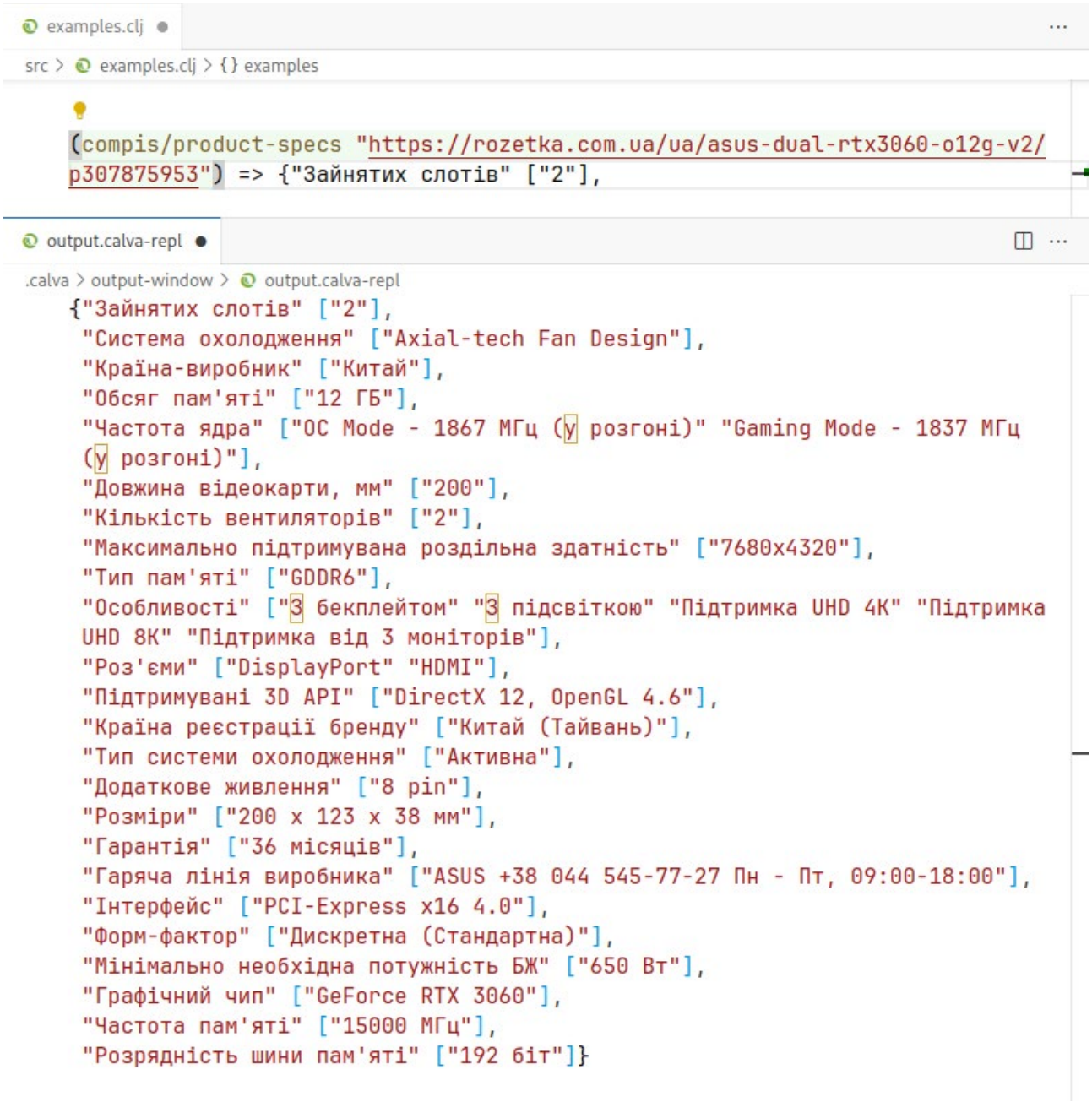

Рис. 2.9 Робимо виклик функції product-specs передавши URL-адресу на товар в магазині в якості аргументу і в результаті отримаємо хеш-таблицю, в якій в якості ключів і значень будуть відповідні значення характеристик товару. Оскільки під одним ключем може зберігатися більше, ніж одне значення, то значення представлені у вигляді масиву

Також, для магазинів, які підтримують відповідний функціонал, є можливість отримання коментарів товару (Рис 2.10).

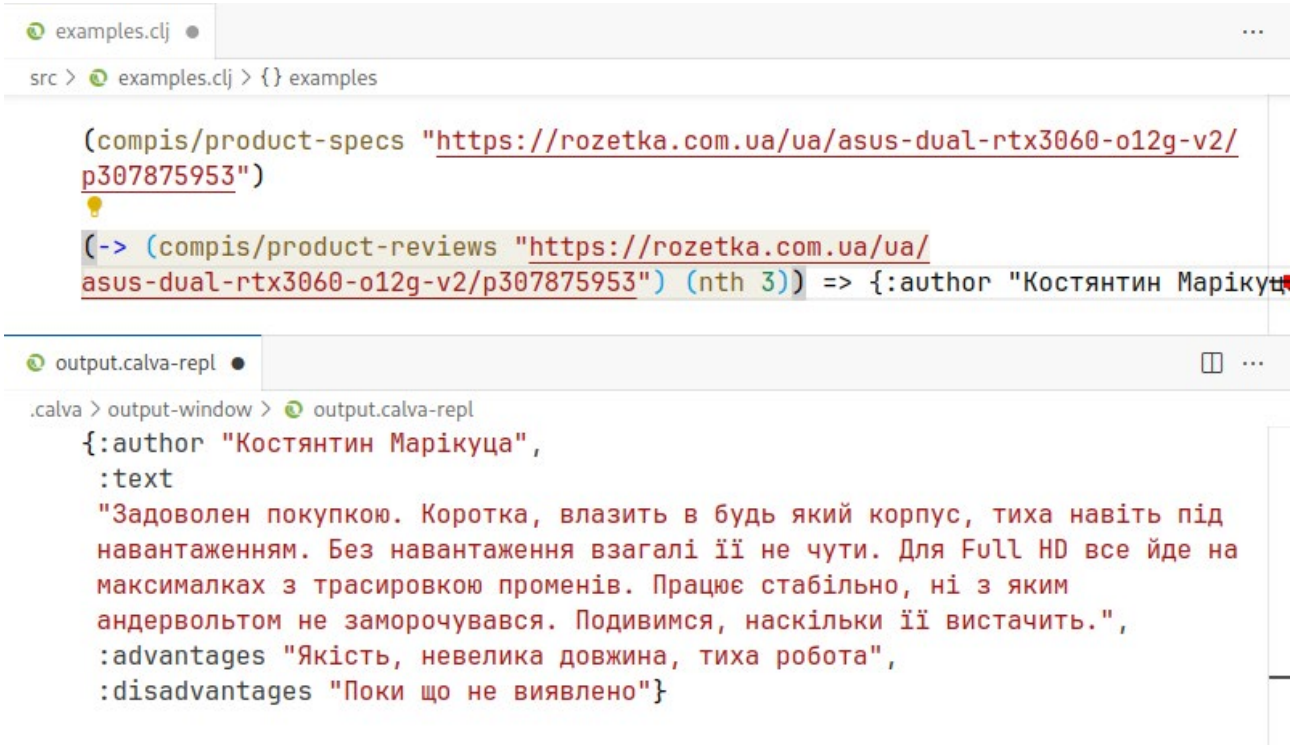

Рис. 2.10 Робимо виклик функції product-reviews передавши URL-адресу на товар в магазині в якості аргументу і в результаті отримуємо масив, що складається з хеш-таблиць, які представляють коментарі. Для демонстрації результату виведено один із коментарів

РОЗДІЛ 3.
ЕКОНОМІЧНИЙ РОЗДІЛ

Вихідні дані розробки програмного забезпечення:

1) передбачуване число операторів – 400
2) коефіцієнт складності програми – 1,1
3) коефіцієнт кореляції програми в ході її розробки - 0,1
4) середня годинна заробітна плата програміста, грн/год - 50
5) вартість машино-години ЕОМ, грн/год – 10

3.1 Розрахунок трудомісткості та вартості розробки інформаційної системи

Нормування праці в процесі створення ПЗ істотно ускладнено в силу творчого характеру праці програміста. Тому трудомісткість розробки ПЗ може бути розрахована на основі системи моделей з різною точністю оцінки.

Трудомісткість розробки ПЗ можна розрахувати за формулою:

$$t = t_o + t_u + t_a + t_n + t_{отл} + t_д \text{, людино-годин,} \qquad (3.1)$$

де t_o – витрати праці на підготовку й опис поставленої задачі (приймається 20);

t_u – витрати праці на дослідження алгоритму рішення задачі;

t_a – витрати праці на розробку блок-схеми алгоритму;

t_n – витрати праці на програмування по готовій блок-схемі;

$t_{отл}$ – витрати праці на налагодження програми на ЕОМ;

$t_д$ – витрати праці на підготовку документації.

Складові витрати праці визначаються через умовне число операторів у ПЗ, яке розробляється.

Умовне число операторів (підпрограм):

$$Q = q \cdot \mathrm{C}(1 + p), \text{ де} \qquad (3.2)$$

q– передбачуване число операторів;

C– коефіцієнт складності програми;

p – коефіцієнт кореляції програми в ході її розробки.

$$Q = 400 \cdot 1{,}1 \cdot (1 + 0{,}1) = 440;$$

Витрати праці на вивчення опису задачі t_u визначається з урахуванням уточнення опису і кваліфікації програміста:

$$t_u = \frac{Q \cdot B}{(75 \ldots 85)K}, \text{ людино-годин,} \qquad (3.3)$$

де B – коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі, В=1.2 ... 1.5;

K – коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності. До 2 – 0,8;

$$t_u = \frac{440 \cdot 1{,}2}{85 \cdot 1{,}2} = 5{,}1, \text{ людино-годин.}$$

Витрати праці на розробку алгоритму рішення задачі:

$$t_a = \frac{Q}{(20 \ldots 25)K} \quad (3.4)$$

$t_a = \frac{440}{20 \cdot 1,2} = 18,3,$ людино-годин.

Витрати на складання програми по готовій блок-схемі:

$$t_n = \frac{Q}{(20 \ldots 25)K} \quad (3.5)$$

$t_n = \frac{440}{25 \cdot 1,2} = 14,6,$ людино-годин.

Витрати праці на налагодження програми на ЕОМ:

- за умови автономного налагодження одного завдання:

$$t_{\text{отл}} = \frac{Q}{(4 \ldots 5)K} \quad (3.6)$$

$t_{\text{отл}} = \frac{440}{5 \cdot 1,2} = 73,3,$ людино-годин,

- за умови комплексного налагодження завдання:

$$t_{\text{отл}}^{\text{к}} = 1,2 \cdot t_{\text{отл}} \quad (3.7)$$

$t_{\text{отл}}^{\text{к}} = 1,2 \cdot 73,3 = 87,9,$ людино-годин

Витрати праці на підготовку документації:

$$t_{\text{д}} = t_{\text{др}} + t_{\text{до}} \tag{3.8}$$

де $t_{др}$ – трудомісткість підготовки матеріалів і рукопису

$$t_{\text{др}} = \frac{Q}{(15 \dots 20)K} \tag{3.9}$$

$t_{\text{др}} = \frac{440}{20 \cdot 1.2} = 18{,}3$, людино-годин.

$t_{до}$ – трудомісткість редагування, печатки й оформлення документації

$$t_{\text{др}} = 0{,}75 \cdot t_{\text{др}} \tag{3.10}$$

$t_{\text{др}} = 0{,}75 \cdot 18{,}3 = 13{,}7$, людино-годин.

$t_{\text{д}} = 18{,}3 + 13{,}7 = 32$, людино-годин.

Отримаємо трудомісткість розробки програмного забезпечення:

t = 20+5,1+18,3+14,6+73,3+32 = 168,3 людино-годин.

У результаті ми розрахували, що в загальній складності необхідно 168,3 людино-годин для розробки даного програмного забезпечення.

3.2. Розрахунок витрат на створення програми

Витрати на створення ПЗ $K_{по}$ включають витрати на заробітну плату виконавця програми $З_{зп}$ і витрат машинного часу, необхідного на налагодження програми на ЕОМ.

$$К_{по} = З_{зп} + З_{мв}, \text{ грн}, \quad (3.11)$$

де $З_{ЗП}$ – заробітна плата виконавців, яка визначається за формулою:

$$З_{зп} = t \cdot С_{пр}, \text{ грн}, \quad (3.12)$$

де t – загальна трудомісткість, людино-годин;

$С_{ПР}$ – середня годинна заробітна плата програміста, грн/година

$$З_{ЗП} = 168{,}3 \cdot 50 = 8415, \text{ грн.}$$

$З_{МВ}$– Вартість машинного часу, необхідного для налагодження програми на ЕОМ:

$$З_{мв} = t_{отл} \cdot с_{м}, \text{ грн}, \quad (3.13)$$

де $t_{отл}$ – трудомісткість налагодження програми на ЕОМ, год.

$С_{МЧ}$ – вартість машино-години ЕОМ, грн/год.

$$З_{МВ} = 73{,}3 \cdot 10 = 733, \text{ грн.}$$

$$К_{ПО} = 8415 + 733 = 9148, \text{ грн.}$$

Очікуваний період створення ПЗ:

$$T = \frac{t}{B_k \cdot F_p}, \text{ мес.} \quad (3.14)$$

де B_k - число виконавців;

F_p - місячний фонд робочого часу (при 40 годинному робочому тижні Fp=176 годин).

$$T = \frac{168,3}{1 \cdot 176} = 0,9 \text{ міс.}$$

Висновки. Час розробки даного програмного забезпечення складає 168,3 людино-годин. Таким чином, очікувана тривалість розробки складе 0,9 місяця при 40 годинному робочому тижні (місячний фонд робочого часу 176 годин), а витрати на створення програмного забезпечення складатимуть 9148 грн.

ВИСНОВКИ

Розроблена система стала значним кроком на шляху до надання цінних інсайтів для інтернет-магазину. Її інтеграційні можливості, дотримання стандартів інформаційної безпеки та ефективне використання таких технологій, як Clojure та Jsoup, сприяють її успішному впровадженню. Система надає онлайн-бізнесу точні дані, що дозволяє приймати обґрунтовані рішення та зберігати конкурентну перевагу на ринку.

Однією з помітних переваг системи є її універсальність і сумісність з існуючими системами інтернет-магазинів. Реалізована у вигляді бібліотеки, вона легко інтегрується в робочий процес програміста і може бути легко включена в інфраструктуру інтернет-магазину. Це дозволяє оптимізувати оновлення каталогу та надає маркетологам цінні дані для аналізу та прийняття рішень.

Якщо порівнювати систему з існуючими альтернативами, то вона має кілька переваг. Використання Clojure як мови програмування забезпечує переваги функціонального програмування, дозволяючи створювати стислий і виразний код. Інтеграція таких бібліотек, як Jsoup, спрощує завдання веб-скрепінгу, дозволяючи ефективно і точно витягувати дані. Крім того, модульна та розширювана архітектура системи забезпечує гнучкість, роблячи її пристосованою до різних бізнес-потреб.

Надалі є кілька ключових напрямків для подальшого розвитку. По-перше, слід визначити пріоритети масштабованості та оптимізації продуктивності, щоб забезпечити здатність системи обробляти більші обсяги даних і збільшувати робочі навантаження. Це може передбачати впровадження методів розподілених обчислень і використання хмарної інфраструктури для ефективної обробки та зберігання даних.

Розширення джерел даних за межі веб-сайтів конкурентів може забезпечити більш повне уявлення про ринок. Інтеграція додаткових джерел, таких як платформи соціальних мереж, оглядові сайти або галузеві бази даних, може збагатити дані та підвищити точність інсайтів, що генеруються системою.

Покращення можливостей аналізу даних за допомогою передових методів, таких як машинне навчання та статистичне моделювання, може розкрити глибші уявлення та можливості прогнозування. Це може допомогти бізнесу ефективніше визначати нові тенденції, вподобання клієнтів та конкурентні стратегії.

Зручний інтерфейс має важливе значення для забезпечення легкого доступу та інтерпретації зібраних даних. Розробка інтуїтивно зрозумілих інформаційних панелей, звітів, що налаштовуються, та інтерактивних інструментів візуалізації може дати користувачам можливість досліджувати та витягувати значущу інформацію з системи, не вимагаючи при цьому високих технічних навичок.

Нарешті, створення механізмів постійного моніторингу та оновлень має вирішальне значення для того, щоб система відповідала динаміці ринку, яка постійно змінюється. Отримання даних у режимі реального часу, автоматичні сповіщення та регулярне обслуговування забезпечують точність та актуальність інформації, що надається системою.

Звертаючись до цих напрямків розвитку, система веб-скрейпінгу може продовжувати розвиватися і надавати цінну інформацію для інтернет-магазину, дозволяючи приймати рішення на основі даних, залишатися конкурентоспроможними та адаптуватися до постійно мінливого ринкового ландшафту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Liu B. Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data. Springer Berlin, 2011

2. Provost, F., Fawcett, T. Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking. O'Reilly Media, 2013

3. What is Web Scraping and How Can It Benefit Your Business? URL: https://dataforest.ai/blog/what-is-web-scraping-and-how-can-it-benefit-your-business (дата звернення: 14.05.2023)

4. Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., Berners-Lee T. Hypertext Transfer Protocol -- HTTP/1.1 : RFC 2616. Internet Engineering Task Force (IETF), 1999. URL: https://www.rfc-editor.org/rfc/rfc2616

5. Berners-Lee T., Fielding, R., Masinter L. Uniform Resource Identifier (URI): Generic Syntax : RFC 3986. Internet Engineering Task Force (IETF), 2005. URL: https://www.rfc-editor.org/rfc/rfc3986

6. HTML Living Standard. URL: https://html.spec.whatwg.org/multipage/ (дата звернення: 14.05.2023)

7. What Is Web Scraping? A Complete Beginner’s Guide. URL: https://careerfoundry.com/en/blog/data-analytics/web-scraping-guide (дата звернення: 14.05.2023)

8. 10 Best Open Source Web Scrapers in 2023. URL: https://www.octoparse.com/blog/10-best-open-source-web-scraper (дата звернення: 14.05.2023)

9. In-Depth Guide to Web Scraping Challenges in 2023. URL: https://research.aimultiple.com/web-scraping-challenges (дата звернення: 14.05.2023)

10. CSS Selectors Level 3. World Wide Web Consortium (W3C), 2018. URL: https://www.w3.org/TR/selectors-3

11. SKU. URL: https://www.britannica.com/technology/SKU (дата звернення: 14.05.2023)

12. Rescorla E. HTTP Over TLS : RFC2818. Internet Engineering Task Force (IETF), 2000. URL: https://www.rfc-editor.org/rfc/rfc2818

13. Emerick C., Carper B., Grand C. Clojure Programming. O'Reilly Media, 2012

14. Miller A., Halloway S., Bedra A. Programming Clojure. The Pragmatic Bookshelf, 2018

15. jsoup: Java HTML parser, built for HTML editing, cleaning, scraping, and XSS safety. URL: https://jsoup.org (дата звернення: 14.05.2023)

16. Clojure — Java Interop. URL: https://clojure.org/reference/java_interop (дата звернення: 14.05.2023)

17. The JavaScript Object Notation (JSON) Data Interchange Format : RFC 8259. Internet Engineering Task Force (IETF), 2017. URL: https://www.rfc-editor.org/rfc/rfc8259

18. Data elements and interchange formats - Information interchange - Representation of dates and times : ISO 8601:2004. International Organization for Standardization (ISO), 2004

19. Методичні рекомендації до виконання кваліфікаційних робіт здобувачів першого рівня вищої освіти спеціальності 122 Комп'ютерні науки/ В.В. Спірінцев, П.О. Іщук, О.С. Шевцова; Д : НТУ «Дніпровська політехніка», 2021. – 59 с.

20. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання: (ГОСТ 7.1-2003, IDT) : ДСТУ ГОСТ 7.1:2006. – Чинний з 2007–07–01. – К. : Держспоживстандарт України, 2007. – 47 с. – (Система стандартів з інформації, бібліотечної та видавничої справи) (Національний стандарт України).

Додаток А

ЛІСТИНГ ПРОГРАМИ

Файл jsoup.clj

```
(ns jsoup
  (:refer-clojure :exclude [get])
  (:import [org.jsoup Jsoup]
           [org.jsoup.nodes Element]
           [org.jsoup.select Elements]))

(defn get [url]
  (.get (Jsoup/connect url)))

(defn parse [html]
  (Jsoup/parse html))

(defn select [node css-selector]
  (.select node css-selector))

(defn select-first [node css-selector]
  (.selectFirst node css-selector))

(defn html [node]
  (.html node))

(defn text [node]
  (.text node))

(defn attr [node a]
  (cond (instance? Element node)
        (.attr node a)
        (instance? Elements node)
        (.eachAttr node a)))
```

Файл core.clj

```
(ns core
  (:require [jsoup :as j]
            [etaoin.api :as e]
            [cheshire.core :as json]
            [clojure.string :as str]
            [clojure.java.io :as io]))

(defn- url->hostname [url]
  (-> url
      (str/split #"://") second
      (str/split #"/") first))

(defn- url->path [url]
  (-> url
      (str/split #"://") second
      (str/split #"/") rest
      (->> (str/join "/"))))

(defn- string->number [s]
  (let [s (apply str (re-seq #"[0-9]" s))]
```

```
    (when (seq s)
      (Integer/parseInt s))))

(defmulti product-name
  (fn [store & _] store))

(defmethod product-name "artline.ua" [_ doc]
  (j/text (j/select-first doc "h1.product__title")))

(defmethod product-name "brain.com.ua" [_ doc]
  (j/text (j/select-first doc "h1[itemprop=name]")))

(defmethod product-name "rozetka.com.ua" [_ doc]
  (j/text (j/select-first doc "h1.product__title")))

(defmethod product-name "hard.rozetka.com.ua" [_ doc]
  (product-name "rozetka.com.ua" doc))

(defmethod product-name "stylus.ua" [_ doc]
  (j/text (j/select-first doc "h1.page-name")))

(defmethod product-name "touch.com.ua" [_ doc]
  (j/text (j/select-first doc "h1.changeName")))

(defn product-name->sku [name]
  (second (re-find #"^.*\((.*)\)$" name)))

(defmulti product-sku
  (fn [store & _] store))

(defmethod product-sku "artline.ua" [_ doc]
  (j/text (j/select-first doc "span.product__article-item")))

(defmethod product-sku "brain.com.ua" [store doc]
  (product-name->sku (product-name store doc)))

(defmethod product-sku "rozetka.com.ua" [store doc]
  (product-name->sku (product-name store doc)))

(defmethod product-sku "hard.rozetka.com.ua" [_ doc]
  (product-sku "rozetka.com.ua" doc))

(defmethod product-sku "stylus.ua" [store doc]
  (product-name->sku (product-name store doc)))

(defmethod product-sku "touch.com.ua" [store doc]
  (product-name->sku (product-name store doc)))

(defmulti product-description
  (fn [store & _] store))

(defmethod product-description "artline.ua" [_ doc]
  (j/text (j/select-first doc "div.product__detail-item.c-article__content")))

(defmethod product-description "brain.com.ua" [_ doc]
  (j/text (j/select-first doc "div[itemprop=description]")))

(defmethod product-description "rozetka.com.ua" [_ doc]
  (j/text (j/select-first doc "div.product-about__description-content")))

(defmethod product-description "hard.rozetka.com.ua" [_ doc]
  (product-description "rozetka.com.ua" doc))
```

```
(defmethod product-description "stylus.ua" [_ doc]
  (j/text (j/select-first doc "div#product-tabs div.text-block")))

(defmethod product-description "touch.com.ua" [_ doc]
  (j/text (j/select-first doc "div.changeDescription")))

(defmulti product-price
  (fn [store & _] store))

(defmethod product-price "artline.ua" [_ doc]
  (when-let [e (j/select-first doc "div.product__price span.js-price")]
    (-> e j/text string->number)))

(defmethod product-price "brain.com.ua" [_ doc]
  (when-let [e (j/select-first doc "span[itemprop=price]")]
    (-> e j/text string->number)))

(defmethod product-price "rozetka.com.ua" [_ doc]
  (when-let [e (j/select-first doc "p.product-price__big")]
    (-> e j/text string->number)))

(defmethod product-price "hard.rozetka.com.ua" [_ doc]
  (product-price "rozetka.com.ua" doc))

(defmethod product-price "stylus.ua" [_ doc]
  (when-let [e (j/select-first doc "div.regular-price")]
    (-> e j/text string->number)))

(defmethod product-price "touch.com.ua" [_ doc]
  (when-let [e (j/select-first doc "a.price")]
    (-> e j/text string->number)))

(defn product-details [url]
  (let [store (url->hostname url)
        doc   (j/get url)]
    {:sku         (product-sku store doc)
     :name        (product-name store doc)
     :price       (product-price store doc)
     :description (product-description store doc)}))

(defmulti catalog-products url->hostname)

(defmethod catalog-products "artline.ua" [url]
  (-> (j/get url)
      (j/select "a.product-cart__title")
      (j/attr "abs:href")))

(defmethod catalog-products "brain.com.ua" [url]
  (-> (j/get url)
      (j/select "div.description-wrapper a[itemprop=url]")
      (j/attr "abs:href")))

(defmethod catalog-products "rozetka.com.ua" [url]
  (-> (j/get url)
      (j/select "a.goods-tile__heading")
      (j/attr "abs:href")))

(defmethod catalog-products "hard.rozetka.com.ua" [url]
  (catalog-products (str "https://rozetka.com.ua/" (url->path url))))

(defmethod catalog-products "stylus.ua" [url]
```

```
  (-> (j/get url)
      (j/select "div.content-block a.name-block")
      (j/attr "abs:href")))

(defmethod catalog-products "touch.com.ua" [url]
  (-> (j/get url)
      (j/select "div.productList div.tabloid a.name")
      (j/attr "abs:href")))

(defmulti product-url
  (fn [store & _] store))

(defmethod product-url "artline.ua" [store sku]
  (let [url (str "https://" store "/uk/search/result?search=" sku)
        first-product-url (first (catalog-products url))
        first-product-sku (:sku (product-details first-product-url))]
    (when (= first-product-sku sku) first-product-url)))

(defmethod product-url "brain.com.ua" [store sku]
  (let [url (str "https://" store "/ukr/search/?Search=" sku)
        first-product-url (first (catalog-products url))
        first-product-sku (:sku (product-details first-product-url))]
    (when (= first-product-sku sku) first-product-url)))

(defmethod product-url "rozetka.com.ua" [store sku]
  (let [url (str "https://" store "/ua/search/?text=" sku)]
    (e/with-firefox-headless driver
      (e/go driver url)
      (e/wait-visible driver {:css "div.goods-tile__inner"})
      (let [html (e/get-element-inner-html driver {:css "body"})
            first-product-url (-> (j/parse html)
                                  (j/select-first "div.goods-tile__inner a")
                                  (j/attr "abs:href"))
            first-product-sku (:sku (product-details first-product-url))]
        (when (= first-product-sku sku) first-product-url)))))

(defmethod product-url "stylus.ua" [store sku]
  (let [url (str "https://" store "/search?q=" sku)
        response-url (-> (j/get url)
                         (j/select-first "link[rel=canonical]")
                         (j/attr "abs:href"))
        redirected? (not= response-url
                          (str "https://" store "/search"))]
    (when redirected? response-url)))

(defmethod product-url "touch.com.ua" [store sku]
  (let [url (str "https://" store "/search/?q=" sku)
        link (-> (j/get url)
                 (j/select-first "link[rel=canonical]"))]
    (when link (j/attr link "abs:href"))))

(defn product-urls [sku]
  (for [store (keys (methods product-url))]
    (product-url store sku)))

(defn product-prices [sku]
  (->> (for [url (product-urls sku)]
         [(url->hostname url)
          (:price (product-details url))])
       (sort-by second)
       (into {})))
```

```
(defn product-best-price [sku]
  (-> sku
      product-prices
      first))

(defmulti product-specs url->hostname)

(defn- span->vector [span]
  (-> (.html span)
      (str/split #"<br>")
      (->> (map #(j/text (j/parse %)))
           vec)))

(defn- ul->vector [ul]
  (let [span (j/select ul "span")]
    (if (seq span)
      (span->vector span)
      (vec (map #(j/text %) (j/select ul "a"))))))

(defmethod product-specs "rozetka.com.ua" [url]
  (let [url (str url "/characteristics")
        items (-> (j/get url)
                  (j/select "div.characteristics-full__item"))]
    (->> items
         (map (fn [item]
                (let [label (j/select item "dt.characteristics-full__label")
                      value (j/select item "dd.characteristics-full__value")]
                  [(j/text label)
                   (ul->vector (j/select value "ul.characteristics-full__sub-list"))
                   ])))
         (into {}))))

(defmethod product-specs "hard.rozetka.com.ua" [url]
  (product-specs (str "https://rozetka.com.ua/" (url->path url))))

(defmulti product-reviews url->hostname)

(defmethod product-reviews "rozetka.com.ua" [url]
  (let [url (if-not (str/ends-with? url "/comments")
              (str url "/comments")
              url)
        doc (j/get url)]
    (for [comment (j/select doc "div.comment")]
      {:author (j/text (j/select-first comment "div.comment__author div div"))
       :text (j/text (j/select-first comment "p.comment__text"))
       :advantages (when-let [e (first (j/select comment "div.comment__essentials-
item"))]
                     (j/text (j/select-first e "dd")))
       :disadvantages (when-let [e (second (j/select comment "div.comment__essentials-
item"))]
                        (j/text (j/select-first e "dd")))})))

(defmethod product-reviews "hard.rozetka.com.ua" [url]
  (product-reviews (str "https://rozetka.com.ua/" (url->path url))))

(defn update-prices! [sku & [dest]]
  (let [dest (io/file (or dest "resources/prices.json"))]
    (when-not (.exists dest)
      (io/make-parents dest)
      (spit dest nil))
    (let [data (json/decode (slurp dest))
          prices (product-prices sku)
```

```
    datetime (str (java.time.Instant/now))]
(->> (json/encode (assoc-in data [sku datetime] prices))
     (spit dest)))))
```

ДОДАТОК Б

ПЕРЕЛІК ФАЙЛІВ НА ДИСКУ

Ім’я файлу	**Опис**
Пояснювальні документи	
Диплом_Крамаренко.docx	Пояснювальна записка до дипломного проекту. Документ Word.
Диплом_Крамаренко.pdf	Пояснювальна записка до дипломного проекту в форматі PDF
Програма	
Програма.zip	Архів. Містить коди програми
Презентація	
Презентація_Крамаренко.zip	Архів. Містить презентацію дипломного проекту