

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

«До захисту допущено»
Завідувач кафедри

(підпис) (ініціали, прізвище)

“ ____ ” червня 2023 р.

Дипломний проєкт

На здобуття ступеня бакалавра

Зі спеціальності 123 «Комп'ютерна інженерія»

На тему: Автоматизована система аналізу ринку та розрахування ціни товару

Виконала: студентка IV курсу, групи КВ-91

Денисенко Олександра Максимівна

Керівник: асистент каф. СПСКС, Радченко К.О.

Консультант з нормконтролю: доц. каф. СПСКС, к.т.н. Клятченко Я.М.

Рецензент _____

Засвідчую, що у цьому дипломному проєкті
немає запозичень з праць інших авторів без
відповідних посилань.

Студентка _____

(підпис)

Київ – 2023 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем
Рівень вищої освіти – перший (бакалаврський)
Спеціальність 123 «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

(підпис)

(ініціали, прізвище)

«__» червня 2023 р.

ЗАВДАННЯ

на дипломний проєкт студентки

Денисенко Олександрі Максимівни

1. Тема проєкту «Автоматизована система аналізу ринку та розрахування ціни товару»,

керівник проєкту асистент каф. СПСКС, Радченко К.О.,

затверджені наказом по університету від «__» _____ 2023 р. № _____

2. Термін подання студенткою проєкту _____

3. Вихідні дані до проєкту: див. Технічне завдання

4. Зміст пояснювальної записки:

- Аналіз існуючих рішень та обґрунтування теми дипломного проєкту;
- Вибір технології та програмних засобів для реалізації проєкту;
- Опис розробленої автоматизованої системи;
- Перспективи розвитку автоматизованої системи;
- Висновок.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо)

- Користувальська частина модуля. Схема алгоритму;
- Архітектура Amazon API Gateway. Схема структурна;
- Виклик функцій. Схема структурна;
- Алгоритм функції пошуку. Схема алгоритму;

6. Консультанти розділів проєкту*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання «04» листопада 2023р.

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	Вивчення літератури за тематикою проєкту.	17.04.2023	
2.	Розробка та узгодження технічного завдання.	30.04.2023	
3.	Аналіз існуючих рішень.	05.05.2023	
4.	Підготовка матеріалів першого розділу дипломного проєкту.	10.05.2023	
5.	Підготовка матеріалів другого розділу дипломного проєкту.	18.05.2023	
6.	Підготовка матеріалів третього та четвертого розділу дипломного проєкту.	20.05.2023	
7.	Підготовка графічної частини.	22.05.2023	
8.	Оформлення документації дипломного проєкту.	25.05.2023	
9.	Попередній огляд матеріалів диплому на кафедрі.	31.05.2023	

Студентка

(підпис)

О.М. Денисенко

(ініціали, прізвище)

Керівник проєкту

(підпис)

К.О. Радченко

(ініціали, прізвище)

* Консультантом не може бути зазначено керівника дипломного проєкту.

АНОТАЦІЯ

Кваліфікаційна робота включає пояснювальну записку (61 с., 29 рис. 3 табл., - додатки).

Об'єкт розробки – створення автоматизованої системи аналізу ринку та розрахування ціни товару.

Автоматизована система дозволяє: здійснювати збір та аналіз даних товарів на Amazon для подальшого порівняння та обчислення оптимальної ціни; зокрема додаток збирає ціни та кількість відгуків.

В ході розробки:

- проведено аналіз вимог до автоматизованої системи і формулювання вимог до неї.
- розроблено алгоритми пошуку, аналізу та порівняння даних з використанням Кеера API.
- розроблено та протестовано автоматизовану систему.
- внесено необхідні зміни та вдосконалено автоматизовану систему на основі результатів тестування.

Результатом розробки автоматизованої системи є забезпечення швидкого аналізу релевантних товарів на Amazon з метою обчислення ціни. Використання системи в компаніях-виробниках товарів і організаціях, що працюють з маркетингом на Amazon, дозволить швидко аналізувати ринок та удосконалювати маркетингові стратегії за допомогою калькулятора ціни.

Ключові слова:

АВТОМАТИЗОВАНА СИСТЕМА, КЕЕРА API, ЦІНА, ПОШУК ТОВАРІВ, АНАЛІЗ РИНКУ.

ABSTRACT

The Bachelor's thesis project includes an explanatory note (61 pages, 29 images, 3 tables, - appendices).

The object of development is the creation of an automated system for market analysis and product price calculation.

The automated system allows: collection and analysis of product data on Amazon for further comparison and calculation of the optimal price; in particular, the application collects prices and the number of reviews.

In the course of development:

- analysis of requirements for the automated system and formulation of requirements for it was carried out.
- developed algorithms for searching, analyzing and comparing data using the Keepa API.
- an automated system was developed and tested.
- the necessary changes were made and the automated system was improved based on the test results.

The result of the development of an automated system is to provide a quick analysis of relevant products on Amazon in order to calculate the price. Using the system in companies producing goods and organizations working with marketing on Amazon will allow you to quickly analyze the market and improve marketing strategies with the help of a price calculator.

Keywords:

AUTOMATED SYSTEM, KEEPA API, PRICE, PRODUCT SEARCH, MARKET ANALYSIS.

[illegible]

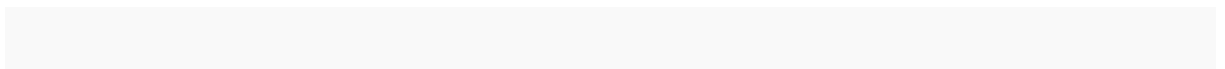
Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	A4	ІАЛПЦ.045490.005 Д1	Автоматизована			
			система для аналізу			
			ринку та розрахування			
			ціни			
			Користувальця частина			
			модуля. Схеми алгоритму			
	A4	ІАЛПЦ.045490.006 Д2	Автоматизована			
			система для аналізу			
			ринку та розрахування			
			ціни			
			Архітектура Amazon API			
			Gateway. Схеми структурна			
	A4	ІАЛПЦ.045490.007 Д3	Автоматизована			
			система для аналізу			
			ринку та розрахування			
			ціни			
			Виклик функцій. Схеми			
			структурна			
	A4	ІАЛПЦ.045490.008 Д4	Автоматизована			
			система для аналізу			
			ринку та розрахування			
			ціни			
			Алгоритм функції пошуку.			
			Схеми алгоритму			
ІАЛПЦ.045490.001 ОА						
Зміст.	Арк.	№ докум.	Підпис	Дата		
					Арк.	2

ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ.....	3
2. ПІДСТАВА ДЛЯ РОЗРОБКИ	3
3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ	3
4. ДЖЕРЕЛА РОЗРОБКИ.....	4
5. ТЕХНІЧНІ ВИМОГИ.....	4
5.1. Вимоги до програмного продукту, що розробляється	4
5.2. Вимоги до апаратного забезпечення.....	4
5.3. Вимоги до програмного та апаратного забезпечення користувача	5
6. ЕТАПИ РОЗРОБКИ	6
ВСТУП.....	4
1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБҐРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЄКТУ	5
1.1. Загальний огляд проблематики	5
1.2. Helium 10.....	6
1.3. Jungle Scout	8
1.4. Кеера.....	10
1.5. DataHawk.....	12
1.6. Висновки до розділу	16
2. ВИБІР ТЕХНОЛОГІЇ ТА ПРОГРАМНИХ ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ ПРОЄКТУ	17
2.1. Python.....	17

					ІАЛЦ. 467200.002 ТЗ			
Зм	Лист	№ докум.	Підп.	Дата	Автоматизована система аналізу ринку та розрахування ціни товару Технічне завдання	Лім.	Лист	Листів
Розроб.		Денисенко О.М.						
Перев.		Радченко К.О.					1	75
Н. контр.		Клятенко Я.М.				НТУУ «КПІ ім. Ігоря Сікорського», ФПМ, КВ-91		
Затв.		Тарасенко В.О.						

2.2. Логування активності	22
2.3. Кеера API	24
2.4. Вибір бази даних	27
2.5. Висновки до розділу	29
3. ОПИС РОЗРОБЛЕНОЇ АВТОМАТИЗОВАНОЇ СИСТЕМИ	31
3.1. Функціональні можливості	31
3.2. Архітектура системи	32
3.3. Опис користувацького інтерфейсу	33
3.4. Опис модуля пошуку товарів та обчислення ціни	36
3.5. Опис модуля бази даних	38
3.6. Висновки до розділу	39
4. ТЕСТУВАННЯ СИСТЕМИ ТА ЇЇ ВЗАЄМОДІЯ З КОРИСТУВАЧЕМ	41
4.1. Тестування функції пошуку продуктів автоматизованої системи	41
4.2. Тестування функції збереження історії пошуку та пошуку за збереженими фільтрами	44
4.3. Тестування функції логування активності користувача	49
4.4. Перспективи розвитку та розширення функціоналу	53
4.5. Висновки до розділу	54
ВИСНОВКИ	58
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	60



1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ

Назва розробки: «Автоматизована система аналізу ринку та розрахування ціни товару».

Галузь застосування: маркетинг та продаж товарів на Amazon (аналіз категорії товару на обчислення оптимальної ціни та потенційну прибутковість товару).

2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на дипломне проєктування на здобуття першого (бакалаврського) рівня вищої освіти, затверджене кафедрою системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський Політехнічний Інститут імені Ігоря Сікорського».

3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ

Метою даного проєкту є створення автоматизованої системи, що дозволяє швидко аналізувати ринок та ціновий діапазон товарів у певній категорії. Система зберігає час та робить обчислення ціни швидким процесом, а також потенційно допомагає збільшити продажі, оскільки ціна стає більш конкурентною на ринку. Завдяки простому інтерфейсу, користувачі можуть запустити аналіз за хвилину і швидко отримати обчислену ціну.

Призначенням продукту є автоматизація аналізу ринку та найбільш конкурентної ціни у заданій категорії. Це значно скорочує час пошуку релевантних товарів і дозволяє враховувати найважливіші фактори, що впливають на ціну.

					ІАЛЦ.467200.002 ТЗ	Лист
Зм	Лист	№ докум.	Підп.	Дата		3

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом інформації є технічна та науково-технічна література, технічна документація, публікації у періодичних виданнях та електронні статті у мережі Інтернет.

5. ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до програмного продукту, що розробляється

- Простий інтерфейс: автоматизована система повинна мати чистий та зручний інтерфейс, яким легко користуватися. Користуває повинен мати можливість вписувати ключові слова для пошуку товару або категорії у додатку. Додаток має показувати список продуктів, які є найближчими до заданого, вказуючи ціни, відгуки та іншу важливу інформацію. Користувачам також потрібна можливість зберігати історію пошуку та здійснювати пошук за існуючими фільтрами.
- Функціональність: система має отримувати дані про продукти за допомогою Кеера API та ключового слова, після чого він має обчислити найкращу ціну для продукту. Додаток має мінімізовувати усі можливі помилки та показувати повідомлення про помилку, якщо потрібно. Також він має періодично оновлювати інформацію про продукти, щоб ціна та інша інформація були найрелевантнішими.
- Продуктивність: автоматизована система має швидко завантажуватися та відповідати користувачу з результатом пошуку якнайшвидше.

5.2. Вимоги до апаратного забезпечення

- Процесор: Intel Pentium 4 або вище;

					ІАЛЦ.467200.002 ТЗ	Лист 4
Зм	Лист	№ докум.	Підп.	Дата		

- Оперативна пам'ять: 2 Гб;
- Місце на жорсткому диску: 200 Мб

5.3. Вимоги до програмного та апаратного забезпечення користувача

- Доступ до PyChart;
- Наявність операційної системи Windows, MacOS, або Linux;
- Доступ до бази даних та графічного клієнта pgAdmin 4.

					ІАЛЦ.467200.002 ТЗ	Лист
						5
Зм	Лист	№ докум.	Підп.	Дата		

6. ЕТАПИ РОЗРОБКИ

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів
1.	Вивчення літератури за тематикою проєкту	15.04.2023
2.	Розроблення та узгодження технічного завдання	30.04.2023
3.	Аналіз існуючих рішень	05.05.2023
4.	Підготовка матеріалів першого розділу дипломного проєкту	10.05.2023
5.	Підготовка матеріалів другого розділу дипломного проєкту	18.05.2023
6.	Підготовка графічної частини дипломного проєкту	20.05.2023
7.	Оформлення документації дипломного проєкту	25.05.2023
8.	Попередній огляд матеріалів диплому на кафедрі	30.05.2023

ВІДОМІСТЬ ДИПЛОМНОГО ПРОЄКТУ

№ з/п	Формат	Позначення	Найменування	Кількість листів	Примітка
1	A4	ІАЛЦ.045440.000	Завдання на дипломний проєкт	2	
2	A4	ІАЛЦ. 045440.001 ОА	Опис альбому	2	
3	A1	ІАЛЦ. 045440.002 ТЗ	Технічне завдання	5	
4	A1	ІАЛЦ. 045440.003 ТП	Відомість технічного проєкту	1	
5	A1	ІАЛЦ. 045440.004 ПЗ	Пояснювальна записка	61	
6	A1	ІАЛЦ. 045440.		1	
7	A1	ІАЛЦ. 045440.		1	
8	A1	ІАЛЦ. 045440.		1	
9	A1	ІАЛЦ. 045440.		1	

					ІАЛЦ. 045440.003 ТП			
Змін	Арк.	№ докум.	Підпис	Дата				
Розробив	Денисенко О.М.				Автоматизована система аналізу ринку та розрахування ціни товару Відомість технічного проєкту	Літ.	Аркуш	Аркушів
Перевірив	Радченко К.О.						1	2
						КПІ ім. Ігоря Сікорського, ФПМ КВ-91		
Н. контроль	Клятченко Я.М.							
Затвердив	Гарасенко В.П.							

**Пояснювальна записка
до дипломного проєкту**

На тему: Автоматизована система аналізу ринку та розрахування ціни товару

ЗМІСТ

ВСТУП	4
1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБҐРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЄКТУ	5
1.1. Загальний огляд проблематики	5
1.2. Helium 10	6
1.3. Jungle Scout	8
1.4. Кеера.....	10
1.5. DataHawk.....	12
1.6. Висновки до розділу	16
2. ВИБІР ТЕХНОЛОГІЇ ТА ПРОГРАМНИХ ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ ПРОЄКТУ	17
2.1. Python	17
2.2. Логування активності	22
2.3. Кеера API	24
2.4. Вибір бази даних	27
2.5. Висновки до розділу	29
3. ОПИС РОЗРОБЛЕНОЇ АВТОМАТИЗОВАНОЇ СИСТЕМИ	31
3.1. Функціональні можливості	31
3.2. Архітектура системи.....	32
3.3. Опис користувацького інтерфейсу.....	33
3.4. Опис модуля пошуку товарів та обчислення ціни.....	36
3.5. Опис модуля бази даних.....	38
3.6. Висновки до розділу	39
4. ТЕСТУВАННЯ СИСТЕМИ ТА ЇЇ ВЗАЄМОДІЯ З КОРИСТУВАЧЕМ.....	41
4.1. Тестування функції пошуку продуктів автоматизованої системи.....	41

4.2. Тестування функції збереження історії пошуку та пошуку за збереженими фільтрами	44
4.3. Тестування функції логування активності користувача	49
4.4. Перспективи розвитку та розширення функціоналу	53
4.5. Висновки до розділу	54
ВИСНОВОК.....	58
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	60

ВСТУП

Все більше людей здійснюють свої покупки через різні онлайн-платформи, однією з найбільш відомих і популярних серед них є платформа Amazon [18]. Під час пошуку найкращої ціни на товари користувачам часто доводиться переглядати безліч пропозицій та зіткнутися зі складнощами вибору оптимального варіанту. Оскільки Amazon має мільйони продуктів та користувачів [19], ця платформа є гарним джерелом пошуку продуктів та їхньої актуальності для подальшого аналізу ринку.

Цей дипломний проєкт присвячений розробці автоматизованої системи, яка забезпечує зручний та ефективний спосіб розрахунку найкращої ціни товару, враховуючи інформацію про товари з тієї ж самої категорії, їхню кількість відгуків, тощо. Для досягнення цієї мети використовується Кеера API, що дозволяє отримувати актуальні дані з онлайн-платформи Amazon.

Ця система має на меті сприяти зручному та об'єктивному аналізу при продажі товару, шляхом надання розширених можливостей порівняння цін, доступу до актуальної інформації про знижки та пропозиції замінників. Крім того, додаток враховує контекстні фактори, такі як рейтинг продавця, відгуки користувачів та інші параметри, що сприяють прийняттю обґрунтованих рішень при покупках. Також передбачено логування активності споживачів, зберігання історії їхнього пошуку та зручний інтерфейс.

У даній пояснювальній записці буде представлений детальний аналіз розробленої автоматизованої системи, її функціональності, способу взаємодії з Кеера API, обробки та аналізу даних, а також оцінка його надійності та користувацького інтерфейсу.

Цей дипломний проєкт важливий для вирішення актуальної проблеми пошуку найкращої ціни на товари в онлайн-магазинах та сприяє подальшому розвитку електронної комерції.

					ІАЛЦ.466538.004 ПЗ	Лист 4
Зм	Лист	№ докум.	Підп.	Дата		

1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБҐРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЄКТУ

1.1. Загальний огляд проблематики

Оскільки сфера онлайн-магазинів та електронно комерційних платформ швидко зростає, з'являється все більше рішень, які допомагають користувачам аналізувати ринок та поведінку споживачів, а також сприяють покращенню рівня продажів продавців. Однак вирішення проблеми оптимальної ціни досі не надано, а рішення, які пропонують аналіз ринку, потребують подальшого обчислення ціни та не мають фільтру по важливим показникам, таким як рейтинг та відгуки про товар, знижки, тощо. У цьому розділі будуть надані деякі приклади інструментів, які максимально наближені до вирішення проблеми точного аналізу ринку та обчислення ціни для заданого товару.

Helium 10: інструмент, який допомагає підприємцям на Amazon проводити комплексний аналіз ринку та конкурентів. Він надає можливість досліджувати ключові слова, виявляти нові ніші, а також здійснювати моніторинг рейтингу продуктів та аналізувати їхні зображення та описи.

Jungle Scout: Цей інструмент допомагає продавцям на Amazon знаходити прибуткові можливості для своїх продуктів. Він надає корисні дані про популярні продукти, їхні рейтинги, продажі та інші метрики, що допомагають приймати обґрунтовані рішення стосовно розміщення товарів та конкурентної стратегії.

Кеера: Цей інструмент забезпечує доступ до історичних даних про ціни на Amazon. Він надає графіки та діаграми, що демонструють зміни цін на товари, дозволяючи продавцям аналізувати рух цін на ринку та приймати обґрунтовані рішення щодо стратегії ціноутворення.

DataHawk: Цей інструмент надає розширений аналітичний звіт та моніторинг продажів на Amazon. Він допомагає відстежувати продажі своїх

товарів та конкурентів, аналізувати ключові метрики, такі як рейтинг, відгуки та конверсія, що допомагає підприємцям приймати рішення на основі даних.

Усі ці інструменти є корисними для продавців та організацій, які займаються маркетингом на Amazon. Однак кожен із представлених вище додатків має свої переваги і недоліки, тому користувачі обирають інструмент, який відповідає їхнім вимогам. Розглянемо, як саме відбувається аналіз ринку у кожному з цих інструментів та які недоліки вони мають.

1.2. Helium 10

Helium 10 розробляє та підтримує власний інфраструктурний шар, який взаємодіє з Amazon API. Цей шар забезпечує взаємодію між інструментами Helium 10 та Amazon API, обробку та збереження отриманих даних. Після отримання даних Helium 10 використовує власні алгоритми та аналітичні методи для обробки та аналізу цих даних. Це включає пошук потенційно прибуткових ніш, аналіз конкурентів, визначення популярних ключових слів та багато іншого. Результати аналізу надаються користувачам у зручному форматі через веб-інтерфейс або інші інструменти Helium 10 (рис. 1.1).

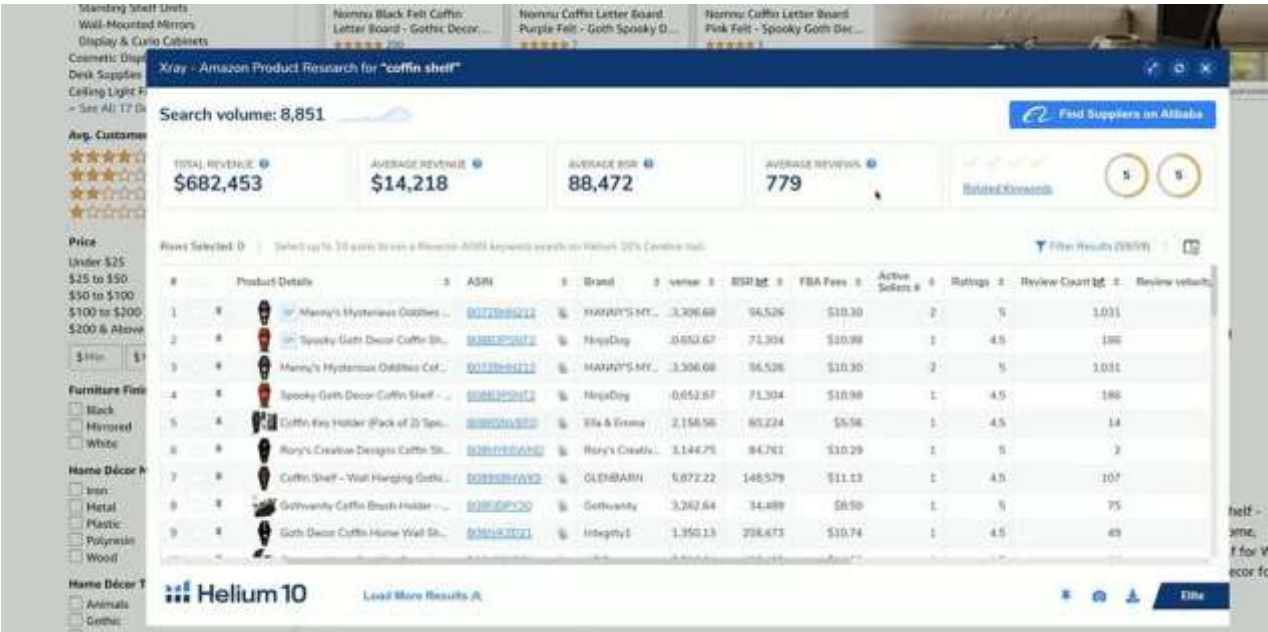


Рис. 1.1. Виведення результатів аналізу категорії у Helium 10

Цей інструмент використовує різні технології для аналізу та зберігання даних, наприклад:

- Системи розподіленого зберігання даних: Helium 10 використовує системи зберігання даних, такі як Amazon S3 (Simple Storage Service) або власні системи зберігання, для зберігання та обробки великих обсягів даних, що включають інформацію про ринок та товари на Amazon.
- Машинне навчання та аналіз даних: Helium 10 використовує техніки машинного навчання та аналізу даних для виявлення тенденцій, популярних ключових слів, конкурентних товарів та рейтингів. Це допомагає користувачам робити висновки щодо оптимальних стратегій продажу та маркетингу.
- Веб-скрапінг: Для збору даних зі сторінок Amazon, Helium 10 використовує техніку веб-скрапінгу. Це дозволяє отримувати інформацію про ціни, рейтинги, відгуки та інші деталі товарів з різних категорій та ринків.
- Natural Language Processing (NLP): Helium 10 використовує NLP для аналізу текстової інформації, такої як описи товарів, відгуки клієнтів та ключові слова. Це допомагає виявляти популярні та ефективні ключові слова для оптимізації списку продуктів.
- Обробка зображень: Для аналізу зображень товарів на Amazon, Helium 10 використовує технології обробки зображень, такі як computer vision. Це дозволяє автоматично визначати характеристики та особливості товарів на основі їх зображень.

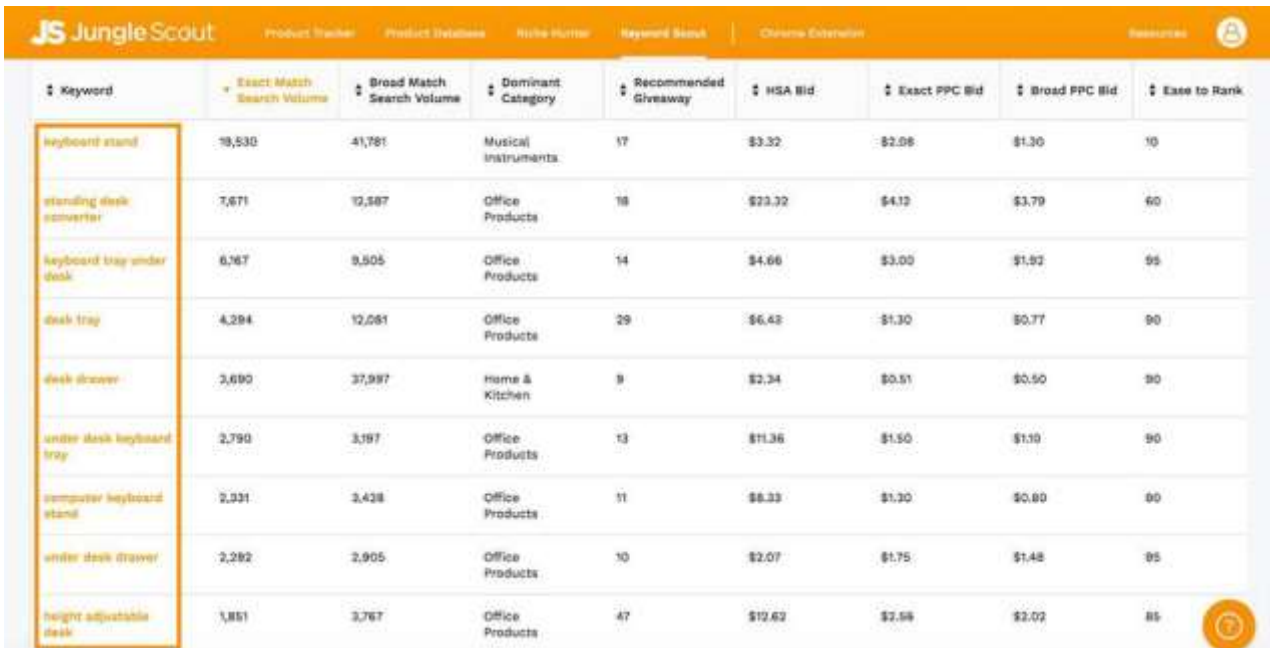
Основними недоліками Helium 10 є:

- Залежність від навчальних даних: Алгоритми машинного навчання можуть бути залежними від якості та репрезентативності навчальних даних. Також, обробка зображень може бути складною при наявності низької якості або неоднорідності вхідних зображень.

- Ціна: Одним з основних недоліків Helium 10 є висока вартість користування. Хоча план з базовими можливостями доступний безкоштовно, деякі продвинуті функції та плани вимагають щомісячної абонплати, що може бути високою для початківців та невеликих підприємств.
- Важкість для освоєння: Використання всіх функцій інструментів Helium 10 може бути вимогливим у плані навчання та освоєння. Для повного використання потенціалу цього інструменту може знадобитися час та додаткові зусилля, щоб ознайомитися з усіма функціями та їх оптимальним використанням.

1.3. Jungle Scout

Jungle Scout здебільшого використовується для пошуку успішних продуктів і брендів на Amazon. Завдяки їхній аналітиці, вони можуть приблизно прогнозувати поведінку ринку, а також продажі конкретних товарів, що дозволяє іншим продавцям розробляти нові продукти та покращувати існуючі. Jungle Scout схожий за функціоналом до Helium 10, але технології в цих інструментах різні. Приклад виведення інформації про популярні запити користувачів на Amazon наведено на рис. 1.2.



Keyword	Exact Match Search Volume	Broad Match Search Volume	Dominant Category	Recommended Giveaway	MSA Bid	Exact PPC Bid	Broad PPC Bid	Ease to Rank
keyboard stand	18,530	41,781	Musical Instruments	17	\$3.32	\$2.08	\$1.30	10
standing desk converter	7,671	12,587	Office Products	18	\$23.32	\$4.12	\$3.79	60
keyboard tray under desk	6,767	9,505	Office Products	14	\$4.66	\$3.00	\$1.92	95
desk tray	4,284	12,061	Office Products	29	\$6.42	\$1.30	\$0.77	90
desk drawer	3,690	37,997	Home & Kitchen	9	\$2.34	\$0.51	\$0.50	90
under desk keyboard tray	2,790	3,197	Office Products	13	\$11.36	\$1.50	\$1.10	90
computer keyboard stand	2,331	3,438	Office Products	11	\$8.33	\$1.30	\$0.80	90
under desk drawer	2,282	2,905	Office Products	10	\$2.07	\$1.75	\$1.48	95
height adjustable desk	1,851	3,767	Office Products	47	\$12.62	\$2.94	\$2.02	85

Рис. 1.2. Виведення інформації про популярні запити у Jungle Scout

Серед технологій, які використовує Jungle Scout, є наступні:

- Apache Hadoop: Це відкрите програмне забезпечення для обробки великих обсягів даних у розподіленому середовищі. Hadoop використовує модель програмування MapReduce, яка дозволяє розподілено обробляти та аналізувати великі обсяги даних. Це дозволяє Jungle Scout ефективно виконувати операції збору, обробки та аналізу даних.
- Apache Spark: Jungle Scout також використовує Apache Spark - потужну систему для обробки та аналізу великих обсягів даних у реальному часі. Spark пропонує широкий спектр функціональності для розподіленої обробки даних, включаючи операції маппінгу, згортання, фільтрації та статистичний аналіз. Це дозволяє Jungle Scout швидко і ефективно виконувати складні аналітичні завдання.
- Python та бібліотеки для аналізу даних: Jungle Scout використовує мову програмування Python та різноманітні бібліотеки, такі як pandas, NumPy і SciPy, для статистичного аналізу та обробки даних. Ці бібліотеки надають потужні інструменти для маніпулювання даними, виконання операцій агрегації, моделювання та виконання статистичних тестів.
- SQL бази даних: Jungle Scout може використовувати SQL бази даних, такі як MySQL або PostgreSQL, для зберігання та організації даних. SQL запити дозволяють Jungle Scout легко взаємодіяти з даними, виконувати складні запити та отримувати цінну інформацію для аналізу.

Хоча інструменти, як Apache Hadoop, Apache Spark, Python та SQL бази даних, мають багато переваг у обробці великих обсягів даних та статистичному аналізі, вони також мають свої недоліки.

- Складність налаштування: Ці інструменти часто вимагають досвідчених спеціалістів та часу на їх налаштування та розгортання. Налаштування кластерів Apache Hadoop або Apache Spark може бути складним завданням, і вимагає глибокого розуміння їх архітектури та компонентів.

- Великі вимоги до апаратного забезпечення: Для ефективної роботи з цими інструментами, зазвичай потрібні потужні сервери та великі кластери обчислювальних ресурсів. Це може бути дорогим і вимагати значних інвестицій у апаратну інфраструктуру.
- Складність управління: Для використання інструменту звичайному користувачеві потрібно витратити багато часу та зусиль. Деякі функції є незрозумілими або складними для вивчення.

1.4. Кеера

Кеера значно відрізняється від інших інструментів для аналітики категорій товарів, оскільки вона зберігає дані не цілої ніші, а окремих продуктів і постійно відстежує такі дані, як ціна, позиція в пошуку та категорії, відгуки, деталі продуктової сторінки та всіх продавців на цій сторінці про кожний існуючий товар (рис. 1.3.).

Details		Current & Average	
Image		Sales Rank - Current	# 63,761
Title	Signables Premium - Paris Saint Germain Lionel Messi Collectible - Official Football Facsimile - Premium Soccer Memorabilia Collectible	Sales Rank - 30 days avg.	# 52,501
Sales Rank - Reference	Sports & Outdoors	Sales Rank - 60 days avg.	# 48,163
Reviews - Rating	★★★★☆ 4.4	Sales Rank - 90 days avg.	# 47,297
Reviews - Review Count	58	Sales Rank - 180 days avg.	# 47,592
Last Price Change	14 days ago	Sales Rank - Drops last 30 days	49
Buy Box Seller	SIGNABLES (96%)	Sales Rank - Drops last 90 days	143
Buy Box: Is FBA	✓	Sales Rank - Drops last 180 days	276
Lowest FBA Seller	SIGNABLES (96%)	Reviews - Review Count - 90 days avg.	49
FBA Pick&Pack Fee	\$ 5.40	Reviews - Review Count - 180 days avg.	42
Referral Fee %	15 %	Buy Box - Current	\$ 19.99
Referral Fee based on current Buy Box price	\$ 3.00	Buy Box - 90 days avg.	\$ 23.56
Lowest FBM Seller	SIGNABLES (96%)	Buy Box - 180 days avg.	\$ 23.79
Tracking since	2022/01/26	Buy Box - Stock	35
Listed since	2021/12/20	Buy Box - 90 days OOS	0 %
Categories - Root	Sports & Outdoors	Amazon - 90 days OOS	100 %
Categories - Sub	Action Figures & Toy Figurines	New - Current	\$ 19.99
		New - 90 days avg.	\$ 23.56
		New - 180 days avg.	\$ 23.67
		New - 90 days OOS	0 %
		New, 3rd Party FBA - Current	\$ 19.99

Рис. 1.3. Виведення інформації про конкретний товар у Кеера
Для цього використовуються наступні технології:

- Amazon API: Кеера використовує Amazon API для отримання різноманітної інформації про продукти, включаючи ціни, рейтинги, обсяги продажів, історію цін та багато іншого. Це дозволяє Кеера збирати актуальні дані безпосередньо з джерела і використовувати їх для подальшого аналізу.
- Реляційні бази даних: Кеера використовує реляційні бази даних, такі як MySQL або PostgreSQL, для зберігання структурованих даних про продукти. Ці бази даних дозволяють ефективно організувати та зберігати великі обсяги даних, забезпечуючи швидкий доступ до них.
- NoSQL бази даних: Кеера також використовує NoSQL бази даних, наприклад MongoDB або Cassandra, для зберігання неструктурованих або напівструктурованих даних, таких як історія цін, рейтинги або коментарі. Ці бази даних дозволяють гнучко зберігати та обробляти різноманітні дані.
- Технології обробки даних: Кеера використовує різні технології обробки даних, включаючи потокову обробку даних (stream processing), таку як Apache Kafka або Apache Flink, для реалізації в реальному часі моніторингу та аналізу даних[5]. Такі технології дозволяють обробляти великі потоки даних та реагувати на них незабаром.
- Хмарні сервіси: Кеера може використовувати хмарні сервіси, такі як Amazon Web Services (AWS) або Google Cloud Platform (GCP), для забезпечення масштабованості, надійності та ефективності обробки та зберігання даних. Це дозволяє Кеера працювати з великими обсягами даних та швидко обслуговувати користувачів.

Загалом, використання різних баз даних та технологій обробки даних дозволяє Кеера ефективно зберігати, організовувати та аналізувати дані про продукти на Amazon для надання користувачам цінної інформації та аналітики.

Перелічимо недоліки інструменту Кеера:

- Складність інтерпретації даних: Кеера надає велику кількість даних про ціни та ранг продуктів на Amazon, але інтерпретація цих даних може бути складною для новачків. Вимагається певне розуміння ринку та контексту для ефективного використання інформації, наданої Кеера.
- Відсутність прогностичних можливостей: Кеера зосереджується на наданні історичних даних про ціни та ранг продуктів. Він не надає прогностичних аналітичних можливостей або передбачень щодо майбутньої продуктивності продуктів.
- Залежність від сторонніх сервісів: Кеера використовує сторонні сервіси для зберігання та обробки даних. Це може створювати проблеми, якщо сервіси, з якими вони взаємодіють, стають недоступними або зазнають перебоїв.
- Обмежена масштабованість: При обробці великих обсягів даних, Кеера може зіткнутися з обмеженнями масштабованості. Він може виявити недостатню продуктивність або ефективність при опрацюванні великої кількості даних або при великому навантаженні.

1.5. DataHawk

Інструмент для аналітики DataHawk дозволяє бачити велику кількість метрик на одній платформі. Він показує статистику по продуктах, динаміку ринку по цінах та відгуках, дозволяє швидко реагувати на зміни в продуктах конкурентів. Цей інструмент можна прив'язати до свого акаунту продавця на Amazon, щоб слідкувати за тенденціями продажів, відгуків, тощо (рис. 1.4).

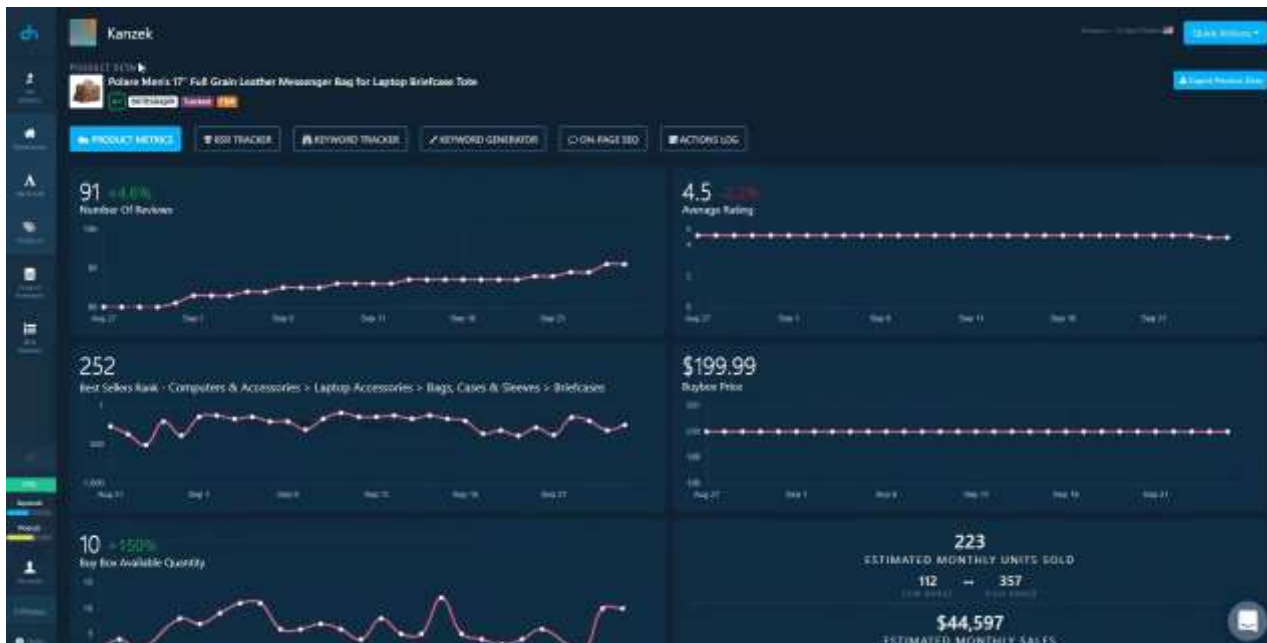


Рис. 1.4. Панель для відстежування метрик у DataHawk

Основні технології, якими користуються DataHawk, включають:

- **Web-скрапінг:** DataHawk використовує web-скрапінг для отримання даних з веб-сторінок Amazon. Він аналізує структуру та вміст сторінок продуктів, отримує інформацію про ціни, рейтинги, відгуки та інші параметри продуктів.
- **API Amazon:** DataHawk використовує API Amazon для отримання доступу до додаткової інформації та даних, що не доступні шляхом web-скрапінгу. API Amazon надає програмний інтерфейс, який дозволяє DataHawk отримувати дані про продажі, статистику продуктів, інформацію про конкурентів та інше.
- **Машинне навчання:** DataHawk використовує техніки машинного навчання для аналізу та обробки великих обсягів даних. Це дозволяє автоматично виявляти тенденції, розпізнавати патерни та здійснювати прогнозування щодо продажів, рейтингів та інших параметрів продуктів.
- **Amazon S3 (Simple Storage Service):** DataHawk може використовувати Amazon S3 для зберігання великого обсягу даних, включаючи деталі продуктів, ціни, рейтинги та відгуки. S3 надає

масштабовану та надійну систему зберігання з можливістю доступу до даних з будь-якого місця.

- Amazon EC2 (Elastic Compute Cloud): EC2 використовується DataHawk для розгортання та керування віртуальними серверами, необхідними для обробки даних та виконання обчислювальних завдань. EC2 дозволяє масштабувати обчислювальні ресурси залежно від потреб користувача.

У використанні перелічених технологій є свої недоліки, наприклад:

- Залежність від мережі: Використання хмарних технологій передбачає постійний доступ до Інтернету і стабільне з'єднання з хмарними ресурсами. При відсутності або обмеженому доступі до мережі можуть виникати проблеми з доступом до даних та виконанням операцій[7].

- Проблеми з безпекою: Збереження даних та обробка їх в хмарному середовищі може ставити питання щодо безпеки. Хоча провайдери хмарних послуг приймають заходи для захисту даних, існує ризик злому безпеки, втрати даних або несанкціонованого доступу до них.

Узагальнення переваг і недоліків вище описаних програмних рішень наведено у таблиці 1.

Табл. 1

Порівняльна характеристика існуючих рішень

	Helium 10	Jungle Scout	Keepa	DataHawk
Переваги	Використовує NLP, щоб виявити популярні та ефективні ключові слова для оптимізації списку продуктів;	Використовує Apache Hadoop для обробки великих обсягів даних, що дозволяє інструменту збирати, обробляти та аналізувати інформацію;	Використовує Amazon API для отримання даних про продукти;	Використовує Amazon API для отримання даних про продукти;

	Автоматично визначає характеристики та особливості товарів на основі їх зображень;	Використовує Apache Spark для швидкого і ефективного виконання складних аналітичних завдань;	Використовує різні технології обробки даних, включаючи потокову обробку даних, для великого обсягу інформації;	Використовує Amazon S3 для зберігання великого обсягу даних, включаючи деталі продуктів;
	Використовує системи розподіленого зберігання даних для зберігання та обробки великих обсягів даних, що включаються інформацію про ринок та товари на Amazon;	Використовує Python, що дозволяє легко маніпулювати даними та виконувати статистичні тести;	Використовує хмарні сервіси (Amazon Web Services, Google Cloud Platform) для роботи з великими обсягами даних та швидкого обслуговування користувачів;	Використовує Amazon EC2 для масштабування обчислювальних ресурсів залежно від потреб користувача;
	Використовує машинне навчання та аналіз даних для виявлення тенденцій, популярних ключових слів, конкурентних товарів та рейтингів;	Використовує SQL бази даних для зберігання, організації даних та легкої взаємодії з ними.	Використовує SQL та NoSQL бази даних для операцій над даними.	Використовує машинне навчання та аналіз даних для виявлення тенденцій, популярних ключових слів, конкурентних товарів та рейтингів;
	Використовує веб-скрапінг для отримання інформації про ціни, рейтинги та відгуки.			Використовує веб-скрапінг для отримання інформації про ціни, рейтинги та відгуки.

Недоліки	Залежність від навчальних даних;	Складність налаштування кластерів Apache Hadoop та Apache Spark;	Відсутність прогностичних можливостей;	Залежність від мережі;
	Висока ціна, найважливіші функції доступні лише у платній версії;	Великі вимоги до апаратного забезпечення;	Залежність від сторонніх сервісів для зберігання даних та обмежена масштабованість;	Проблеми з безпекою.
	Складність управління для користувача.	Складність управління для користувача.	Складність інтерпретації даних для користувачів.	

1.6. Висновки до розділу

Наявні рішення мають свої переваги та недоліки, хоча всі вони користуються неабиякою популярністю, оскільки користувачам потрібен інструмент для кращого менеджменту та просування товарів на Amazon.

Створення рішення, яке дозволить швидко аналізувати ринок та обчислювати оптимальну ціну на даний момент, а також матиме зручний і зрозумілий інтерфейс, сприятиме розвитку електронної комерції та допоможе організаціям вдало вести бізнес на Amazon.

2. ВИБІР ТЕХНОЛОГІЇ ТА ПРОГРАМНИХ ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ ПРОЄКТУ

Є багато інструментів, які допомагають організаціям та продавцям на Amazon аналізувати ринок та відстежувати дані про товари. Однак кожен з них має свої недоліки з точки зору використаних технологій, зручності для користувача або наявних функцій. Саме тому важливо створити такий додаток, який вирішить проблему обчислення ціни, при цьому швидко аналізуючи категорії товарів.

Основними критеріями, яким має відповідати автоматизована система, є:

- Аналіз даних за заданою категорією товарів;
- Фільтрування товарів за кількістю відгуків та ціновим діапазоном;
- Обчислення оптимальної ціни в заданій категорії;
- Логування активності споживачів;
- Збирання історії пошуку споживачів;
- Простий і зрозумілий інтерфейс.

2.1. Python

Для створення автоматизованої системи аналізу ринку та розрахування ціни товару потрібна мова програмування з широким функціоналом та можливістю швидко змінювати код, параметри продуктів. Зазвичай для цього використовуються мови Python та JavaScript.

Python - високорівнева, інтерпретована і багатоцільова мова програмування[14]. Він відомий своєю простотою, зрозумілістю та ефективністю розробки.

Переваги Python:

1. Простота використання: Python має простий синтаксис та зрозумілі правила, що дозволяє легко вчитися і розуміти код.

2. Висока читабельність: Python підтримує чіткі та лаконічні конструкції, що полегшує розуміння коду і сприяє спільній роботі в команді.

3. Багата стандартна бібліотека: Python поставляється з великою кількістю корисних модулів та функцій, що спрощує розробку і дозволяє швидко втілювати різноманітні функціональності.

4. Переносимість: Python може працювати на різних платформах, таких як Windows, macOS та Linux [\[4\]](#), що забезпечує гнучкість у розгортанні додатків.

Хоча Python є потужною та популярною мовою програмування, він також має свої недоліки:

1. Пам'ять: Python споживає більше пам'яті порівняно з деякими іншими мовами програмування [\[15\]](#). Це може бути проблемою, особливо при роботі з великими обсягами даних.

2. Обмеження в мобільній розробці: Хоча Python має фреймворки для розробки мобільних додатків, він не є найпопулярнішим вибором для цієї сфери. Інші мови, такі як Java або Kotlin, є більш поширеними в середовищі мобільної розробки.

3. Версійна несумісність: Python має кілька версій, таких як Python 2 та Python 3. Це може створювати проблеми, оскільки деякі старі бібліотеки та інструменти можуть бути несумісними з новішими версіями Python.

4. Низька підтримка для мобільних платформ: Якщо вам потрібно розробити мобільний додаток, специфічний для iOS або Android, Python може бути обмеженим в цьому відношенні. Існують інші мови, які мають кращу підтримку для мобільної розробки.

Ці недоліки не означають, що Python неефективний або непридатний для розробки. Враховуючи його простоту використання, розширюваність та багату екосистему, Python залишається однією з найпопулярніших мов програмування для широкого спектру завдань.

Для розробки веб-додатків та багатьох рішень у браузері використовують також JavaScript. JavaScript - це мова програмування, яка була створена для надання можливостей взаємодії з користувачем на стороні клієнта (у браузері). JavaScript є вбудованою мовою для всіх сучасних браузерів і виконується безпосередньо у середовищі браузера. Це дозволяє розробникам створювати динамічні та інтерактивні веб-сторінки з реагуючими елементами, анімацією, перевіркою даних та багато іншого[3].

Одна з ключових переваг JavaScript полягає в його експресивності та гнучкості. Вона має не дуже складний синтаксис і надає можливість використання функціонального, об'єктно-орієнтованого та імперативного стилю програмування. JavaScript також підтримує асинхронне програмування, що дозволяє виконувати операції в фоновому режимі та не блокувати основний потік виконання.

У розробці JavaScript використовується для реалізації функціональності на стороні клієнта, такої як валідація форм, взаємодія зі сторінкою, динамічне оновлення даних, анімація та багато іншого. Він дозволяє створювати багатофункціональні та зручні інтерфейси, що забезпечують покращену користувацьку взаємодію.

JavaScript має досить багато переваг, через які розробники часто обирають цю мову:

1. Веб-орієнтованість: JavaScript є основною мовою для веб-розробки і має потужні інструменти для взаємодії зі стороною клієнта[3]. Це дозволяє легко візуалізувати та відображати дані на веб-сторінках.

2. Багатий вибір фреймворків: JavaScript має велику екосистему фреймворків, таких як React, Angular та Vue.js. Ці інструменти дозволяють швидко розробляти інтерактивні користувацькі інтерфейси та обробляти дані в реальному часі.

3. Асинхронний код: JavaScript підтримує асинхронне програмування, що дозволяє виконувати операції в фоновому режимі без блокування

основного потоку виконання. Це особливо корисно для обробки великих обсягів даних та взаємодії зі зовнішніми сервісами.

4. Швидкодія в браузері: JavaScript виконується безпосередньо в браузері, що дозволяє зменшити час завантаження сторінок та забезпечує більш плавну інтерактивну взаємодію з користувачем[2].

Серед недоліків JavaScript у контексті автоматизованої системи для аналізу ринку цін можна вказати наступне:

1. Синтаксична помилковість: JavaScript має ряд особливостей в своєму синтаксисі, які можуть призвести до помилок, особливо для новачків[8]. Такі помилки можуть бути складними для відлагодження та виправлення.

2. Відсутність вбудованих інструментів аналізу даних: У порівнянні з Python, JavaScript має обмежені можливості у сфері аналізу даних. Хоча JavaScript має певні бібліотеки для цього, вони не настільки розширені та потужні, як у Python.

3. Недостатня підтримка мови: Хоча JavaScript є стандартом для браузерів, він не має такої широкої підтримки і інструментів, як Python. Наприклад, бібліотеки та модулі для роботи зі штучним інтелектом, наукових обчислень та аналізу даних у JavaScript менш поширені.

Переваги та недоліки кожної з цих мов можна порівняти та узагальнити в табл.2.

Табл.2

Порівняння переваг та недоліків Python та JavaScript

Характеристика	Python	JavaScript
Синтаксис	Чистий та лаконічний	Гнучкий та експресивний

Читабельність	Висока читабельність коду	Вимагає уваги до деталей та керівництва стилем
Простота використання	Легко вивчається та розуміється	Широкий спектр можливостей, вимагає досвіду
Багатоцільовість	Використовується для веб, наукових досліджень, штучного інтелекту тощо	Зосереджений на веб-розробці
Екосистема	Велика стандартна бібліотека та сторонні пакети	Велика кількість бібліотек для фронтенду
Обробка помилок	Зручна система обробки винятків	Керування помилками засноване на виключеннях
Швидкодія	Завдяки компіляції в байткод, Python виконується досить швидко	Виконується безпосередньо в браузері

Оскільки автоматизована система націлена на роботу в консолі, а не в браузері, доцільно обрати Python як основну мову програмування. Також він простий у використанні, має велику стандартну бібліотеку та сторонні пакети, що дозволяє швидко реалізувати потрібну функціональність. Python також забезпечує високу читабельність коду, що спрощує його розуміння та

підтримку у майбутньому. З урахуванням цих переваг, Python є оптимальним вибором для автоматизованої системи.

У системі для аналізу ринку та розрахування ціни Python використовуватиметься для наступних задач:

Розробка: Python використовується для написання основної логіки додатку. Можна створювати класи, функції та модулі, які забезпечують необхідну функціональність, включаючи взаємодію з Кеера API, роботу з базою даних та обробку результатів пошуку продуктів.

Бібліотеки: Python має велику кількість сторонніх бібліотек, що дозволяють розширити функціональність додатку. Наприклад, ця мова використовує бібліотеки, які спрощують роботу з базою даних PostgreSQL, здійснюють HTTP-запити до Кеера API, тощо.

2.2. Логування активності

Логування - це процес записування активності, подій та інших важливих відомостей у спеціально створений файл, який називається log-файл[9]. У Python існує вбудований модуль logging, який надає зручні і потужні засоби для логування в програмах.

Переваги використання логування в автоматизованій системі на основі Python:

1. Відстеження помилок: Лог-файл дозволяє виявляти та відстежувати помилки та проблеми, які виникають під час роботи системи. Він містить інформацію про виниклі помилки, стек викликів та інші деталі, що допомагають виявити та усунути проблеми.

2. Аналіз та статистика: Лог-файл може містити важливі дані про виконані операції, статуси процесів, тривалість виконання та іншу інформацію, яка дозволяє аналізувати роботу системи[13]. Це допомагає виявляти тренди, удосконалювати продуктивність та приймати рішення на основі зібраних даних.

3. Аудит та безпека: Лог-файли є важливим інструментом для аудиту системи та забезпечення безпеки. Вони зберігають інформацію про дії користувачів, зміни в системі та інші події, які можуть бути важливими для виявлення можливих загроз або несанкціонованої діяльності.

4. Відладка: Лог-файли допомагають у відладці програм, оскільки містять інформацію про стан системи на різних етапах виконання. Вони дозволяють відстежувати значення змінних, стан об'єктів та іншу важливу інформацію, що сприяє виявленню та виправленню помилок у програмі.

Логування в Python здійснюється за допомогою модуля logging, який надає різні рівні логування, такі як DEBUG, INFO, WARNING, ERROR та CRITICAL. Функціонал дозволяє налаштувати форматування повідомлень, вказати, куди зберігати лог-файли, обробляти помилки та інше.

Логування є унікальною функцією Python, яка надає зручність та прозорість у відстеженні роботи програми. Воно допомагає розробникам та користувачам зрозуміти, як працює програма, знайти та виправити проблеми, а також покращити продуктивність і безпеку системи.

Щоб забезпечити логування, використовується досить простий алгоритм дій:

1. Імпортування модуля logging: Спочатку потрібно імпортувати модуль logging у свою програму:

```
import logging
```

2. Конфігурація логгера: потрібно налаштувати логгер та встановити рівень логування та обробники:

```
logger = logging.getLogger()
logger.setLevel(logging.INFO)
```

3. Логування повідомлень: метод info() використовується для запису повідомлень:

```
logger.info('Це повідомлення для логування')
```

4. Закриття логгера: після завершення роботи потрібно закрити логгер:

logging.shutdown()

2.3. Кеера API

Для пошуку та аналізу товарів, а також для порівняння цін можна використовувати Кеера API або Amazon API Gateway, які надають дані із своєї системи для подальшого використання розробниками у своїх додатках, системах, тощо.

Amazon API - це набір програмних інтерфейсів, які надають розробникам доступ до різноманітних сервісів та даних, що надаються платформою Amazon. Ці сервіси включають електронну комерцію, обчислювальні послуги, зберігання даних, штучний інтелект, машинне навчання та багато інших.

Один з ключових сервісів, що надаються Amazon API, це Amazon Product Advertising API. Цей API дозволяє розробникам отримувати доступ до бази даних Amazon, що містить інформацію про мільйони продуктів, їх характеристики, ціни, відгуки користувачів та багато іншого. За допомогою цього API розробники можуть створювати додатки та сервіси, які здійснюють пошук, порівняння та аналіз продуктів на платформі Amazon. Для використання Amazon API розробник повинен отримати ключ доступу (API Key) та підписатися на використання конкретних сервісів Amazon.

Одним із сервісів, які дозволяють користуватися API від Amazon, є Amazon API Gateway. Технічно зв'язок з Amazon API Gateway відбувається за допомогою HTTP-запитів. Основні кроки включають:

1. Створення API: Розробник створює API у консолі Amazon API Gateway. В цьому процесі визначаються маршрути, методи, параметри та налаштування безпеки.

2. Визначення ресурсів та методів: Ресурси визначають структуру URL-адреси API, наприклад, /products або /orders. Кожний ресурс містить методи, такі як GET, POST, PUT або DELETE, які виконують певні дії над ресурсом.

3. Налаштування інтеграції: Розробник налаштовує інтеграцію API Gateway з бекенд-системою, яка надає фактичну функціональність API. Це може бути сервіс Amazon S3, Amazon Lambda або будь-який інший веб-сервіс або серверний додаток.

4. Розгортання API: Після налаштування API він може бути розгорнутий для публічного доступу. Amazon API Gateway надає URL-адресу, за допомогою якої можна звертатися до API.

5. Використання API: Розробник або клієнтська програма може використовувати URL-адресу API для виконання HTTP-запитів до ресурсів API. Ці запити можуть бути виконані з різних додатків або сервісів[1].

6. Коли клієнт здійснює запит до API, Amazon API Gateway приймає цей запит, перевіряє його за допомогою налаштованої політики безпеки, передає його до бекенд-системи та повертає відповідь клієнту.

Схема роботи Amazon API Gateway зображена на рис.2.1.

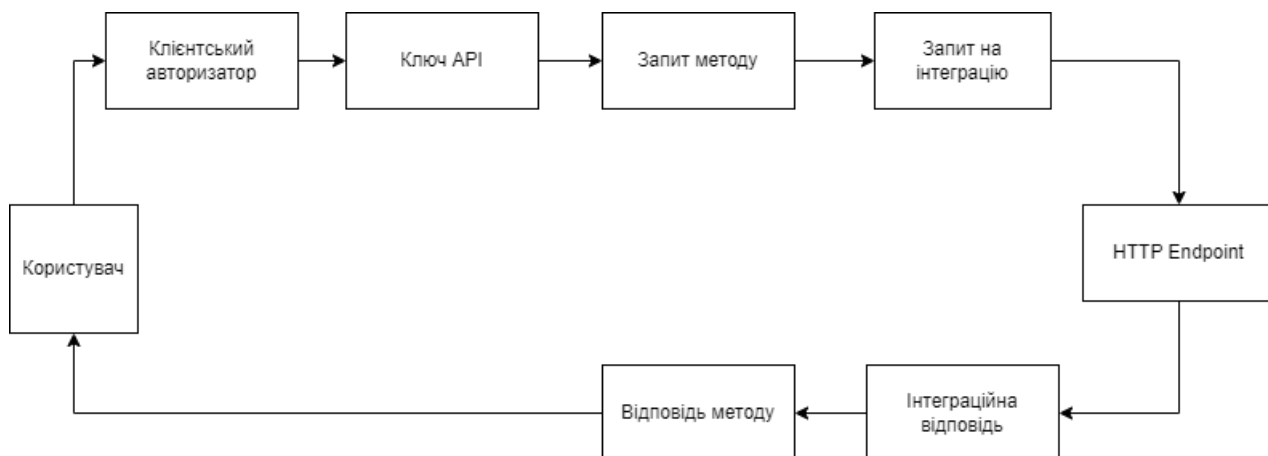


Рис.2.1. Схема роботи Amazon API Gateway

Кеера API - це програмний інтерфейс (API), який наає доступ до інформації про товари та ціни на Amazon. Він забезпечує розширені можливості для отримання даних про товари, їх історію цін, рейтинги, статистику продажів та іншу корисну інформацію, яка допомагає при аналізі ринку та прийнятті рішень[5].

Кеера API відрізняється від Amazon API наступними основними аспектами:

1. Доступ до історії цін: Однією з ключових переваг Кеера API є можливість отримання даних про історію цін на товари на протязі тривалого періоду. Це дозволяє аналізувати тенденції цін, виявляти знижки та інші фактори, які впливають на ринкову конкуренцію.

2. Розширена статистика та рейтинги: Кеера API надає детальну статистику та рейтинги для товарів на Amazon. Це включає в себе інформацію про рейтинг продукту, кількість продажів, популярні категорії, ранжування впливових продавців та багато іншого. Ці дані допомагають визначити популярність товару і його конкурентоспроможність на ринку.

3. Гнучкі фільтри та параметризація: Кеера API надає широкий спектр фільтрів та параметрів для зручного пошуку та сортування товарів. Ви можете налаштувати фільтри за ключовими словами, ціновими діапазонами, рейтингами, категоріями та іншими характеристиками, щоб знайти потрібні товари для аналізу та порівняння.

4. Легкість використання: Кеера API забезпечує простий та зрозумілий спосіб взаємодії з системою. Вона працює на основі HTTP-запитів, що дозволяє легко інтегрувати її в ваші програми або сервіси. Кеера API надає детальну документацію та приклади, як використовувати різні ендпоінти для отримання потрібної інформації.

У порівнянні з Amazon API, Кеера API надає більш розширений набір функцій та можливостей для аналізу ринку та збору даних про товари на Amazon. Вона дозволяє отримувати детальну інформацію про ціни, статистику продажів та рейтинги, що допомагає приймати обґрунтовані рішення та визначати конкурентну стратегію.

Порівняння переваг і недоліків Кеера API та Amazon API можна узагальнити у табл. 3.

					ІАЛЦ.466538.004 ПЗ	Лист 26
Зм	Лист	№ докум.	Підп.	Дата		

Порівняння переваг і недоліків Кеера API та Amazon API

Характеристика	Кеера API	Amazon API
Доступ до історії цін	Надає повну історію цін на товари	Обмежена доступність історії цін
Статистика та рейтинги	Забезпечує детальну статистику та рейтинги	Обмежена статистика та рейтинги
Фільтри та параметризація	Надає гнучкі фільтри для пошуку товарів	Обмежені фільтри та параметризація
Легкість використання	Легко інтегрується та має простий інтерфейс	Має складніший процес інтеграції
Обмеження доступу	Немає обмежень доступу до API	Обмеження доступу до деяких функцій та даних
Підтримка та документація	Надає детальну документацію та підтримку	Обмежена документація та підтримка
Комплексність даних	Забезпечує широкий набір даних для аналізу	Обмежений набір даних для аналізу
Ціна	Має підписку на користування сервісами Кеера, які включають використання API.	Має окрему платформу для використання API, кожна з функцій коштує досить дорого.

2.4. Вибір бази даних

Реляційна база даних (RDBMS) і нереляційна база даних (NoSQL) - це два основних типи баз даних, які використовуються в програмуванні та зберіганні даних[11].

Реляційна база даних (RDBMS):

- Реляційна база даних зберігає дані в структурованому форматі у вигляді таблиць, які мають стовпці та рядки.
- Дані у реляційній базі даних організовані у взаємозв'язки між таблицями за допомогою ключів (первинних та зовнішніх ключів) [\[20\]](#).
- Реляційна база даних використовує мову структурованих запитів (SQL) для роботи з даними, здійснення запитів, оновлення, вставки та видалення даних [\[6\]](#).
- Цей тип бази даних добре підходить для сценаріїв, де структура даних сталі або мало змінюється, і коли потрібні суворі перевірки цілісності та відношення між даними.

Нереляційна база даних (NoSQL):

- Нереляційна база даних зберігає дані у неструктурованому або полуструктурованому форматі, такому як JSON-подібні документи, ключ-значення, стовпцеві схеми тощо.
- Вона не вимагає фіксованої схеми даних, що дозволяє гнучко зберігати різнорідні дані.
- Нереляційна база даних підтримує горизонтальне масштабування, що означає, що можна легко розширити обсяг даних шляхом додавання нових серверів [\[12\]](#).
- Цей тип бази даних зазвичай використовує спеціалізовані запитові мови або програмні інтерфейси для взаємодії з даними.
- Нереляційні бази даних добре підходять для сценаріїв, де потрібно швидко збереження та отримання великого обсягу змінних даних, таких як соціальні мережі, аналітика веб-даних, IoT (речі Інтернету) тощо [\[11\]](#).

При виборі бази даних для зберігання активності та уподобань користувачів в автоматизованій системі, важливо враховувати кілька факторів, таких як швидкодія, масштабованість, надійність і зручність роботи з базою даних. Одним з популярних варіантів для таких сценаріїв є використання

реляційних баз даних, таких як MySQL або PostgreSQL, або нереляційних баз даних, таких як MongoDB.

Для даної автоматизованої системи для аналізу ринку можна обрати PostgreSQL. Він надає розширені можливості для зберігання та обробки даних[10]. Основні переваги PostgreSQL включають:

- Розширені функції: PostgreSQL підтримує багато розширених функцій, таких як вбудований пошук повнотекстового пошуку, географічні запити, JSON-операції тощо[16].
- Надійність: PostgreSQL забезпечує високу стійкість до відмов, має механізми відновлення та підтримує транзакції з високим рівнем ізоляції[17].
- Розширюваність: PostgreSQL має можливість створювати власні типи даних та функції, що дозволяє гнучко налаштовувати базу даних під ваші потреби.

2.5. Висновки до розділу

Під час розробки дипломного проєкту, використовувалися різні технології та програмні засоби для створення автоматизованої системи аналізу ринку на основі даних з Amazon. Основними технологіями, які були використані, є Python, логування активності через log-файли, база даних PostgreSQL та Keepa API.

Python - це потужна та універсальна мова програмування, яка надає багато переваг для розробки системи. Його простота використання, велика кількість доступних бібліотек та фреймворків, а також широкий спектр можливостей роблять його ідеальним вибором для реалізації проєкту. За допомогою Python, вдалося створити функціонал пошуку продуктів, обробку даних, взаємодію з Keepa API та збереження інформації у базі даних PostgreSQL.

Логування активності було дуже важливою частиною проєкту. Використання лог-файлу дозволило записувати всю активність користувача,

включаючи запити до API, отримані результати та інші важливі події. Це дозволило зберігати історію дій користувачів та аналізувати цю інформацію для виявлення трендів, вдосконалення системи та вирішення проблем.

База даних PostgreSQL була використана для збереження інформації про користувачів, їхні фільтри пошуку та історію пошуків. PostgreSQL - це потужна та надійна реляційна база даних, яка надає широкі можливості для зберігання та оптимізації даних. За допомогою PostgreSQL, вдається зберігати та витягувати дані для відображення історії пошуку користувачів.

Кеера API було використано для отримання розширених даних про товари та ціни на Amazon. Його перевагами є доступ до історії цін, статистики продажів та рейтингів, а також гнучкі фільтри для пошуку товарів. Це надало можливість аналізувати тенденції цін, розробляти стратегії ціноутворення та визначати конкурентну позицію на ринку.

Усі ці технології та програмні засоби допомогли створити потужну та ефективну автоматизовану систему аналізу ринку. Використання цих інструментів забезпечило надійність, гнучкість та зручність у роботі з даними Amazon, а також можливість виявлення та аналізу ключових факторів, що впливають на ціни та ринкову конкуренцію.

3. ОПИС РОЗРОБЛЕНОЇ АВТОМАТИЗОВАНОЇ СИСТЕМИ

Автоматизована система дозволить користувачам отримувати детальну інформацію про товари на Amazon та обчислювати їх оптимальну ціну. Система буде базуватися на зборі даних з публічного Кеера API та використовуватиме алгоритми для аналізу цін, перегляду відгуків користувачів. Крім того, система буде збирати активність користувачів та історію їхнього пошуку для поліпшення персоналізації та вдосконалення аналітичних можливостей.

3.1. Функціональні можливості

Автоматизована система має досить широкі функціональні можливості, а саме:

1. Пошук товарів: Користувачі зможуть використовувати автоматизовану систему для пошуку товарів на Amazon за різними критеріями, такими як категорія, ціновий діапазон тощо.

2. Аналіз товарів: Система надасть користувачам детальну інформацію про кожен товар, включаючи ціну, відгуки користувачів та характеристики товару.

3. Обчислення оптимальної ціни: Автоматизована система буде використовувати алгоритми для обчислення оптимальної ціни товару.

4. Збір активності користувачів: Система буде вести журнал активності користувачів, збираючи дані про їх пошукові запити та взаємодію з додатком.

5. Збереження історії пошуку товарів: Система буде зберігати пошукові запити користувачів та надавати можливість використати їх ще раз для повторного пошуку.

3.2. Архітектура системи

Архітектура системи може бути розбита на кілька компонентів, які взаємодіють між собою для забезпечення функціональності та ефективності системи. Основні компоненти архітектури моєї системи включають:

1. Користувацький інтерфейс: Це частина системи, з якою користувач взаємодіє. Вона забезпечує можливість вводу фільтрів пошуку, перегляду результатів та взаємодії з іншими функціями системи.

2. Керуючий модуль: Цей модуль відповідає за керування виконанням системи. Він обробляє вхідні дані від користувача, передає їх до відповідних модулів для обробки та координації дій. Керуючий модуль також забезпечує взаємодію з базою даних та зовнішніми сервісами, такими як Кеера API.

3. Модуль обробки даних: Цей модуль відповідає за обробку та аналіз даних. Він взаємодіє з Кеера API для отримання інформації про товари та ціни на Amazon. Модуль обробки даних також виконує фільтрацію та сортування результатів відповідно до введених користувачем фільтрів.

4. Модуль бази даних: Цей модуль відповідає за збереження та управління даними. Він використовує базу даних PostgreSQL для збереження історії пошуку користувачів. Модуль бази даних забезпечує доступ до даних та виконує операції читання та запису.

5. Логування активності: Цей компонент відповідає за запис активності користувачів у log-файл. Він реєструє запити до API, результати пошуку та інші події, що відбуваються в системі. Log-файл забезпечує можливість відстеження та аналізу активності користувачів для покращення системи та виявлення проблем.

Ці компоненти взаємодіють між собою за допомогою відповідних інтерфейсів та протоколів обміну даними. Користувацький інтерфейс передає дані до керуючого модуля, який координує дії та взаємодіє з модулем обробки

даних та модулем бази даних. Log-файл забезпечує запис активності, який може бути використаний для аналізу та відстеження подій у системі.

Ця архітектура забезпечує ефективну та надійну роботу системи аналізу ринку, забезпечуючи можливість вводу фільтрів, отримання даних з Amazon за допомогою Кеера API, збереження інформації в базі даних PostgreSQL та логування активності користувачів.

Роботу автоматизованої системи та структуру модулів можна представити у вигляді структурної схеми на рис.3.1.

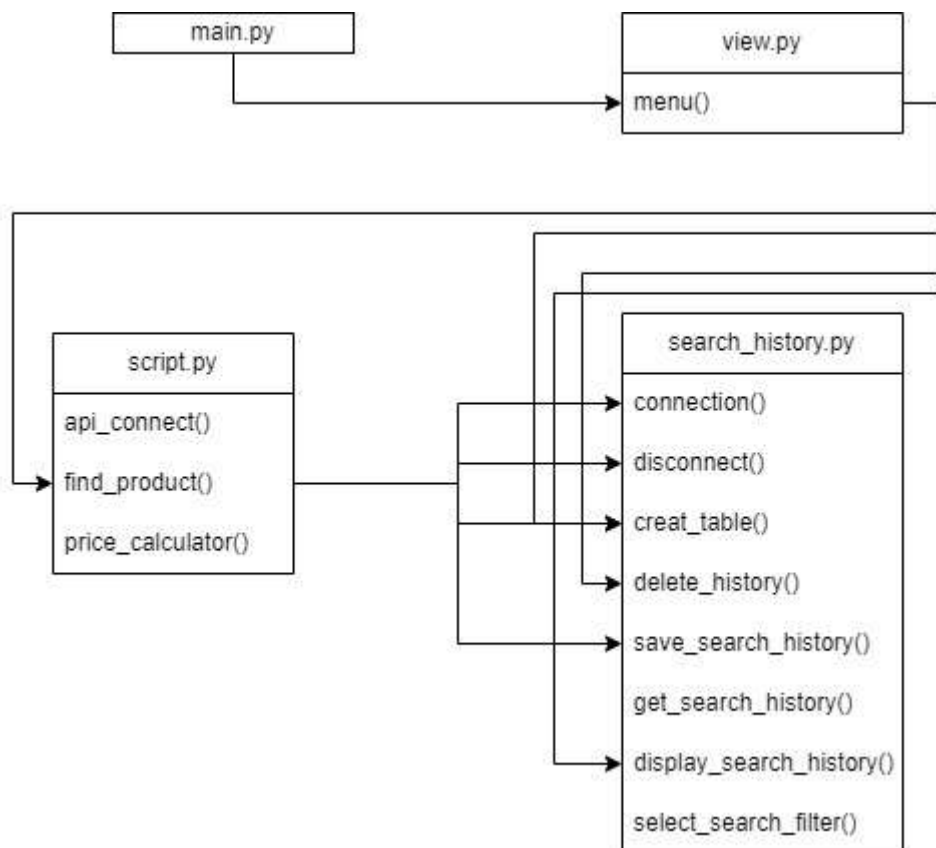


Рис.3.1. Структурна схема взаємодії модулів системи

3.3. Опис користувацького інтерфейсу

Користувацький інтерфейс є ключовим елементом, що дозволяє користувачеві зручно та ефективно взаємодіяти з системою, виконувати різні дії та отримувати необхідні результати.

Користувацький інтерфейс автоматизованої системи реалізований у вигляді меню у модулі view.py, що надає користувачеві можливість вибрати різні опції та взаємодіяти з системою. Меню складається з наступних пунктів:

1. Пошук та обчислення ціни
2. Подивитися історію пошуку
3. Пошук за існуючим фільтром
4. Видалити історію пошуку
5. Вихід

Щоб обрати опцію, користувач має ввести відповідний номер опції в консолі та натиснути «Enter». Розглянемо кожний з пунктів меню, його призначення та відповідну функцію, яку він виконує.

1. "Пошук та обчислення ціни": Користувач обирає цей пункт меню, введенням відповідного номера. Після цього, виконується функція `find_product` з модулю `script.py`, яка забезпечує пошук продуктів за введеними фільтрами і обчислення оптимальної ціни. Користувач отримує результат у вигляді виводу в консолі або відображення на екрані, залежно від налаштувань програми.

2. "Подивитися історію пошуку": Користувач обирає цей пункт, щоб переглянути історію своїх попередніх пошуків. Виконується функція `display_search_history`, яка отримує дані з бази даних та відображає їх у вигляді списку в консолі або на екрані.

3. "Пошук за існуючим фільтром": Користувач обирає цей пункт, щоб здійснити пошук продуктів за попередньо збереженими фільтрами з історії пошуку. Виконується функція `select_search_filter`, яка отримує введений користувачем номер фільтра та використовує його для здійснення пошуку. Результати виводяться у консолі або на екрані.

4. "Видалити історію пошуку": Користувач обирає цей пункт, щоб очистити всю історію пошуку. Виконується функція `delete_history`, яка видаляє

всі записи про пошук з бази даних. Користувач отримує підтвердження про успішне видалення.

5. "Вихід": Користувач обирає цей пункт, щоб завершити роботу з програмою. Відбувається виклик функції exit, яка завершує виконання програми і закриває користувацький інтерфейс.

Після виконання кожної з опцій меню користувач може обрати іншу функцію і так продовжується до тих пір, поки не буде обрана опція «5. Вихід». Таким чином, взаємодія з меню є простою та зрозумілою, що дозволяє зручно та ефективно користуватися системою.

Взаємодію користувача з автоматизованою системою та подальше виконання алгоритмів можна представити схемою на рис.3.2.

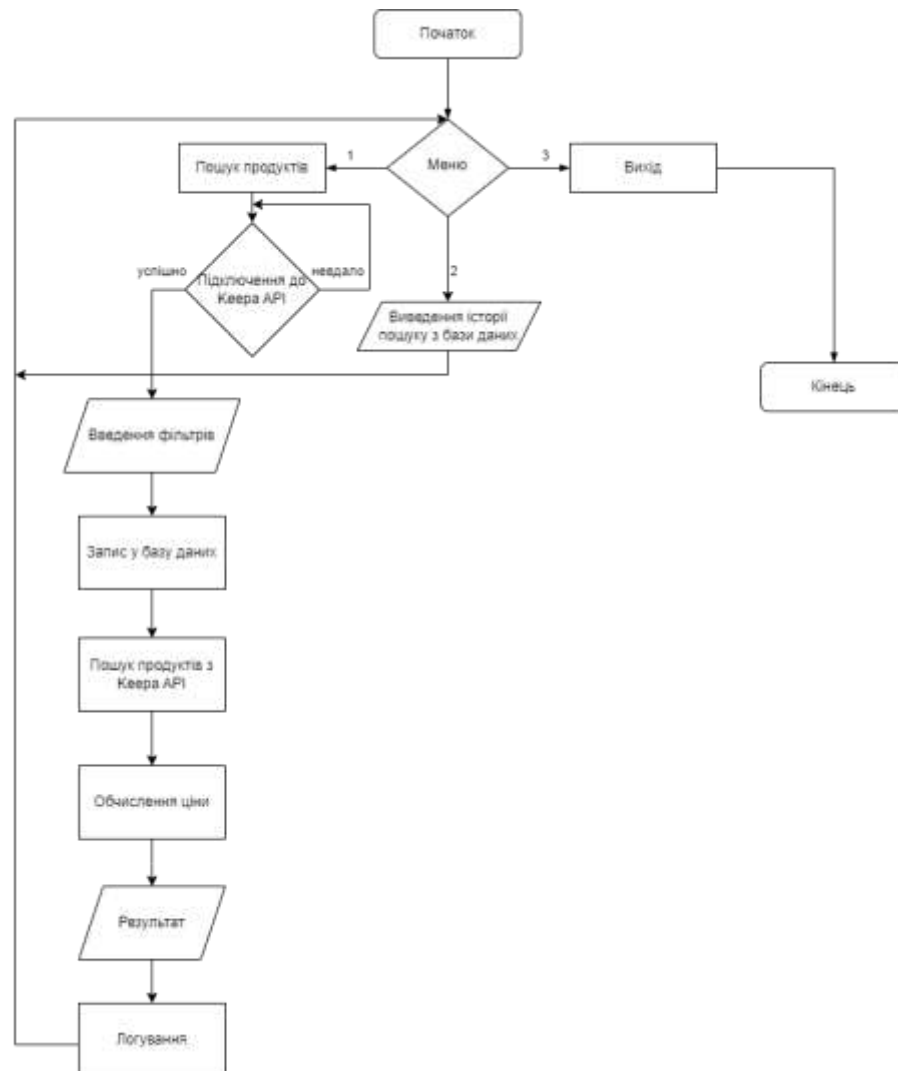


Рис.3.2. Взаємодія користувача з автоматизованою системою

3.4. Опис модуля пошуку товарів та обчислення ціни

Модуль пошуку товарів та обчислення ціни реалізований у файлі `script.py` і складається з трьох функцій: `api_connect`, `find_product` і `price_calculator`. Розглянемо кожну функцію детальніше:

1. `api_connect`: Ця функція відповідає за підключення до Кеера API за допомогою переданого ключа. Вона ініціалізує з'єднання з API та забезпечує аутентифікацію з використанням ключа. Цей крок необхідний для здійснення запитів до Кеера API і отримання даних про продукти.

2. `find_product`: Ця функція виконує алгоритм пошуку продуктів за заданими фільтрами, такими як ключове слово, мінімальна та максимальна ціна, мінімальна та максимальна кількість відгуків. Вона використовує з'єднання з Кеера API, отримує відповідні дані і сортує їх за кількістю продажів. Після цього функція виводить три знайдених товари, які відповідають заданим фільтрам. Цей процес дозволяє користувачеві знайти релевантні продукти за його вказаними критеріями.

3. `price_calculator`: Ця функція обчислює оптимальну ціну для товарів у заданому проміжку. Вона отримує список знайдених товарів з `find_product` і застосовує внутрішній алгоритм для визначення оптимальної ціни. Після обчислення оптимальної ціни для кожного товару, функція повертає результат, який може бути використаний для подальшого аналізу або прийняття рішень.

Алгоритм пошуку товарів у функції `find_product` можна зобразити у вигляді схеми алгоритму, показаної на рис.3.3.

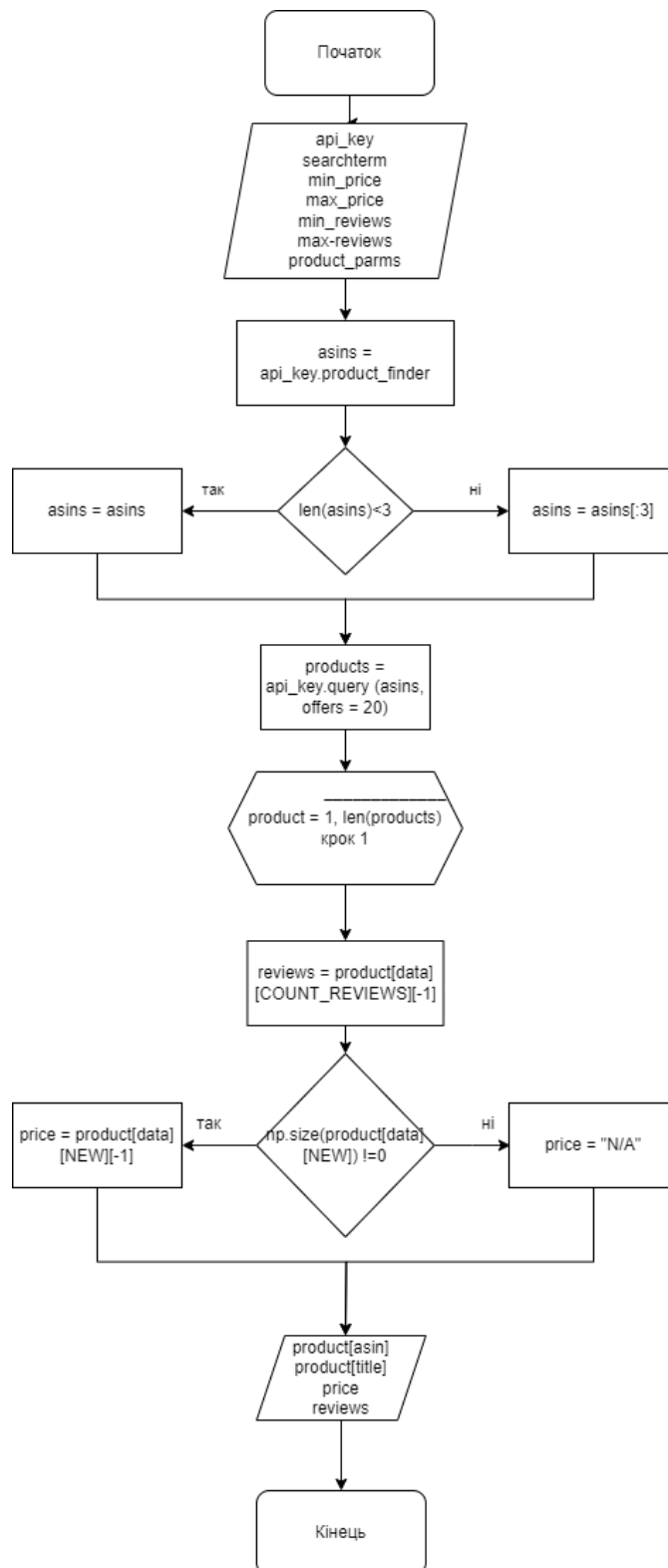


Рис.3.3. Схема алгоритму пошуку товарів за заданими фільтрами

Отже, цей модуль дозволяє здійснити пошук товарів за заданими фільтрами, відсортувати їх за кількістю продажів та обчислити оптимальну ціну. Використання Кеера API дозволяє отримати актуальну інформацію про

продукти, що допомагає зробити вибір на основі надійних даних. Функції модуля взаємодіють між собою, дозволяючи користувачеві здійснювати пошук, аналізувати результати і приймати обґрунтовані рішення щодо ціноутворення власних товарів.

3.5. Опис модуля бази даних

Модуль для взаємодії з базою даних реалізований у файлі `search_history.py` і складається з наступних функцій:

1. `connection`: Ця функція відповідає за підключення до бази даних PostgreSQL. Вона використовує встановлені параметри підключення, такі як хост, порт, ім'я бази даних, ім'я користувача та пароль. Після успішного підключення функція повертає об'єкт підключення, який використовується для виконання запитів до бази даних.

2. `disconnect`: Ця функція відповідає за роз'єднання з базою даних. Вона приймає об'єкт підключення і закриває з'єднання з базою даних, звільняючи ресурси.

3. `create_table`: Ця функція створює таблицю `"search_history"` у базі даних. Вона використовує SQL-запит для створення таблиці з відповідними колонками, які відповідають фільтрам пошуку, таким як `"search_term"`, `"min_price"`, `"max_price"`, `"min_reviews"` та `"max_reviews"`. Також можуть бути додані інші колонки, які відображають додаткову інформацію про пошук.

4. `delete_history`: Ця функція видаляє всю історію пошуку з таблиці `"search_history"`. Вона використовує SQL-запит `DROP TABLE` для видалення всіх записів з таблиці.

5. `save_search_history`: Ця функція зберігає історію пошуку в таблицю `"search_history"`. Вона приймає введені користувачем фільтри і виконує SQL-запит `INSERT` для додавання нового запису в таблицю.

6. `get_search_history`: Ця функція отримує всю історію пошуку з таблиці `"search_history"` за допомогою SQL-запиту `SELECT`. Вона повертає результат

запиту, який може бути використаний для подальшої обробки або відображення.

7. `display_search_history`: Ця функція виводить історію пошуку на екран. Вона приймає результат запиту `SELECT` і використовує цикл для перебору записів та виведення їх на екран користувачу.

8. `select_search_filter`: Ця функція обирає запис з таблиці `"search_history"` за введеним користувачем `ID` і використовує цей фільтр для виконання пошуку товарів. Вона використовує SQL-запит `SELECT` з умовою `WHERE` для вибору відповідного запису та використання його фільтрів для пошуку товарів.

Отже, цей модуль дозволяє зберігати історію пошуку, отримувати її, відображати на екрані, а також обирати попередні фільтри для подальшого використання. Використання бази даних PostgreSQL забезпечує збереження даних між сеансами роботи програми та забезпечує зручну роботу з історією пошуку.

3.6. Висновки до розділу

У цьому розділі було розглянуто архітектуру автоматизованої системи для аналізу ринку та обчислення ціни товарів. Архітектура системи була побудована на основі модульного підходу, де різні функціональні частини були розбиті на окремі модулі для кращої організації та зручності розробки.

Перший модуль, який було розглянуто, це модуль пошуку товарів та обчислення ціни. Він використовує Кеера API для отримання актуальних даних про продукти, та надає можливість користувачеві шукати товари за певними фільтрами, такими як ціна, кількість відгуків тощо. Також в цьому модулі є функція для обчислення оптимальної ціни для товарів у заданому діапазоні.

Другий модуль, який було розглянуто, це модуль для взаємодії з базою даних. Він використовує PostgreSQL для зберігання історії пошуку

користувача. Цей модуль дозволяє зберігати, видаляти та отримувати історію пошуку користувача. Також, він надає можливість обрати попередні фільтри для подальшого використання при пошуку товарів.

Усі модулі взаємодіють між собою, створюючи єдину систему для аналізу ринку та обчислення ціни товарів. Модуль пошуку товарів використовує дані з Кеера API, фільтри та параметри введені користувачем, а також модуль взаємодії з базою даних, щоб забезпечити збереження історії пошуку та використання попередніх фільтрів.

Також у цьому розділі було розглянуто користувацький інтерфейс автоматизованої системи. Інтерфейс реалізований у вигляді простого меню в консолі, яке містить кілька пунктів для вибору: пошук продуктів, перегляд історії пошуку, видалення історії пошуку, пошук за існуючим фільтром, вихід із програми.

Користувацький інтерфейс дозволяє користувачу зручно та ефективно взаємодіяти з системою, вибирати потрібні опції та отримувати необхідну інформацію. Він простий у використанні та надає можливість зручно керувати автоматизованою системою для аналізу ринку та обчислення цін.

В цілому, користувацький інтерфейс є важливою складовою частиною автоматизованої системи, яка дозволяє забезпечити зручну та ефективну взаємодію з користувачами.

Ця архітектура системи дозволяє зручно та ефективно здійснювати пошук товарів, обчислювати оптимальну ціну та зберігати історію пошуку для подальшого використання. Вона демонструє використання різних технологій та програмних засобів, таких як Python, Кеера API, PostgreSQL, для досягнення поставленої мети.

4. ТЕСТУВАННЯ СИСТЕМИ ТА ЇЇ ВЗАЄМОДІЯ З КОРИСТУВАЧЕМ

4.1. Тестування функції пошуку продуктів автоматизованої системи

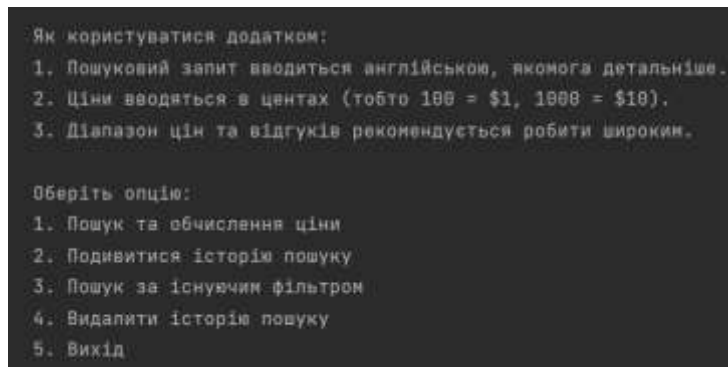
При тестуванні функції пошуку продуктів за фільтрами важливо переконатися, що вона працює належним чином та повертає очікувані результати. Для детального тестування, потрібно перевірити коректність результатів, роботу з різними фільтрами, тестування продуктивності та помилок.

Загальною метою тестування є переконатися, що функція пошуку продуктів за фільтрами працює надійно, точно та ефективно відповідно до очікуваних специфікацій та вимог.

1. Перевірка коректності результатів.

Для перевірки правильності результатів, задамо фільтри пошуку продуктів у створеній автоматизованій системі та порівняємо з даними, які виводяться у додатку Кеера.

Після запуску системи на екрані виводяться правила користування системою та меню (рис.3.1). Меню складається з пошуку та обчислення ціни, виведення історії пошуку, пошуку за існуючим фільтрами, видалення історії пошуку та виходу з програми.



```
Як користуватися додатком:  
1. Пошуковий запит вводиться англійською, якомога детальніше.  
2. Ціни вводяться в центах (тобто 100 = $1, 1000 = $10).  
3. Діапазон цін та відгуків рекомендується робити широким.  
  
Оберіть опцію:  
1. Пошук та обчислення ціни  
2. Подивитися історію пошуку  
3. Пошук за існуючим фільтром  
4. Видалити історію пошуку  
5. Вихід
```

Рис. 4.1. – Початок роботи системи

Для початку пошуку обираємо пункт меню «1» та вводимо наступні фільтри:

1. Ключове слово для пошуку продуктів: за вимогами, має бути написано англійською, а також має бути якомога детальнішим, щоб пошук відбувався точніше.

2. Мінімальну та максимальну ціну: максимальна ціна має бути більшою за мінімальну, а також бажано робити діапазон цін ширшим. Ціни мають бути введені у центах.

3. Мінімальну та максимальну кількість відгуків: максимальна кількість відгуків має бути більше за мінімальну, за вимогами, діапазон має бути ширшим.

У цей момент система зберігає введені фільтри у базу даних, історія активності користувача зберігається у log-файлі, а користувач бачить екран, як на рис.3.2.

```
Введіть ключове слово для пошуку продуктів:
teeth whitening strips
Введіть мінімальну ціну:
1000
Введіть максимальну ціну:
4200
Введіть мінімальну кількість відгуків:
100
Введіть максимальну кількість відгуків:
10000
```

Рис. 4.2. Введення фільтрів для тестування

Після цього система з'єднується з Кеера API для отримання даних про товари, сортує усі знайдені товари за кількістю продажів, виводить 3 перших товари із списку та друкує деталі про кожний продукт на екрані (рис.3.3). Деталі про продукт включають:

1. ASIN (Amazon Standard Identification Number): Стандартний ідентифікаційний номер Amazon. Він є вказівником на конкретний товар і

дозволяє користувачу легко знайти товар на Amazon, щоб подивитися більше деталей.

2. Назва: Назва товару, яка відповідає актуальній інформації на сторінці продукту.

3. Ціна: Поточна ціна товару у доларах.

4. Кількість відгуків: Поточна кількість відгуків на сторінці товару.



Рис.4.3. Результат пошуку за заданими фільтрами у системі

Тепер порівнюємо результат пошуку у створеній системі з результатом пошуку у функції «Product Finder» у додатку Кеера. Вводимо такі самі фільтри, як у створеній системі, тобто ціну, пошуковий запит, кількість відгуків (рис.3.4).

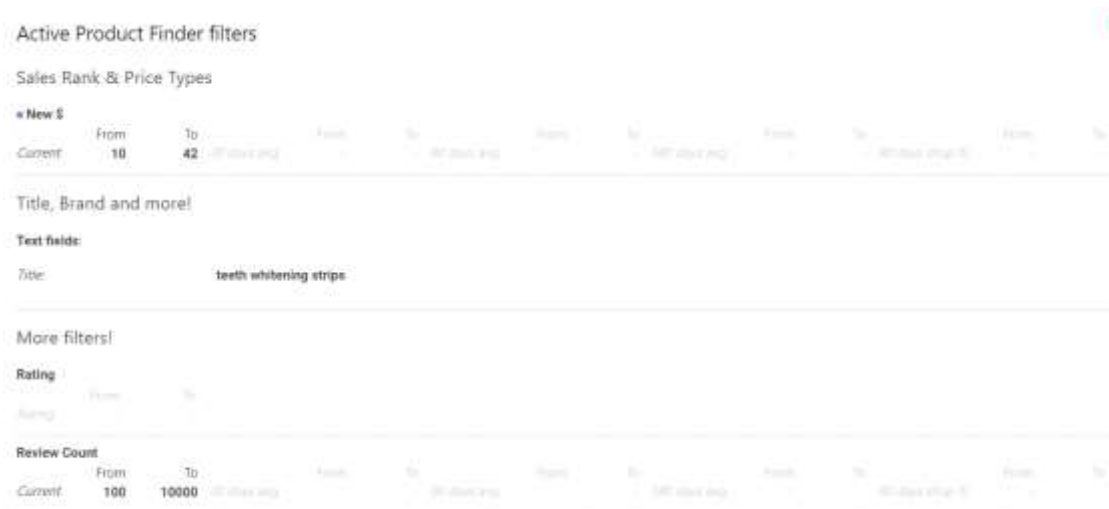


Рис.4.4. Введення фільтрів у додатку Кеера

Отримуємо результат пошуку, який так само відсортований за зростанням продажів (рис.3.5).

		● Reviews		● New	
Image	Title ▾	Review Co...	Current ▾		
	Colgate Max Fresh Toothpaste, Whitening Toothpaste with Mini...	166	\$ 10.61		
	Glorida Teeth Whitening Strips, Pack of 28 Strips (14...	4,685	\$ 19.99		
	Teeth Whitening Kit Gel Pen Strips - Hydrogen Carbamide Peroxide fo...	2,603	\$ 29.99		

Рис.4.5. Результат пошуку за заданими фільтрами в додатку Кеера

Результати пошуку у створеній системі та додатку Кеера збігаються, що доводить коректність роботи функції пошуку товарів за заданими фільтрами. Після завершення пошуку в автоматизованій системі активність користувача зберігається у log-файл, а також знову з'являється меню для вибору опції.

4.2. Тестування функції збереження історії пошуку та пошуку за збереженими фільтрами

Після того як користувач ввів фільтри для пошуку товару, вони одразу зберігаються у базу даних search_history, яка має відповідну таблицю та колонки з id, ASIN, назвою товару, мінімальною та максимальною ціною, мінімальною та максимальною кількістю відгуків. Для зручного управління базою даних рекомендується використовувати графічний інструмент PgAdmin 4 (рис.3.6), але найважливіші функції доступні в автоматизованій системі.

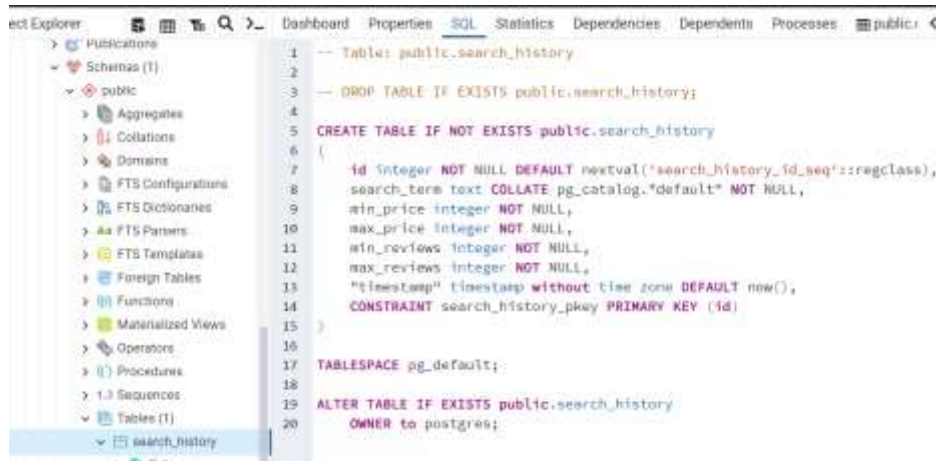


Рис.4.6. Робоче вікно графічного інструменту PgAdmin 4

Для того щоб подивитися історію пошуку в автоматизованій системі, потрібно обрати пункт «2» в меню, після чого буде виведена історія пошуку таким чином, що першою виводиться останні введені фільтри (рис.3.7). На екрані виводяться наступні параметри: id (унікальний ідентифікатор запису, який в даному випадку є порядковим номером кожного введенного фільтра); пошуковий запит, введений користувачем; мінімальна та максимальна ціна, введені у центах; мінімальна та максимальна кількість відгуків; дата та час запису фільтрів у базі даних.

```

5: wine gift ideas
Мінімальна / максимальна ціна: 1000 / 3000
Мінімальна / максимальна кількість відгуків: 10 / 1000 (2023-06-07 23:35:02.254436)
4: red short dress
Мінімальна / максимальна ціна: 1000 / 10000
Мінімальна / максимальна кількість відгуків: 100 / 5000 (2023-06-06 23:31:48.054731)
3: daily planner
Мінімальна / максимальна ціна: 1000 / 3000
Мінімальна / максимальна кількість відгуків: 100 / 1600 (2023-06-02 09:16:44.136385)
2: phone protector tether
Мінімальна / максимальна ціна: 10 / 30
Мінімальна / максимальна кількість відгуків: 100 / 30000 (2023-06-02 08:42:39.689617)
1: strips
Мінімальна / максимальна ціна: 10 / 20
Мінімальна / максимальна кількість відгуків: 100 / 200 (2023-05-31 01:12:43.035340)

```

Рис.4.7. Історія пошуку в автоматизованій системі

У таблиці search_history бази даних ці параметри записані у наступні колонки: id (унікальний ідентифікатор запису), search_term (пошуковий запит), min_price (мінімальна ціна), max_price (максимальна ціна), min_reviews

(мінімальна кількість відгуків), max_reviews (максимальна кількість відгуків), timestamp (дата та час запису фільтрів у базі даних).

Для тестування потрібно зафіксувати поточну історію пошуку в автоматизованій системі та в базі даних (рис.3.8).

Історія пошуку:

1: digital camera blue

Мінімальна / максимальна ціна: 100 / 500

Мінімальна / максимальна кількість відгуків: 10 / 100 (2023-06-09 07:27:09.272993)

2: digital 13 pro max case blue

Мінімальна / максимальна ціна: 800 / 2000

Мінімальна / максимальна кількість відгуків: 50 / 300 (2023-06-07 23:50:36.616968)

3: mini digital case green

Мінімальна / максимальна ціна: 2000 / 20000

Мінімальна / максимальна кількість відгуків: 50 / 100 (2023-06-07 23:52:05.355728)

4: mini gift black

Мінімальна / максимальна ціна: 1000 / 3000

Мінімальна / максимальна кількість відгуків: 10 / 300 (2023-06-07 23:36:20.506097)

5: mini gift black

Мінімальна / максимальна ціна: 1000 / 3000

Мінімальна / максимальна кількість відгуків: 10 / 1000 (2023-06-07 23:35:02.294636)

6: mini gift black

Мінімальна / максимальна ціна: 1000 / 3000

Мінімальна / максимальна кількість відгуків: 100 / 3000 (2023-06-06 23:32:02.956711)

7: daily planner

Мінімальна / максимальна ціна: 1000 / 3000

Мінімальна / максимальна кількість відгуків: 100 / 1000 (2023-06-02 09:56:46.126885)

8: phone protector white

Мінімальна / максимальна ціна: 10 / 30

Мінімальна / максимальна кількість відгуків: 100 / 3000 (2023-06-03 00:42:39.699617)

9: stylus

Мінімальна / максимальна ціна: 10 / 30

Мінімальна / максимальна кількість відгуків: 100 / 300 (2023-06-01 01:12:43.033348)

id	search_date	min_price	max_price	min_reviews	max_reviews	timestamp
(PK)	date	integer	integer	integer	integer	timestamp without time zone
1	2023-06-09 07:27:09.272993	100	500	10	100	2023-06-09 07:27:09.272993
2	2023-06-07 23:50:36.616968	800	2000	50	300	2023-06-07 23:50:36.616968
3	2023-06-07 23:52:05.355728	2000	20000	50	100	2023-06-07 23:52:05.355728
4	2023-06-07 23:36:20.506097	1000	3000	10	300	2023-06-07 23:36:20.506097
5	2023-06-07 23:35:02.294636	1000	3000	10	1000	2023-06-07 23:35:02.294636
6	2023-06-06 23:32:02.956711	1000	3000	100	3000	2023-06-06 23:32:02.956711
7	2023-06-02 09:56:46.126885	1000	3000	100	1000	2023-06-02 09:56:46.126885
8	2023-06-03 00:42:39.699617	10	30	100	3000	2023-06-03 00:42:39.699617
9	2023-06-01 01:12:43.033348	10	30	100	300	2023-06-01 01:12:43.033348

Рис.4.8. Поточна історія пошуку в автоматизованій системі та в базі даних

Після цього обираємо в меню системи опцію «1. Пошук та обчислення ціни» і виконуємо пошук за наступними фільтрами: пошуковий запит – «digital camera with autofocus», мінімальна ціна – «5000», максимальна ціна – «10000», мінімальна кількість відгуків – «150», максимальна кількість відгуків – «3000». Бачимо, що на екрані виводяться товари за заданими фільтрами та обчислена оптимальна ціна (рис.3.9).

100%[#####] 3/3 [00:06:00:00, 2.25s/it]

ASIN: B09JLYFZY8

Назва: Digital Camera, FHD 48.8 MP Vlogging Camera Mini Camera Kids Camera Video Camera Pocket Camera with

Ціна: 63.99

Кількість відгуків: 251

ASIN: B08GHFJ6L9

Назва: Digital Camera, Zosovic 48MP Autofocus Kids Camera with 32GB Card 1080P Video Camera with 16X Zoom,

Ціна: 59.99

Кількість відгуків: 284

ASIN: B08GLQXNGY

Назва: 4K Digital Camera for Photography and Video Autofocus 16X Digital Zoom, 48MP Vlogging Camera with 3

Ціна: 89.99

Кількість відгуків: 790

71.32

Рис.4.9. Виведення товарів за заданими фільтрами для тестування

Тепер потрібно обрати опцію «2. Подивитися історію пошуку» і порівняти з таблицею в базі даних search_history (рис.3.10). Можна побачити, що історія пошуку однакова, тобто ця функція автоматизованої системи працює справно.

10: digital camera with autofocus
Minimálna / maximálna cena: 3000 / 10000
Minimálna / maximálna skvelosť odgryku: 100 / 3000 (2023-06-09 08:02:25.791984)
9: camera case blue
Minimálna / maximálna cena: 300 / 800
Minimálna / maximálna skvelosť odgryku: 10 / 100 (2023-06-09 07:27:09.277292)
8: iphone 12 pro max case blue
Minimálna / maximálna cena: 900 / 2000
Minimálna / maximálna skvelosť odgryku: 50 / 300 (2023-06-07 21:08:34.416464)
7: anti aging eye cream
Minimálna / maximálna cena: 2000 / 10000
Minimálna / maximálna skvelosť odgryku: 50 / 300 (2023-06-07 21:02:03.105728)
6: wine gift ideas
Minimálna / maximálna cena: 3000 / 10000
Minimálna / maximálna skvelosť odgryku: 10 / 100 (2023-06-07 21:06:28.506897)
5: wine gift ideas
Minimálna / maximálna cena: 3000 / 10000
Minimálna / maximálna skvelosť odgryku: 10 / 1000 (2023-06-07 21:05:02.264436)
4: red shirt dress
Minimálna / maximálna cena: 3000 / 10000
Minimálna / maximálna skvelosť odgryku: 100 / 3000 (2023-06-06 20:21:46.954751)
3: daily planner
Minimálna / maximálna cena: 3000 / 10000
Minimálna / maximálna skvelosť odgryku: 100 / 1000 (2023-06-02 09:24:44.126285)
2: phone protector leather
Minimálna / maximálna cena: 30 / 30
Minimálna / maximálna skvelosť odgryku: 100 / 30000 (2023-06-03 08:42:29.689617)
1: utopit
Minimálna / maximálna cena: 20 / 20
Minimálna / maximálna skvelosť odgryku: 100 / 200 (2023-05-31 01:12:43.075140)

id	search_term	min_price	max_price	min_reviews	max_reviews	timestamp
[PK] integer	text	integer	integer	integer	integer	timestamp without time zone
1	utopit	20	20	100	200	2023-05-31 01:12:43.075140
2	phone protector leather	30	30	100	30000	2023-06-02 08:42:29.689617
3	daily planner	1000	3000	100	1000	2023-06-02 09:24:44.126285
4	red shirt dress	1000	10000	100	3000	2023-06-06 20:21:46.954751
5	wine gift ideas	1000	3000	10	1000	2023-06-07 21:05:02.264436
6	wine gift ideas	1000	3000	10	300	2023-06-07 21:06:28.506897
7	anti aging eye cream	2000	10000	50	300	2023-06-07 21:02:03.105728
8	iphone 12 pro max case blue	500	2000	50	300	2023-06-07 21:08:34.416464
9	camera case blue	100	800	10	100	2023-06-09 07:27:09.277292
10	digital camera with autofocus	3000	10000	130	3000	2023-06-09 08:02:25.791984

Рис.4.11. Порівняння оновленої історії пошуку

Якщо користувач ввів id запису, якого не існує, система покаже відповідну помилку (3.12), після чого користувач може ввести інший id.

Введіть ID вашого фільтра: 15
Запису з таким ID не знайдено

Як користуватися додатком:

1. Пошуковий запит вводиться англійською, якомога детальніше.

2. Ціни вводяться в центрах (тобто 100 = \$1, 1000 = \$10).

3. Діапазон цін та відгуків рекомендується робити широким.

Оберіть опцію:

1. Пошук та обчислення ціни

2. Подивитися історію пошуку

3. Пошук за існуючим фільтром

4. Видалити історію пошуку

5. Вихід

Рис.4.12. Результат введення неіснуючого id

В автоматизованій системі також присутня функція видалення історії пошуку, її можна запустити, якщо обрати опцію «4. Видалити історію пошуку». Якщо хоча б один запис вже був збережений, система видалить всю збережену історію та видасть повідомлення про успішне видалення записів (рис.3.13).

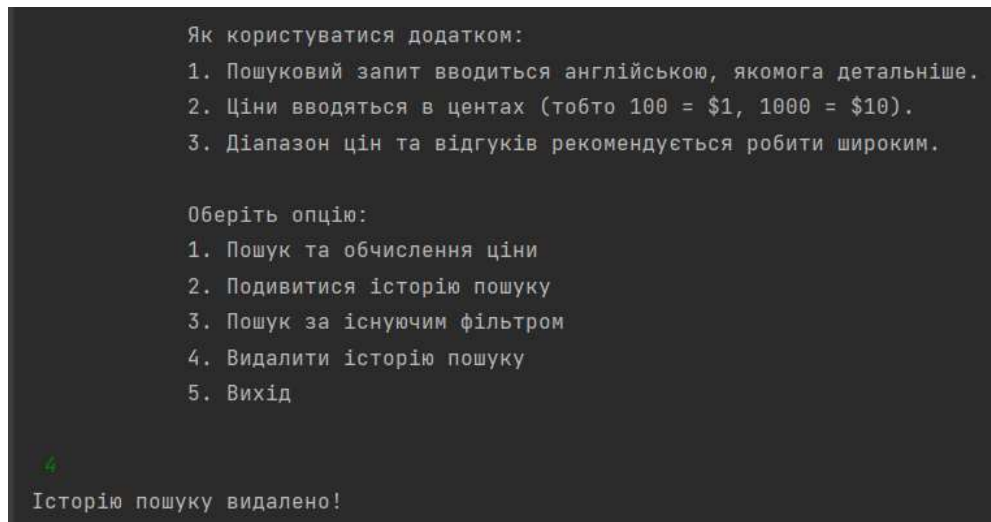


Рис.4.13. Повідомлення про видалення історії пошуку

Якщо після цього оновити таблицю бази даних у PgAdmin 4, записів не буде знайдено (рис 3.14).

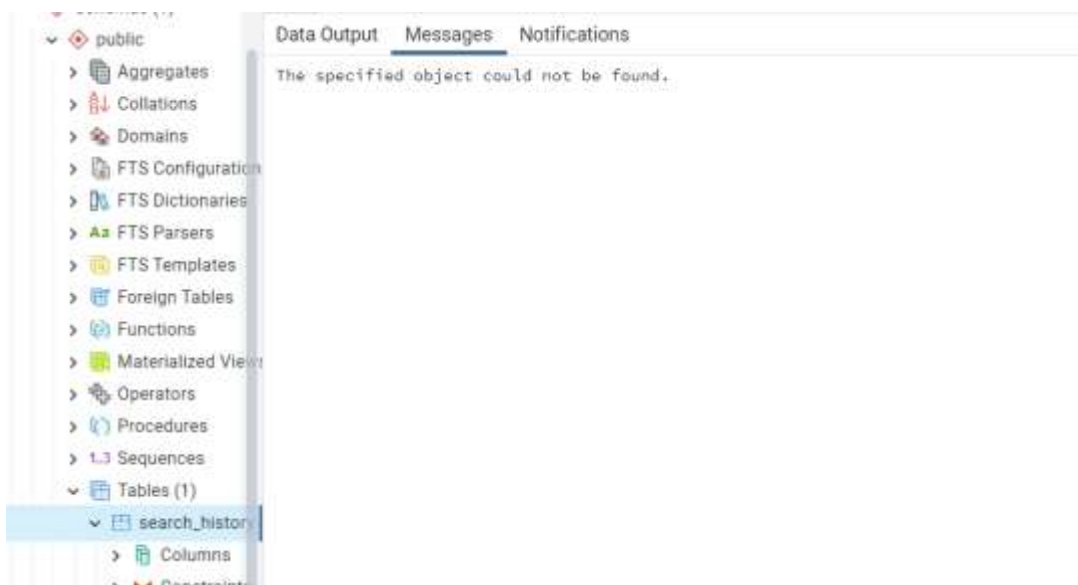


Рис.4.14. Відсутність даних у PgAdmin 4

4.3. Тестування функції логування активності користувача

Логування активності користувача відбувається у log-файлі app.log, яке створюється автоматично при першому запуску автоматизованої системи. Цей файл збирає усю інформацію про виконані операції, використані токени, виведений результат, помилки, тощо.

Щоб провести тестування функції логування, потрібно виконати всі можливі операції з меню системи.

1. Пошук та обчислення ціни

Вводимо наступні фільтри для пошуку товару: пошуковий запит – «white wedding dress», мінімальна ціна – «5000», максимальна ціна – «30000», мінімальна кількість відгуків – «100», максимальна кількість відгуків – «2000». У системі відбувається пошук товарів та обчислення ціни, результат виводиться на екран (3.15).

```
Введіть пошуковий запит: white wedding dress
Введіть мінімальну ціну: 5000
Введіть максимальну ціну: 30000
Введіть мінімальну кількість відгуків: 100
Введіть максимальну кількість відгуків: 2000
100%[#####] 3/3 (88:56-88:56, 1.24s/1s)
ASIN: B075N6V58G
Назва: 85 Pack of 85 Yard Bridal Solid Sheer Organza Fabric Bolt for Wedding Dress,Fashion, Crafts, Decorations Silky Shiny Organza 44"- White
Ціна: 115.0
Кількість відгуків: 1649

ASIN: B075N12CPS
Назва: 75 Pack of 75 Yard Bridal Solid Sheer Organza Fabric Bolt for Wedding Dress,Fashion, Crafts, Decorations Silky Shiny Organza 44"- White
Ціна: 105.0
Кількість відгуків: 1649

ASIN: B08BN4VFA8
Назва: 90 Pack of 90 Yard Bridal Solid Sheer Organza Fabric Bolt for Wedding Dress,Fashion, Crafts, Decorations Silky Shiny Organza 44"- White
Ціна: 120.0
Кількість відгуків: 1649

Витримана ціна: 113.33
```

Рис.4.15. Результат пошуку та обчислення ціни за запитом «white wedding dress»

Щоб подивитися активність користувача, потрібно відкрити файл app.log через додаток «Блокнот» або у середовищі розробки PyCharm. Можна побачити, що введені фільтри, а також процес зв'язку з Кеера API записані у файлі (рис.3.16). Читати такий файл дуже легко завдяки налаштованим повідомленням про відповідну дію.

```

2023-06-09 08:48:37,232 INFO Вибір опції: 1
2023-06-09 08:49:02,331 INFO Пошуковий запит: white wedding dress long
Мінімальна/максимальна ціна: 5000 / 30000
Мінімальна/максимальна кількість відгуків: 100 / 2000
2023-06-09 08:50:05,409 INFO Вибір опції: 1
2023-06-09 08:50:23,558 INFO Пошуковий запит: white wedding dress
Мінімальна/максимальна ціна: 5000 / 30000
Мінімальна/максимальна кількість відгуків: 100 / 2000
2023-06-09 08:50:23,734 INFO Історію пошуку збережено
2023-06-09 08:50:23,734 INFO Connecting to keera using key ending in to6uor
2023-06-09 08:50:23,941 DEBUG 0 tokens consumed
2023-06-09 08:50:23,945 INFO 300 tokens remain
2023-06-09 08:50:24,108 DEBUG 0 tokens consumed
2023-06-09 08:50:25,750 DEBUG 10 tokens consumed
2023-06-09 08:50:25,754 DEBUG Executing 3 item product query
2023-06-09 08:50:25,754 DEBUG Estimated time to complete 3 request(s) is 0.50 minutes
2023-06-09 08:50:25,755 DEBUG with a refill rate of 5 token(s) per minute
2023-06-09 08:50:25,991 DEBUG 0 tokens consumed
2023-06-09 08:50:32,556 DEBUG 18 tokens consumed

```

Рис.4.16. Запис введених фільтрів та зв'язку з Кеера API у файлі app.log

На рис.3.16 взаємодія користувача з автоматизованою системою виводиться після слова «INFO», а взаємодія системи з Кеера API починається з рядка «INFO Connecting to keera using key ending in to6uor». Кожна операція, задіяна у пошуку та аналізі товарів, використовує певну кількість токенів, що також записано у log-файлі. Наприклад, рядок «DEBUG 18 tokens consumed» свідчить про те, що було використано 18 токенів для виведення даних про кількість відгуків та ціну трьох товарів. Також присутній запис про початкову кількість токенів – «INFO 300 tokens remain», тобто на початку використання системи було 300 токенів. Запис «DEBUG Estimated time to complete 3 request(s) is 0.50 minutes with a refill rate of 5 token(s) per minute» означає, що для очікуваний час виконання операції 0.50 хвилини із швидкістю відновлення 5 токенів за хвилину.

Результат пошуку продуктів, інформація про них і обчислена оптимальна ціна також записуються у log-файл (рис.3.17). Формат представлення даних схожий на результат, який показує система користувачеві на екрані у робочому режимі. Усі рядки починаються із слова «INFO».

```

2023-06-09 08:50:32,572 INFO Product:
ASIN: B075HGY5GG
Price: 115.0
Reviews: 1649

2023-06-09 08:50:32,572 INFO Product:
ASIN: B075HJ2CPS
Price: 105.0
Reviews: 1649

2023-06-09 08:50:32,573 INFO Product:
ASIN: B08BK4YF6R
Price: 120.0
Reviews: 1649

2023-06-09 08:50:32,574 INFO Знайдені продукти:
340.0

```

Рис.4.17. Логування знайдених продуктів у файлі app.log

2. Перегляд історії пошуку

Для перегляду історії пошуку потрібно обрати опцію «2. Подивитися історію пошуку», після чого система виведе всі минулі записи про введені фільтри. Така сама інформація з'явиться у файлі app.log (рис.3.18), що свідчить про коректність роботи логування.

```

2023-06-09 09:21:56,999 INFO Показати історію пошуку:
2023-06-09 09:21:57,125 INFO [(1, 'white wedding dress', 5000, 30000, 100, 2000, datetime.datetime(2023, 6, 9, 8, 50, 23, 739077))]

```

Рис.4.18. Логування процесу виведення історії пошуку

Кожний запис починається із слова «INFO» та виводить порядковий номер запису, пошуковий запит, мінімальну та максимальну ціну, мінімальну та максимальну кількість відгуків, дату та час запису.

3. Пошук за існуючим фільтром

Для пошуку за існуючим фільтром, потрібно обрати опцію «3. Пошук за існуючим фільтром» та ввести Id запису. Після цього у log-файлі з'являться відповідні рядки із інформацією про обрану опцію в меню та Id запису (рис.3.19). Також знову з'являється інформація про наявні та витрачені токени, що демонструє зв'язок з Кеера API.

```
2023-06-09 09:32:29,241 INFO Вибір опції: 3
2023-06-09 09:32:29,241 INFO Виконати пошук за існуючими фільтрами:

2023-06-09 09:32:30,330 INFO Обрано запис №1
2023-06-09 09:32:30,394 INFO Connecting to keera using key ending in to6uop
2023-06-09 09:32:30,529 DEBUG 0 tokens consumed
2023-06-09 09:32:30,532 INFO 300 tokens remain
2023-06-09 09:32:30,681 DEBUG 0 tokens consumed
2023-06-09 09:32:31,142 DEBUG 10 tokens consumed
2023-06-09 09:32:31,146 DEBUG Executing 3 item product query
2023-06-09 09:32:31,146 DEBUG Estimated time to complete 3 request(s) is 0.50 minutes
2023-06-09 09:32:31,147 DEBUG with a refill rate of 5 token(s) per minute
2023-06-09 09:32:31,321 DEBUG 0 tokens consumed
2023-06-09 09:32:37,034 DEBUG 13 tokens consumed
```

Рис.4.19. Логування вибору існуючого фільтру для пошуку

Знайдені товари відображаються так само, як і при виконанні функції пошуку за введеними фільтрами, тобто є інформація про ASIN, назву товару, мінімальну та максимальну ціну, мінімальну та максимальну кількість відгуків.

4. Видалення історії пошуку та вихід із програми

Для видалення історії пошуку потрібно обрати опцію меню «4. Видалити історію пошуку», після чого на екрані з'явиться відповідне повідомлення про успішне видалення записів, а в log-файлі з'явиться рядок «INFO Історію видалено» (рис.3.20).

```
2023-06-09 09:39:34,276 INFO Видалення історії пошуку

2023-06-09 09:39:34,349 INFO Історію видалено
```

Рис.4.20. Логування функції видалення історії пошуку

Щоб завершити роботу та вийти з програми, потрібно обрати опцію меню «5. Вихід», після чого на екрані висвітиться повідомлення про успішне завершення роботи, це також буде записано в log-файлі (рис.3.21).

```
2023-06-09 09:39:39,790 INFO Вибір опції: 5
2023-06-09 09:39:39,790 INFO Вихід із програми
```

Рис.4.21. Логування функції виходу з програми

4.4. Перспективи розвитку та розширення функціоналу

Оскільки електронна комерція постійно розвивається, є багато простору для подальшого покращення автоматизованої системи для аналізу товарів та обчислення оптимальної ціни. Залежно від розвитку Amazon та інших платформ можна додати більше функціоналу. Нижче наведено кілька перспектив розвитку цієї системи:

1. Розширення функціональності: Система може бути розширена для підтримки додаткових функцій та можливостей. Наприклад, додаткові алгоритми аналізу даних можуть бути впроваджені для надання користувачам більш детальної інформації про товари, таких як прогнозування змін цін, порівняння товарів, рекомендації тощо.

2. Розширення підтримуваних платформ: Початково система розроблена для Google Chrome, але може бути розширена для підтримки інших популярних веб-браузерів, таких як Mozilla Firefox, Microsoft Edge і Safari. Це дозволить більшій кількості користувачів використовувати додаток.

3. Покращення інтерфейсу користувача: Інтерфейс користувача може бути покращений, забезпечуючи більш зручну та інтуїтивно зрозумілу навігацію, покращену візуалізацію даних та більші можливості налаштування для користувачів.

4. Інтеграція з іншими платформами та сервісами: Систему можна розширити для інтеграції з іншими платформами та сервісами, такими як соціальні медіа, електронні комерційні платформи, споживчі додатки тощо. Це

може дозволити збирати більше даних про користувачів та розширити можливості аналізу товарів.

5. Мобільний додаток: Систему можна розробити в мобільний додаток, що дозволить користувачам отримувати доступ до функціональності системи на своїх смартфонах та планшетах. Це зробить додаток більш доступним та зручним для користувачів.

6. Підтримка додаткових мов та регіонів: Систему можна розширити для підтримки додаткових мов та регіонів. Це дозволить користувачам з різних країн отримувати інформацію про товари у своїй рідній мові та з контекстом, відповідним їхнім регіональним потребам.

7. Покращення безпеки та захисту: Враховуючи, що система збирає та обробляє дані користувачів, важливо покласти особливу увагу на безпеку та захист інформації. Систему можна розширити для використання додаткових заходів безпеки, таких як шифрування даних, автентифікація користувачів та моніторинг зловмисної діяльності.

Ці перспективи розвитку відкривають широкі можливості для подальшого вдосконалення автоматизованої системи та її адаптації до змінних потреб користувачів та ринку. Важливо також спиратися на потреби користувачів та постійно обробляти відгуки, які допоможуть покращити усі вбудовані функції.

4.5. Висновки до розділу

Після проведення тестування різних функцій автоматизованої системи, включаючи пошук товарів та обчислення ціни, перегляд існуючих фільтрів, пошук за існуючими фільтрами з історії пошуку, видалення історії та вихід з програми, а також логування кожної з цих функцій в log-файлі, можна зробити наступні висновки:

1. Функція пошуку товарів та обчислення ціни була успішно протестована і продемонструвала свою працездатність. Під час тестування

було введено різні параметри пошуку, такі як ключові слова, діапазони цін та кількість відгуків. Система здійснювала пошук відповідних продуктів згідно з заданими фільтрами та обчислювала оптимальну ціну на основі цих результатів.

2. Функція перегляду існуючих фільтрів також була протестована та виконала свої завдання. Вона надавала користувачу можливість переглядати збережені фільтри з бази даних та використовувати їх для подальшого пошуку товарів.

3. Функція пошуку за існуючими фільтрами з історії пошуку була успішно протестована та показала свою ефективність. Вона дозволяла користувачу вибрати один зі збережених фільтрів з історії та використовувати його для пошуку нових товарів.

4. Функція видалення історії виконувала свою роль і успішно видаляла записи про пошукову історію з бази даних. Під час тестування було перевірено, що після видалення історії вона більше не відображалася у системі.

5. Функція виходу з програми була успішно протестована. Після виклику цієї функції програма коректно завершувалася та виходила з режиму роботи.

6. Функція логування кожної з цих функцій в log-файлі була реалізована і успішно протестована. Кожна активність користувача, така як введення фільтрів, пошук товарів та інші дії, реєструвалася у log-файлі з використанням рівнів INFO та DEBUG, що дозволяло зберігати історію активності.

В результаті тестування можна зробити висновок, що всі функції автоматизованої системи, включаючи функцію пошуку товарів та обчислення ціни, перегляд існуючих фільтрів, пошук за існуючими фільтрами з історії пошуку, видалення історії та вихід з програми, працюють належним чином і задовольняють вимоги специфікації проєкту. Також було підтверджено, що

функція логування активності користувача забезпечує збереження історії дій для подальшого аналізу та відлагодження.

Ці функції дозволяють користувачеві зручно та ефективно взаємодіяти з системою, здійснювати пошук, перегляд та аналіз ринку на основі введених фільтрів.

Також підрозділ, що описує перспективи розвитку автоматизованої системи для аналізу товарів та обчислення оптимальної ціни, демонструє потенціал системи для подальшого розширення та покращення. Розглянуті перспективи включають розширення функціональності, підтримку нових платформ та браузерів, покращення інтерфейсу користувача, інтеграцію з іншими платформами та сервісами, розробку мобільного додатку, підтримку додаткових мов та регіонів, а також покращення безпеки та захисту.

Розширення функціональності системи може включати додаткові алгоритми аналізу даних, що дозволять користувачам отримувати більш детальну інформацію про товари та забезпечать додаткові можливості, такі як прогнозування змін цін або порівняння товарів. Розширення підтримки платформ і браузерів розширить аудиторію системи та забезпечить більш широкий доступ до функціональності.

Покращення інтерфейсу користувача сприятиме зручності використання системи та включатиме елементи, такі як зручна навігація, візуалізація даних та можливості налаштування. Інтеграція з іншими платформами та сервісами може забезпечити більше даних для аналізу та розширити можливості системи. Розробка мобільного додатку дозволить користувачам отримувати доступ до системи на своїх смартфонах та планшетах, забезпечуючи зручність використання.

Підтримка додаткових мов та регіонів дозволить системі надати інформацію про товари в рідній мові користувачів та враховувати їхні регіональні потреби. Покращення безпеки та захисту даних включатиме використання шифрування, автентифікації користувачів та моніторингу

зловмисної діяльності, що забезпечить конфіденційність та безпеку інформації користувачів.

Загалом, перспективи розвитку автоматизованої системи для аналізу товарів та обчислення оптимальної ціни відкривають широкий простір для розширення функціональності, покращення інтерфейсу та забезпечення зручного та безпечного використання системи. Розвиток цих перспектив може зробити систему ще більш привабливою та корисною для користувачів у сфері електронної комерції.

					ІАЛЦ.466538.004 ПЗ	Лист
						57
Зм	Лист	№ докум.	Підп.	Дата		

ВИСНОВКИ

У ході дипломного проєкту була розроблена та побудована модель автоматизованої системи, яка має значний потенціал у сфері аналізу товарів на Amazon та обчислення оптимальної ціни. Система була створена з використанням сучасних технологій та програмних засобів, таких як HTML, CSS, JavaScript, React та Amazon API.

Основною метою системи є надання користувачам зручного та ефективного інструменту для аналізу ринку на Amazon та прийняття обґрунтованих рішень щодо ціноутворення. Система здатна збирати дані про товари з використанням Amazon API, проводити аналіз цін, оцінювати конкурентну ситуацію та рекомендувати оптимальну ціну для заданого товару.

Окрім аналізу товарів, система також здатна логувати активність користувачів та збирати статистику їх уподобань. Це дозволяє надавати персоналізовані рекомендації користувачам на основі їхньої попередньої активності та уподобань.

Для забезпечення високої продуктивності та безпеки, система використовує реляційну базу даних, таку як PostgreSQL, для зберігання та керування даними про користувачів, активність та уподобання. База даних дозволяє ефективно зберігати, опрацьовувати та виконувати запити до великого обсягу даних.

З метою забезпечення зручного та інтуїтивно зрозумілого інтерфейсу для користувачів, використання фреймворку React дозволяє будувати компонентну структуру додатку та легко управляти станом і відображенням даних. Комбінація HTML, CSS та JavaScript дозволяє створювати привабливі та респонсивні користувацькі інтерфейси, що покращують взаємодію користувача з системою.

Під час розробки було використано Amazon API для отримання доступу до даних про товари на Amazon. Це дозволяє системі отримувати актуальну інформацію про товари, їхні властивості та ціни. Використання Amazon API спрощує процес отримання та оновлення даних, що робить систему більш ефективною та актуальною.

Завдяки розробленій автоматизованій системі користувачі отримують зручний та надійний інструмент для аналізу ринку та оптимізації ціноутворення на Amazon. Система забезпечує збір та аналіз інформації, персоналізовані рекомендації та зберігання активності користувачів. Вона відповідає сучасним стандартам програмування та дизайну, а також забезпечує гнучкість та розширюваність для подальшого розвитку та покращення.

Загалом, розроблена автоматизована система має значний потенціал для використання в сфері електронної комерції та аналізу ринку. Її функціональність, продуктивність, безпека та зручний інтерфейс роблять її привабливим інструментом для підприємств, продавців та споживачів. З урахуванням потенціалу подальшого розвитку та вдосконалення, автоматизована система може стати важливим фактором у поліпшенні ефективності та успішності бізнесу на платформі Amazon.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Amazon API Documentation (Офіційна документація Amazon API). – 2023. – Режим доступу: <https://docs.aws.amazon.com/apigateway/index.html>.
2. Копес Ф. «Поступове знайомство з структурами даних: як працюють масиви». – freeCodeCamp, 2021.
3. Ніксон Р. «Вивчення PHP, MySQL та JavaScript: з використанням jQuery, CSS та HTML5». – O'Reilly Media, 2014.
4. Офіційна документація Python. – 2023. – Режим доступу: <https://www.python.org/doc/>.
5. Офіційна документація Кеера API. – 2023. – Режим доступу: <https://keepa.com/#!api>.
6. Дем'яненко В.Ю. «Програмні засоби створення та ведення баз даних». – Фінанси та Статистика, 1984.
7. Дженінгс, Ф. – «Практична передача даних. Методи, мережі і протоколи». – Мир, 1989.
8. Клименко Р. – «Веб-мастеринг на 100%». – Пітер, 2023.
9. Корячко, В.П. – «Корпоративні мережі. Технології, протоколи, алгоритми». – Горяча лінія – Телеком, 2021.
10. Рігс С. – «Адміністрування PostgreSQL 9. Книга рецептів». - ДМК Прес, 2018.
11. Хаббард Дж. – «Автоматизоване проєктування баз даних». – Мир, 1984.
12. Кормен Т. – «Алгоритми. Ввідний курс». – Діалектика, 2016.
13. Потопахін, В. – «Мистецтво алгоритмізації». – ДМК Прес, 2011.
14. Самерфілд, М. – «Програмування на Python 3. Детальне керівництво». – Символ-плюс, 2008.
15. Форсьє Дж. – «Розробка веб-додатків на Python». – Символ-плюс, 2012.

					ІАЛЦ.466538.004 ПЗ	Лист 60
Зм	Лист	№ докум.	Підп.	Дата		

16. Стоунз М. – «PostgreSQL. Основи». – Символ-плюс, 2002.
17. Уорслі Дж. – «PostgreSQL. Для професіоналів». – Пітер, 2002.
18. Харіс А. – «Збільшіть продажі електронної комерції і здобудете більше грошей: три сотні порад для підвищення ставок конверсії та створення потенційних клієнтів». – Kindle Store, 2014.
19. Стоун Б., Ларкін П. – «Все магазин: Джеф Безос та Ера Амазонки». – Hachette Audio, 2013.
20. Офіційна документація PostgreSQL. – 2016. – Режим доступу: <https://postgrespro.ru/docs/postgresql>.