

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

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

«На правах рукопису»
УДК 004.8:004.738.5

«До захисту допущено»

Завідувач кафедри

_____ Едуард ЖАРІКОВ

« ____ » _____ 2025 р.

Магістерська дисертація

на здобуття ступеня магістра

**за освітньо-професійною програмою «Інженерія програмного
забезпечення інформаційних систем»**

зі спеціальності 121 «Інженерія програмного забезпечення»

**на тему: «Програмний засіб для інтелектуального аналізу SEO-
параметрів вебресурсів та автоматичної генерації рекомендацій»**

Виконав (-ла):

студент (-ка) II курсу, групи ІІІ-341мп

Вакарчук Каміла Андріївна _____

Керівник:

Науковий керівник: доцент

Полупан Юлія Вікторівна _____

Рецензент:

доцент кафедри ІСТ, к.т.н., доц.,

Шимкович Володимир Миколайович _____

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних посилань.
Студент (-ка) _____

Київ – 2025 року

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Рівень вищої освіти – другий (магістерський)

Спеціальність – 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма «Інженерія програмного забезпечення інформаційних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Едуард ЖАРІКОВ

«__» _____ 2025р.

**ЗАВДАННЯ
на магістерську дисертацію студенту**

Вакарчук Камілі Андріївні

1. Тема дисертації «Програмний засіб для інтелектуального аналізу SEO-параметрів вебресурсів та автоматичної генерації рекомендацій», науковий керівник дисертації Полупан Юлія Вікторівна, доцент, затверджені наказом по університету від «06» листопада 2025 р. № 4842-с
2. Термін подання студентом дисертації «15» грудня 2025 р.
3. Об'єкт дослідження – процес комплексної SEO-оптимізації вебресурсів на основі аналізу їхніх технічних та контентних SEO-показників.
4. Предмет дослідження – моделі, методи та алгоритми інтелектуального аналізу SEO-параметрів вебресурсів і формування автоматизованих рекомендацій для їх оптимізації.
5. Перелік завдань, які потрібно розробити – вивчити стан досліджень методів та програмних засобів SEO-аналізу вебресурсів; розробити модель та алгоритми інтелектуального аналізу SEO-параметрів; створити програмний засіб для автоматичного збору даних, їх обробки та генерації рекомендацій; оцінити ефективність розробленого програмного засобу на прикладі реальних вебресурсів.
6. Орієнтовний перелік графічного (ілюстративного) матеріалу – 4 плакати
7. Орієнтовний перелік публікацій – одна публікація

8. Консультанти розділів дисертації

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

9. Дата видачі завдання «1» вересня 2025 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання	Примітка
1	Видача завдання	01.09.2025	
2	Аналіз існуючих методів та програмних засобів SEO-аналізу шляхом огляду літератури	02.09.2025 – 15.09.2025	
3	Вибір моделі/алгоритму для інтелектуального аналізу SEO-параметрів	16.09.2025 – 22.09.2025	
4	Розробка та удосконалення алгоритмів і методів аналізу	23.09.2025 – 10.10.2025	
5	Проектування програмного засобу	11.10.2025 – 20.10.2025	
6	Реалізація програмного засобу	21.10.2025 – 10.11.2025	
7	Виконання експериментальних досліджень	11.11.2025 – 18.11.2025	
8	Оформлення пояснювальної записки	19.11.2025 – 25.11.2025	
9	Подання дисертації на попередній захист	26.11.2025	
10	Подання дисертації на захист	15.12.2025	

Студент

Каміла ВАКАРЧУК

Науковий керівник

Юлія ПОЛУПАН

РЕФЕРАТ

Розмір пояснювальної записки – 145 аркушів, містить 13 ілюстрацій, 22 таблиці, 5 додатків, 26 посилань на джерела.

Актуальність теми. У роботі розглянуто проблему інтелектуального аналізу SEO-параметрів вебресурсів та автоматичної генерації рекомендацій для їх оптимізації. SEO (Search Engine Optimization) є ключовим елементом просування вебсайтів у пошукових системах, і від правильного налаштування параметрів залежить ефективність роботи вебресурсу. Виявлено потребу у створенні програмного засобу, який дозволяє автоматизувати процес збору, аналізу та оцінки SEO-параметрів, а також надає рекомендації для підвищення видимості сайту в пошукових системах.

Мета дослідження. Метою роботи є розробка програмного засобу для інтелектуального аналізу SEO-параметрів вебресурсів та автоматичної генерації пріоритезованих рекомендацій щодо їх оптимізації.

Об'єкт дослідження: процес комплексної SEO-оптимізації вебресурсів на основі технічних та контентних SEO-показників.

Предмет дослідження: методи, моделі та алгоритми інтелектуального аналізу SEO-показників вебресурсів і формування автоматизованих рекомендацій щодо їх оптимізації.

Для реалізації поставленої мети **сформульовані наступні завдання:**

- провести огляд наявних програмних рішень для SEO-аудиту та оцінити їх переваги й недоліки;
- розробити методи автоматизованого збору, обробки та оцінки SEO-параметрів;
- розробити удосконалені алгоритми автоматичної генерації пріоритезованих рекомендацій щодо SEO-оптимізації для досліджуваного вебресурсу;

- створити програмний засіб для автоматичного збору даних, їх обробки та генерації рекомендацій;
- провести тестування програмного засобу та оцінити ефективність його роботи.

Наукова новизна результатів магістерської дисертації полягає в розробці програмного рішення, яке забезпечує автоматизоване формування пріоритизованих рекомендацій щодо оптимізації сайтів. Результат досягнутий шляхом розробки методики кластеризації SEO-показників, що дає змогу автоматично визначати найбільш значущі показники на основі вагових коефіцієнтів та поєднувати їх з величиною відставання сайту від значень ТОП-конкурентів, формуючи індивідуальні пріоритети для покращення ранжування.

Практичне значення. Удосконалений програмний засіб може бути застосований власниками вебресурсів, маркетологами та SEO-фахівцями для автоматизації аналізу SEO-параметрів та підвищення ефективності просування сайтів.

Зв'язок з науковими програмами, планами, темами. Робота виконувалась на кафедрі інформатики та програмної інженерії Національного технічного університету України "Київський політехнічний інститут імені Ігоря Сікорського".

Апробація. Наукові положення дисертації пройшли апробацію на VIII Міжнародній науково-практичній конференції молодих вчених та студентів «Інженерія програмного забезпечення і передові інформаційні технології (SoftTech-2025)».

Публікації. Наукові положення дисертації опубліковані в:

– Вакарчук К.А., Полупан Ю.В. Програмний засіб для видачі рекомендацій з вирішення SEO-завдань інтернет-ресурсу. Матеріали VIII Міжнародної науково-практичної конференції молодих вчених та студентів «Інженерія програмного забезпечення і передові інформаційні технології (SoftTech-2025)». Секція кафедри інформатики та програмної інженерії. 13-15 травня 2025 р., Київ, Україна.

Ключові слова: SEO, ВЕБРЕСУРС, ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ, АВТОМАТИЧНЕ ГЕНЕРУВАННЯ РЕКОМЕНДАЦІЙ, ПРОГРАМНИЙ ЗАСІБ.

ABSTRACT

Explanatory note size – 145 pages, contains 13 illustrations, 22 tables, 5 applications, 26 references.

Topicality. This work addresses the problem of intelligent analysis of SEO parameters of web resources and automatic generation of recommendations for their optimization. Search Engine Optimization (SEO) is a key component of promoting websites in search engines, and the correct configuration of SEO parameters directly affects the efficiency and visibility of a web resource. There is a clear need for a software tool capable of automating the processes of collecting, analyzing, and evaluating SEO parameters, as well as generating recommendations aimed at improving a website's performance in search engine rankings.

The aim of the study. The aim of the study is to develop a software tool for intelligent analysis of SEO parameters of web resources and automatic generation of prioritized recommendations for their optimization.

The object of research: the process of comprehensive SEO optimization of web resources based on technical, content-related, and external parameters.

The subject of research: methods, models, and algorithms of intelligent analysis of SEO indicators of web resources and the generation of automated recommendations for their optimization.

To achieve this goal, the **following tasks** were formulated:

- to review existing software solutions for SEO auditing and evaluate their advantages and limitations;
- to develop methods for automated collection, processing, and assessment of SEO parameters;
- to design improved algorithms for the automatic generation of prioritized SEO optimization recommendations for the analyzed web resource;

- to develop a software tool for automated data collection, processing, and recommendation generation;
- to perform testing of the software tool and evaluate its effectiveness.

The scientific novelty of the results of the master's thesis lies in the development of a software solution that provides automated generation of prioritized recommendations for website optimization. The result was achieved by designing a methodology for clustering SEO indicators, which enables the automatic identification of the most significant metrics based on weight coefficients and combining them with the degree of the website's lag behind TOP competitors. This allows for the formation of individualized priorities aimed at improving search ranking performance.

The practical value of the improved software tool can be used by website owners, digital marketers, and SEO specialists to automate the analysis of SEO parameters and enhance the effectiveness of website promotion.

Relationship with working with scientific programs, plans, topics. Work was performed at the Department of Computer Science and Software Engineering of the National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute».

Approbation. The scientific provisions of the dissertation were tested at the VIII International Scientific and Practical Conference of Young Scientists and Students «Software Engineering and Advanced Information Technologies (SoftTech-2025)».

Publications. The scientific provisions of the dissertation were published in:

- Vakarchuk K.A., Polupan Y.V. Software tool for issuing recommendations for solving SEO tasks of an Internet resource. Proceedings of the VIII International Scientific and Practical Conference of Young Scientists and Students «Software Engineering and Advanced Information Technologies (SoftTech-2025)». Department of Computer Science and Software Engineering. May 13-15, 2025, Kyiv, Ukraine.

Keywords: SEO, WEB RESOURCE, INTELLIGENT ANALYSIS, AUTOMATIC GENERATION OF RECOMMENDATIONS, SOFTWARE TOOL.

ЗМІСТ

ЗМІСТ	10
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	12
ВСТУП.....	13
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ	15
1.1 Опис предметної області SEO-аналізу вебресурсів	15
1.2 Аналіз існуючих рішень для SEO-аудиту	18
1.2.1 Огляд інструмента 1	22
1.2.2 Огляд інструмента 2	23
1.3 Постановка задачі створення програмного засобу	25
Висновки до розділу	27
2 ТЕОРЕТИЧНІ ОСНОВИ ІНТЕЛЕКТУАЛЬНОГО SEO-АНАЛІЗУ	29
2.1 Основні SEO-показники та їх значущість	29
2.2 Методи збору та оцінки SEO-параметрів.....	33
2.3 Методи машинного аналізу	39
2.4 Метод множинної класифікації.....	46
2.5 Алгоритми формування рекомендацій	51
2.6 NLP-аналіз контенту.....	53
Висновки до розділу	57
3 РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАСОБУ	58
3.1 Архітектура програмного засобу	58
3.2 Вибір технологій і середовища розробки.....	61
3.3 Опис структурних модулів системи	63
3.4 Використовувані API та інтерфейси для збору SEO-параметрів.....	66
3.5 Інтерфейс користувача та логіка взаємодії	70
3.6 Експериментальні дослідження роботи системи	74
3.7 Моделі для роботи з сайтами конкурентами	80
Висновки до розділу	82
4 АНАЛІЗ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	84
4.1 Методи тестування програмного засобу	84
4.2 Функціональне тестування модулів.....	86
4.3 Нефункціональне тестування (швидкість, стійкість, навантаження).....	88

	11
4.4 Аналіз результатів тестування.....	90
Висновки до розділу	92
МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЄКТУ	96
5.1 Опис ідеї стартап-проєкту	96
5.2 Технологічний аудит ідеї проєкту	98
5.3 Аналіз ринкових можливостей запуску стартапу	99
5.4 Розроблення ринкової стратегії стартапу	104
5.5 Розроблення маркетингової програми стартап-проєкту	106
Висновки до розділу	110
ВИСНОВКИ	111
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	113
ДОДАТОК А: Лістинг коду	115
ДОДАТОК Б: Клас-діаграма.....	140
ДОДАТОК В: Діаграми послідовності.....	141
ДОДАТОК Г: Діаграма компонентів	142
ДОДАТОК Д: Інструкція користувача.....	143

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

SEO (Search Engine Optimization) – оптимізація вебресурсів для підвищення їхньої видимості у пошукових системах шляхом покращення внутрішніх та зовнішніх параметрів сайту;

HTML (HyperText Markup Language) – мова розмітки гіпертекстових документів;

H1, H2, ... Hn – заголовки різного рівня у структурі вебсторінки;

Canonical – елемент HTML, що вказує пошуковим системам на канонічну версію вебсторінки для уникнення дублювання контенту;

Alt (Alternative Text) – текстовий опис зображення на вебсторінці;

Text/HTML ratio – співвідношення обсягу текстового контенту до обсягу HTML-коду на сторінці.

ВСТУП

Актуальність теми. У сучасних умовах цифрової економіки вебресурси є ключовим каналом комунікації компаній із користувачами та партнерами. Пошукова оптимізація (SEO) визначає видимість вебсторінок у пошукових системах, що безпосередньо впливає на відвідуваність, конверсію та конкурентоспроможність компаній. Існуючі інструменти SEO-аудиту, такі як Ahrefs, SEMrush, Moz, забезпечують збір базових даних, проте більшість із них обмежуються шаблонними рекомендаціями та не враховують інтелектуальної обробки даних. Тому актуальним є створення програмного засобу, здатного автоматично аналізувати SEO-параметри та формувати персоналізовані рекомендації на основі алгоритмів машинного навчання, що дозволить підвищити ефективність оптимізації вебресурсів.

Мета дослідження: розроблення програмного засобу для інтелектуального аналізу SEO-параметрів вебресурсів та автоматичної генерації пріоритезованих рекомендацій щодо їх оптимізації.

Об'єкт: Процеси аналізу та оптимізації SEO-параметрів вебресурсів.

Предмет: Методи, алгоритми та програмні засоби інтелектуального аналізу SEO-параметрів та автоматичної генерації рекомендацій для вебресурсів.

Завдання дослідження:

- провести огляд наявних програмних рішень для SEO-аудиту та оцінити їх переваги й недоліки;
- розробити методи автоматизованого збору, обробки та оцінки SEO-параметрів;
- розробити удосконалені алгоритми автоматичної генерації пріоритезованих рекомендацій щодо SEO-оптимізації для досліджуваного вебресурсу;
- створити програмний засіб для автоматичного збору даних, їх обробки та генерації рекомендацій;

— провести тестування програмного засобу та оцінити ефективність його роботи.

В роботі запропоновано інтеграцію алгоритмів інтелектуального аналізу та машинного навчання для автоматичного формування рекомендацій щодо оптимізації SEO-параметрів вебресурсів. Розроблено математичну модель системи, що дозволяє оцінювати ефективність вебсторінок комплексно, враховуючи структурні та контентні показники, а також автоматизувати процес формування рекомендацій, що підвищує точність та персоналізацію порад.

Практична значимість. Розроблений програмний засіб може бути використаний у маркетингових, інформаційних та аналітичних підрозділах компаній для підвищення ефективності SEO-оптимізації вебресурсів. Система дозволяє скоротити час проведення аудиту, підвищити якість рекомендацій, автоматизувати процес оцінки вебсторінок і підвищити конкурентоспроможність компаній на ринку.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Опис предметної області SEO-аналізу вебресурсів

Предметна область дослідження охоплює пошукову оптимізацію (Search Engine Optimization, SEO) як комплексну та динамічну дисципліну, що має на меті підвищення онлайн-видимості вебресурсів у результатах органічної видачі пошукових систем. З моменту свого заснування World Wide Web зазнав докорінних трансформацій: від документально-орієнтованого Web 1.0 до соціального Web 2.0 і, нарешті, до семантичного Web 3.0[1]. Ці технологічні зміни постійно підвищують вимоги до вебсайтів, оскільки використання технологій не лише прискорює завдання, але й забезпечує конкурентоспроможність у сучасну епоху.

У сфері пошукових систем існує постійне прагнення удосконалювати алгоритми для забезпечення оптимальних результатів для користувачів. Однак, критерії категоризації вебсайтів залишаються нерозкритими[2]. Як наслідок, SEO-фахівці змушені застосовувати методи впровадження та спостереження для осмислення цих алгоритмів, визнаючи їхню критичну роль в епоху надлишку інформації в Інтернеті.

Процес оптимізації вебсайту вимагає цілеспрямованих змін у контенті, метатеггах та структурі сайту. Оскільки алгоритми пошукових систем є багатограними та динамічними, перевести підвищення рейтингу вебсайту в результатах пошукової видачі на одну єдину SEO-стратегію було б важко. З огляду на це, необхідною є диверсифікована (гібридна) стратегія SEO, що включає комплексний підхід та поєднання on-page і off-page методів[3].

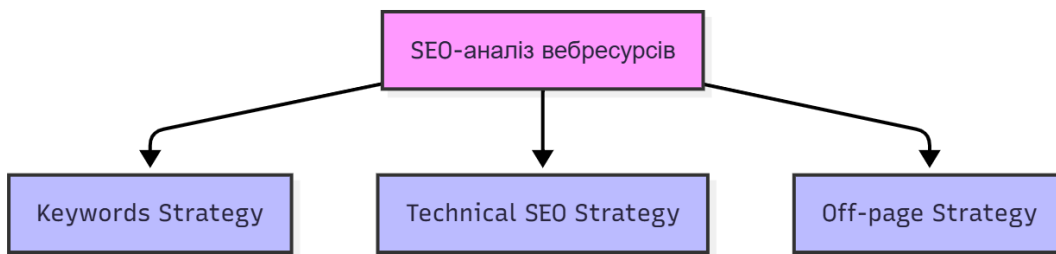


Рисунок 1.1 – Схема SEO-стратегій

Сучасний SEO-аналіз вебресурсів включає оцінку трьох основних кластерів стратегій, які у сукупності функціонують злагоджено, щоб підвищити рейтинг сайту та сприяти загальній онлайн-видимості.

Таблиця 1.1 – Кластерів SEO-параметрів

Кластер SEO-стратегії	Параметр	Метрика / показник	Метод оцінки / інструмент
Keywords Strategy	Аналіз ключових слів	Кількість та популярність	SEO-інструменти (Ahrefs, SEMrush)
	CTR (Click-Through Rate)	Відсоток кліків по результатах пошуку	Аналітика Google Search Console
	Щільність ключових слів	% ключових слів на 100 слів	Контент-аналіз
Technical SEO Strategy	Оптимізація структури сайту	Внутрішні посилання, карта сайту	Аудит сайту (Screaming Frog, Sitebulb)
	Оптимізація коду та серверу	Page Speed, мобільна адаптація	Lighthouse, GTmetrix
	Моніторинг технічних показників	404, 500 помилки, sitemap	Веб-аналітика, Google Search Console
Off-page Strategy	Зворотні посилання	Domain Score, Authority	Ahrefs, Moz, SEMrush
	Релевантність донорів	Тематична відповідність	Аналіз контенту донорів
	Якість контенту з посиланням	Engagement, трафік	Веб-аналітика, Ahrefs

1. Стратегія ключових слів (Keywords Strategy)

Аналіз ключових слів та дослідження їхньої популярності створюють фундаментальну основу будь-якої SEO-кампанії. Це критичний крок для покращення видимості шляхом забезпечення вищого ранжування за релевантними ключовими запитами.

Аналіз ключових слів включає:

- створення вичерпного переліку потенційних ключових слів на основі виявлення зовнішньо популярних релевантних запитів;
- визначення ключових слів, що мають потенціал для позиціювання сайту у топ-50 результатів пошуку, що підвищує клікабельність;
- оцінка частоти ключових слів, де загалом рекомендується підтримувати приблизно 5-7 слів на 100 слів контенту.

Успішні кампанії повинні розуміти, що шукають люди, скільки людей шукає це, та в якому форматі вони бажають отримати інформацію. При цьому необхідно визнати, що аудиторія може бажати зовсім іншого, ніж ключові слова, початково обрані для ранжування.

2. Технічне SEO (Technical SEO Strategy)

Технічне SEO являє собою систематичну оптимізацію структурного каркаса та коду вебсайту. Ця стратегічна увага спрямована на сприяння виявленню та розумінню ресурсу пошуковими системами. Технічні фахівці мають необхідні навички для маніпулювання кодом та конфігурацією сервера, а також для виявлення та усунення технічних перешкод, що заважають пошуковим павукам зрозуміти контент.

Сама внутрішня важливість технічного SEO полягає в його здатності забезпечити узгодженість того, що шукають пошукові системи у вебсайті, та його базової структури чи інфраструктури. Фахівець із технічного SEO забезпечує не лише початкову оптимізацію елементів, але й постійний моніторинг та необхідні коригування цих технічних компонентів протягом життєвого циклу сайту. Втручання технічних експертів є необхідним при масштабуванні сайту, оскільки проблеми

неминуче виникають, а також у випадку порушень безпеки, коли Google не схильний направляти трафік на скомпрометований ресурс. У підсумку, технічна сторона SEO відіграє вирішальну роль у забезпеченні працездатності сайту.

3. Зовнішня оптимізація (Off-page Strategy)

Зовнішня стратегія охоплює всі заходи, що вживаються поза межами вебсайту, ключовим з яких є аналіз зворотних посилань (backlink analysis). Зворотні посилання з авторитетних вебсайтів допомагають пошуковим системам визначити авторитетність та релевантність ресурсу.

Якість посилань є значно важливішою, ніж їхня кількість. Стратегічне формування профілю зворотних посилань вимагає дотримання кількох критеріїв, включаючи релевантність донорів та створення ними високоякісного контенту. Критичною є тематична релевантність сторінки-донора та її відповідність контенту, на який вона посилається.

Для забезпечення мінімального рівня авторитетності та достовірності часто встановлюється нижній поріг Domain Score 50.

Контент, у який включається зворотне посилання, має бути цінним та інформативним, а також природно відповідати наміру користувача для сторінки-донора.

Отже, SEO-аналіз є багатофакторним процесом, що вимагає впровадження гібридного підходу. Дослідження показують сильний та позитивний зв'язок між стратегією ключових слів, технічним SEO та off-page стратегіями з рейтингом вебсайту. Це підтверджує, що для досягнення максимальної ефективності необхідна їхня злагоджена робота. Зважаючи на цю складність, актуалізується задача розробки інтелектуального програмного засобу для автоматизації аудиту та надання персоналізованих рекомендацій.

1.2 Аналіз існуючих рішень для SEO-аудиту

Існуючі програмні засоби для SEO-аналізу є прямим відображенням багатофакторності предметної області, описаної у підрозділі 1.1. В умовах

масштабування вебресурсів та постійного оновлення алгоритмів пошукових систем, ручний аудит стає неефективним, що зумовлює необхідність використання високоспеціалізованих автоматизованих інструментів. Аналіз ринку показує, що рішення можна класифікувати за їхнім функціональним призначенням та архітектурою, при цьому більшість з них охоплюють лише окремі аспекти комплексної SEO-стратегії [4].

Класифікація існуючих SEO-інструментів

1. Комплексні SaaS-платформи (All-in-one Solutions)

Ці платформи є хмарними рішеннями, що пропонують широкий спектр функцій для всіх трьох кластерів SEO-аналізу. Їхньою сильною стороною є аналіз конкуренції, глибоке дослідження ринку та велика база даних ключових слів і зворотних посилань [5]:

- Ahrefs, є лідером у сфері аналізу зворотних посилань (Off-page Strategy).
Дає змогу досліджувати профіль посилань конкурентів, авторитетність доменів і динаміку зростання;
- SEMrush / Serpstat, це баланс між Keywords Strategy, Technical Audit та Off-page. Serpstat має актуальність для локального ринку.

Недоліки: висока ціна, залежність від кваліфікації спеціаліста, велика кількість сирих даних.

2. Десктопні та спеціалізовані краулери

Ці засоби забезпечують глибоке технічне сканування, моделюючи роботу пошукового робота.

- Screaming Frog / Netpeak Spider, це стандарт індустрії для збору даних про відповіді сервера, метатеги, заголовки, внутрішні та зовнішні посилання, атрибути canonical, hreflang.

Переваги: точність технічного аудиту, детальний збір даних.

Недоліки: високе навантаження на ПК, потреба у ручному фільтруванні результатів.

3. Базові та безкоштовні інструменти

Ці інструменти використовують дані безпосередньо від пошукових систем, тому є незамінними в SEO-практиці:

- Google Search Console (GSC), де критична інформація щодо індексації, сканування, Core Web Vitals;
- Google Analytics, поведінкові фактори (Bounce Rate, Time on Page тощо) [6].

Недоліки: мінімум рекомендацій, необхідність інтерпретації.

Аналіз ринку показує, що сучасні інструменти покривають окремі компоненти SEO, але мають обмеження, які створюють передумови для наукової розробки:

- відсутність інтеграції між технічним аудитом, ключовими словами та посилальним аналізом;
- немає AI-пріоритезації, системи генерують багато помилок, але не оцінюють їхній вплив і не формують персоналізовані рекомендації;
- висока вартість SaaS-рішень для малого і середнього бізнесу.

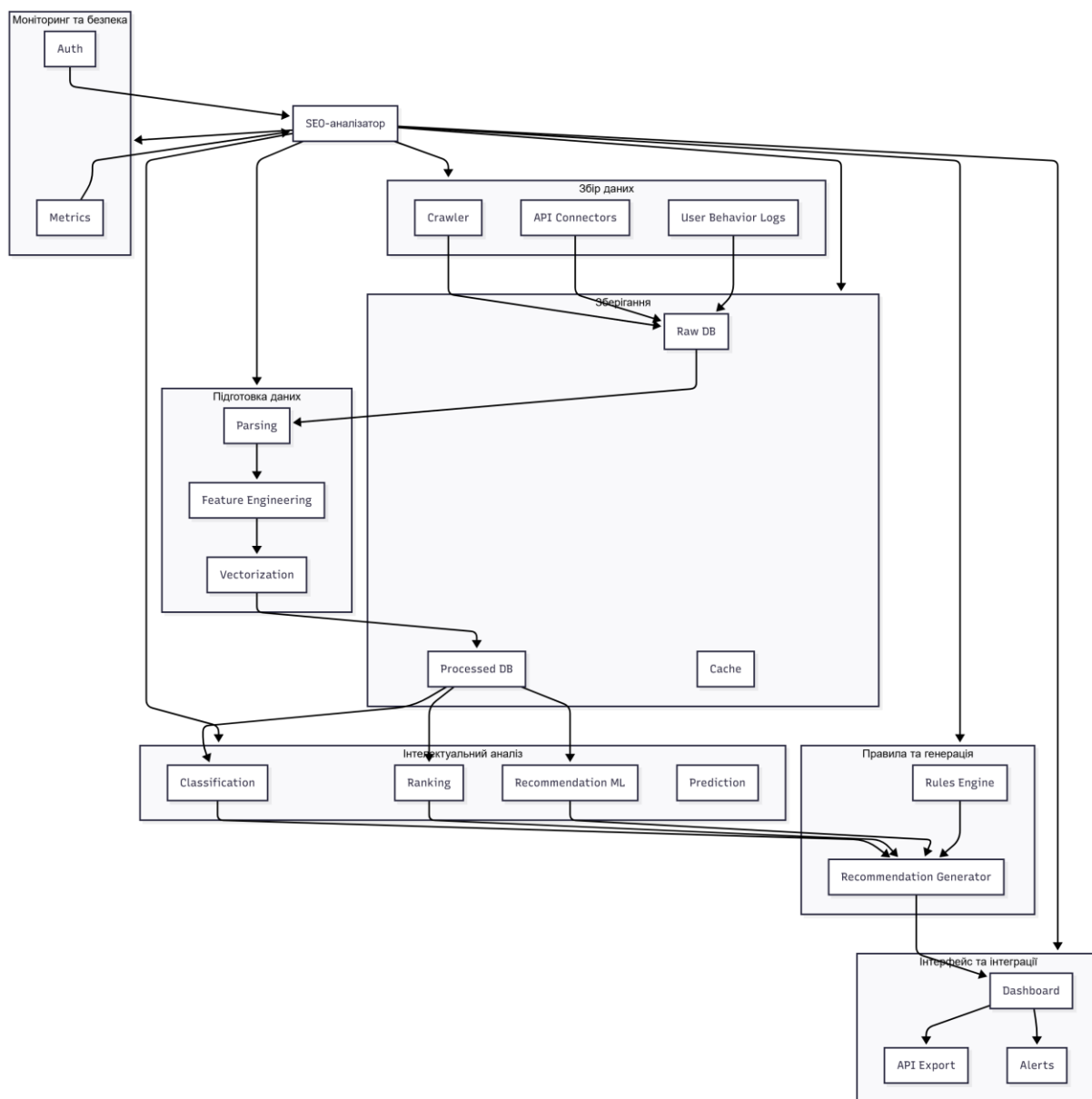


Рисунок - 1.2 Концептуальна схема інтелектуального програмного засобу для SEO-аналізу

Таким чином, актуальною є розробка програмного засобу, що поєднує можливості глибокого технічного сканування з інтелектуальним ядром для автоматичної класифікації проблем та надання пріоритетованих, практичних рекомендацій (згідно зі рисунком 1.2), що усуне ключовий недолік наявних на ринку інструментів.

1.2.1 Огляд інструмента 1

Ahrefs є однією з провідних комплексних (All-in-one) хмарних платформ у галузі SEO-аналітики, яка використовується спеціалістами по всьому світу. Його архітектура базується на масштабній базі даних, що безперервно оновлюється та індексується, включаючи мільярди сторінок та трильйони зворотних посилань [7].

Ahrefs вирізняється своєю спеціалізацією на аналізі зовнішньої оптимізації (Off-page Strategy), але при цьому забезпечує функціонал для всіх трьох ключових кластерів SEO-аналізу [8].

Аналіз посилань (Off-page Strategy):

- Site Explorer, ключовий інструмент для глибокого аналізу профілю зворотних посилань. Надає інформацію про кількість доменів-донорів, їхню авторитетність (Domain Rating, DR), динаміку нарощування посилань та типи анкорних текстів[9]. Використовується для ідентифікації якісних джерел посилань конкурентів, виявлення шкідливих (спамних) посилань, а також для оцінки сили посилального профілю сайту.

Дослідження ключових слів (Keywords Strategy):

- Keywords Explorer, надає детальний аналіз ключових слів, включаючи обсяг пошуку, складність ранжування та аналіз SERP-видачі для обраного запиту[10]. Використовується для формування семантичного ядра, кластеризації запитів та визначення оптимальних стратегій контент-маркетингу.

Технічне та контентне SEO:

- Site Audit, вбудований краулер, що сканує сайт та виявляє технічні помилки (бити посилання, проблеми з метатегами, дублікати контенту, проблеми з індексацією)[11]. Хоча функціонал технічного аудиту є

потужним, він часто слугує допоміжним інструментом у порівнянні з основними можливостями платформи.

Незважаючи на високу ефективність, Ahrefs має низку обмежень, які підтверджують актуальність розробки нової системи, орієнтованої на інтелектуальний аналіз:

- платформа є дорогою, що робить її недоступною для малого бізнесу та індивідуальних розробників, обмежуючи тим самим ринок для повноцінного SEO-аналізу [12];
- Ahrefs генерує значні обсяги даних та списки помилок. Проте, він не використовує алгоритми машинного навчання (ML) для автоматичного прогнозування, яка саме помилка матиме найбільший вплив на рейтинг у поточному конкурентному середовищі [13];
- попри наявність функції Site Audit, Ahrefs історично та функціонально орієнтований на посилальний профіль та конкурентну розвідку, залишаючи глибокий, гнучкий технічний аудит за межами свого основного фокусу [14].

1.2.2 Огляд інструмента 2

Screaming Frog SEO Spider — це десктопний краулер, який зарекомендував себе як індустріальний стандарт для глибокого технічного та внутрішнього (On-page) аудиту вебсайтів. Інструмент імітує поведінку пошукового робота (Googlebot) та дозволяє швидко сканувати велику кількість сторінок, збираючи тисячі критично важливих параметрів.

На відміну від Ahrefs, SF є вузькоспеціалізованим інструментом, орієнтованим переважно на технічний аспект SEO-аналізу:

- збирає дані про коди відповіді сервера (4xx, 5xx, 301, 200), виявляє биті посилання, перевіряє директиви nofollow, noindex, а також валідність файлів robots.txt та sitemap.xml;

- аналізує внутрішню архітектуру посилань, глибину сканування, цикли перенаправлень, та перевіряє налаштування елементів Canonical;
- незамінний для виявлення критичних помилок, які безпосередньо заважають пошуковим роботам індексувати сайт;
- збирає та перевіряє всі елементи On-page оптимізації: довжину та унікальність тегів <title>, <meta description>, наявність та ієрархію заголовків H1–H6;
- виявляє точні дублікати сторінок, а також сторінки з майже ідентичним контентом;
- інструмент може збирати список зовнішніх посилань, але не виконує їхнього якісного аналізу (наприклад, не перевіряє DR донора та його спамність) – для цього потрібна інтеграція з API сторонніх сервісів (Ahrefs, Moz).

Обмеження Screaming Frog прямо вказують на необхідність створення уніфікованої системи:

- SF працює лише з одним сайтом. Він не збирає дані про конкурентів та не аналізує їхні ключові слова, що є критичним для стратегічного SEO-аналізу;
- інструмент вимагає значних ресурсів локального комп'ютера (RAM) для сканування великих корпоративних вебресурсів. Крім того, він надає сирі дані у вигляді таблиць, які фахівець мусить вручну фільтрувати, категоризувати та інтерпретувати для формування рекомендацій;
- SF лише виявляє помилки, але не надає пріоритезованих, автоматично згенерованих рекомендацій, базованих на прогнозуванні впливу. Це основний недолік, який має усунути інтелектуальний програмний засіб.

1.3 Постановка задачі створення програмного засобу

Проведений аналіз предметної області та огляд існуючих програмних рішень чітко окреслили актуальність та наукову новизну задачі розробки інтелектуального програмного засобу. Сучасні SEO-системи доводять, що ефективний аудит вебресурсів неможливий без комплексного поєднання технічних, семантичних і зовнішніх чинників оптимізації, що підтверджується сучасними дослідженнями машинного ранжування [15].

SEO-аналіз на сьогодні вимагає гібридної стратегії, яка включає три взаємозалежні кластери: Технічне SEO, Стратегію ключових слів, та Зовнішню оптимізацію. Наявні інструменти продемонстрували значну фрагментацію:

- комплексні платформи (наприклад, Ahrefs) є дорогими, переважно орієнтовані на Off-page та конкурентний аналіз і не надають інтелектуальної, автоматичної пріоритезації виявлених SEO-помилки, що зазначається у сучасних оглядах інструментів [16];
- спеціалізовані краулери (наприклад, Screaming Frog) забезпечують глибокий технічний аудит, але формують переважно сирі дані, що потребують значної ручної інтерпретації та експертної оцінки [17].

Ключова проблема полягає у відсутності доступного, інтегрованого програмного засобу, здатного не лише виявляти помилки за всіма кластерами SEO, але й автоматично оцінювати та прогнозувати їхній вплив на рейтинг вебресурсу, застосовуючи алгоритми машинного навчання та класифікації, що підтверджується останніми дослідженнями щодо застосування ML для SEO-процесів [18].

Об'єктом дослідження є процес комплексного SEO-аналізу вебресурсів та автоматизована генерація рекомендацій для підвищення їхньої видимості у пошукових системах.

Метою роботи є розробка архітектури та програмного засобу для інтелектуального аудиту SEO-параметрів вебресурсів, який, використовуючи

алгоритми класифікації та пріоритезації, забезпечує формування персоналізованих та практичних рекомендацій для оптимізації.

Для досягнення поставленої мети необхідно вирішити наступні дослідницькі та прикладні задачі:

- аналіз вимог та архітектури, провести детальний аналіз функціональних і нефункціональних вимог до інтелектуального SEO-аналізатора та розробити його модульну архітектуру відповідно до принципів сучасних аналітичних систем [19];
- розробка модуля збору даних, створити програмний модуль, здатний ефективно сканувати вебресурс для збору необхідних технічних та On-page параметрів, а також інтегруватися з джерелами зовнішніх даних (GSC, GA, API посилань). Застосування комбінованих джерел даних рекомендовано сучасними SEO-дослідженнями [20];
- проектування та реалізація бази даних, для ефективного зберігання та аналізу SEO-параметрів, включно з історичними показниками, як це практикується у професійних платформах [21];
- створення інтелектуального ядра аналізу, розробити алгоритми класифікації та механізми машинного навчання для автоматичної ідентифікації SEO-помилки і пріоритезації їхнього впливу на позиції вебресурсу, що відповідає рекомендаціям сучасних ML-моделей для маркетингової аналітики [22];
- розробка модуля генерації рекомендацій, створити механізм, який перетворює пріоритезовані результати аналізу на чіткі, структуровані та практичні рекомендації, враховуючи тип вебресурсу, конкурентне середовище та типові патерни оптимізації [23];
- розробка програмного інтерфейсу (UI), реалізувати інтуїтивний, зручний інтерфейс користувача, у якому результати аудиту, показники

пріоритезації та рекомендації будуть подані доступно. Підхід до UX побудовано відповідно до сучасних принципів візуальної SEO-аналітики [24].

Висновки до розділу

У першому розділі було розглянуто теоретичні основи та здійснено постановку задачі створення інтелектуального програмного засобу для SEO-аналізу вебресурсів.

Аналіз предметної області та її структура. Було детально описано предметну область SEO-аналізу, яка є багатофакторною та динамічною дисципліною. Встановлено необхідність застосування гібридної SEO-стратегії, що ґрунтується на злагодженій взаємодії двох ключових типів SEO-показників: Технічного SEO та контентної складової (on-Page). Для візуалізації цієї ієрархії було розроблено багатофакторну модель SEO (Рисунок 1.1), де Технічне SEO є необхідним фундаментом.

Огляд та оцінка існуючих рішень. В ході аналізу було проведено огляд існуючих програмних рішень, зокрема комплексних платформ (Ahrefs) та спеціалізованих краулерів (Screaming Frog SEO Spider). Було встановлено, що, незважаючи на потужний функціонал, існуючі рішення мають наступні суттєві недоліки:

- інструменти фокусуються лише на окремих кластерах SEO;
- інструменти надають великі обсяги сирих даних, які вимагають ручної експертної інтерпретації та фільтрації;
- існуючі рішення не використовують алгоритми машинного навчання (ML) для автоматичної пріоритезації виявлених помилок за їхнім потенційним впливом на рейтинг.

На основі виявлених прогалин та недоліків ринку була обґрунтована актуальність розробки та сформульована мета роботи: створення інтелектуального програмного засобу, який поєднує комплексний збір даних із автоматичною

генерацією пріоритезованих, персоналізованих рекомендацій. Концептуальну структуру цього засобу було відображено на рисунку 1.2.

Для досягнення мети роботи було поставлено низку задач, ключовими з яких є розробка:

- модуля ефективного збору та інтеграції SEO-параметрів;
- інтелектуального ядра аналізу на основі ML-алгоритмів для класифікації та пріоритезації проблем;
- модуля, що трансформує аналітичні висновки у чіткі, практичні рекомендації.

Таким чином, розділ повністю вирішив завдання теоретичного обґрунтування та постановки проблеми, визначивши ключові вимоги та напрямки для подальшої розробки програмного засобу.

2 ТЕОРЕТИЧНІ ОСНОВИ ІНТЕЛЕКТУАЛЬНОГО SEO-АНАЛІЗУ

2.1 Основні SEO-показники та їх значущість

Ефективність інтелектуального SEO-аналізу визначається якістю зібраних даних та здатністю системи правильно нормалізувати й оцінювати ключові показники, що впливають на позиції вебресурсу в пошуковій видачі. Комплексний аналіз вимагає контролю метрик, які охоплюють три основні кластери SEO: технічне SEO, контент і ключові слова, зовнішня оптимізація. Кожен кластер містить групи показників, що мають підтверджений позитивний вплив на ранжування.

Технічна оптимізація забезпечує можливість пошуковим роботам коректно сканувати, інтерпретувати та індексувати сторінки сайту. Помилки на цьому рівні можуть значно знизити або повністю заблокувати видимість ресурсу. Тому технічні метрики формують фундамент для подальшого аналізу.

Таблиця 2.1 – Технічні показники SEO-аналізу

№	Показник	Опис	Норма/Рекомендація
1	Завантаження сторінки	Швидкість завантаження сайту на мобільних та ПК	> 85 балів
2	Мобільна адаптивність	Коректність відображення на смартфонах	Повністю адаптивний інтерфейс
3	First Contentful Paint (FCP)	Час появи першого видимого контенту на сторінці.	< 1.8 секунди (Задовільно/Good)
4	Файл robots.txt	Коректність інструкцій для Googlebot	Наявний, без блокування сторінок
5	Sitemap.xml	Карта сайту для індексації	Наявний, коректно оновлюється
6	Структура URL	Довжина, логічність, формат	Зрозумілі URL

Продовження таблиці 2.1

№	Показник	Опис	Норма/Рекомендація
7	Largest Contentful Paint (LCP)	Час завантаження найбільшого видимого елемента.	< 2.5 секунди (Задовільно/Good)
8	Cumulative Layout Shift (CLS)	Вимірює візуальну нестабільність сторінки.	< 0.1 (Задовільно/Good)

Показники кластеру ключових слів та контент визначають, наскільки контент відповідає пошуковим запитам користувачів. Семантична оптимізація та якість контенту є критичними факторами, що визначають здатність сторінки задовольнити інформаційний намір користувача.

Таблиця 2.2 – On-page контентні показники

№	Показник	Опис	Рекомендація
1	Title	Унікальність, довжина, релевантність	50–60 символів, 1 ключ
2	Meta Description	Інформативність і релевантність	140–160 символів
3	H1-заголовок	Унікальний головний заголовок	H1 на сторінку
4	Структура H2–H4	Логічна структура тексту	Є ієрархічність підзаголовків
5	Ключові слова	Оптимальна кількість у тексті	1–2% щільності
6	Canonical Link	Наявність канонічного посилання	Встановлено коректне канонічне посилання

Продовження таблиці 2.2

№	Показник	Опис	Рекомендація
7	Alt-атрибути зображень	Текстові описи картинок	Alt присутній
8	Довжина контенту	Кількість слів/символів на сторінці	Оптимальний обсяг повинен відповідати середньому значенню топ-50 конкурентів.

Аналіз конкурентів є важливою складовою інтелектуального SEO-аналізу, оскільки дозволяє порівняти власний вебресурс із найближчими конкурентами та визначити сильні та слабкі сторони у використанні ключових слів, структури контенту та елементів on-page оптимізації. Для цього збираються та оцінюються показники, що відображають стратегії конкурентів у побудові заголовків, метаданих, контентної щільності та внутрішньої структури сайту.

Таблиця 2.3 – Показники аналізу конкурентів

№	Показник	Опис	Рекомендація / Норма
1	Завантаження сторінки	Загальна швидкість завантаження сайту (лабораторні дані)	> 85 балів (PageSpeed Insights Score)
2	Мобільна адаптивність	Коректність відображення на смартфонах	Повністю адаптивний інтерфейс
3	First Contentful Paint (FCP)	Час появи першого видимого контенту на сторінці	< 1.8 секунди (Задовільно/Good)

Продовження таблиці 2.3

№	Показник	Опис	Рекомендація / Норма
4	Файл robots.txt	Коректність інструкцій для сканерів	Наявний, без блокування важливих сторінок
5	Sitemap.xml	Карта сайту для індексації	Наявний, коректно оновлюється
6	Структура URL	Довжина, логічність, формат URL	Зрозумілі, короткі, транслітеровані URL
7	Largest Contentful Paint (LCP)	Час завантаження найбільшого видимого елемента.	< 2.5 секунди (Задовільно/Good)
8	Cumulative Layout Shift (CLS)	Вимірює візуальну нестабільність сторінки.	< 0.1 (Задовільно/Good)
9	Title	Унікальність, довжина, релевантність	50–60 символів, 1 основний ключ
10	Meta Description	Інформативність, релевантність, СТА	140–160 символів, заохочує до кліку
11	H1-заголовок	Унікальний головний заголовок сторінки	Один H1 на сторінку, містить основний ключ

Продовження таблиці 2.3

№	Показник	Опис	Рекомендація / Норма
12	Структура H2– H4	Логічна та ієрархічна структура тексту	Є ієрархічність підзаголовків
13	Ключові слова	Оптимальна кількість входжень ключів у тексті	1–2% щільності
14	Canonical Link	Наявність канонічного посилання	Встановлено коректне канонічне посилання
15	Alt-атрибути зображень	Текстові описи картинок	Alt присутній, релевантний зображенню
16	Довжина контенту	Кількість слів/символів на сторінці	Оптимально – як у конкурентів з ТОП-50

Оцінка структури, контентних рішень та ключових слів конкурентів дозволяє визначати пріоритетні напрямки поліпшення власного вебресурсу та формувати пріоритизовані рекомендації.

2.2 Методи збору та оцінки SEO-параметрів

Ефективність інтелектуального SEO-аналізу залежить не лише від збору великої кількості даних, а й від їх правильної обробки, визначення цільових значень та статистичної оцінки показників. Для кожного вебресурсу необхідно порівнювати його SEO-параметри із орієнтирами, отриманими з конкурентів або галузевих стандартів, що дозволяє виділити слабкі місця та сформувані пріоритетні рекомендації.

Ключовим елементом системи є Парсер, який обробляє HTML-код сторінки, виділяє метатеги та контент, а також аналізує семантичну структуру для оцінки релевантності. Зібрані дані передаються до Модуля порівняльного аналізу.

Модуль порівняльного аналізу дозволяє визначити «орієнтири» — середні та медіанні значення параметрів у ніші, від яких відштовхуються при формуванні рекомендацій для аналізованого сайту.

Щоб оцінити ефективність вебресурсу та визначити його слабкі місця, необхідно мати цільові значення SEO-параметрів. У нашому програмному засобі використовуються два статистичних показники:

1. Середнє значення (Mean)

Середнє значення є арифметичною серединою зібраних даних і обчислюється за формулою:

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n} \quad (2.1)$$

де,

— x_i — значення показника на конкретній сторінці або конкуренті,

— n — кількість вимірювань.

Застосування:

— технічні показники (час завантаження, кількість внутрішніх посилань)

— використовується для нормалізації даних та оцінки загальної тенденції.

2. Медіана (Median)

Медіана — це значення, що ділить впорядковану вибірку на дві рівні частини.

Формула залежить від кількості спостережень (n):

$$\text{Median} = \begin{cases} x_{\frac{n+1}{2}}, & n \text{ не парне} \\ \frac{x_{\frac{n}{2}} + x_{\frac{n}{2}+1}}{2}, & n \text{ парне} \end{cases} \quad (2.2)$$

Застосування:

- контентні та on-page показники, які можуть мати екстремальні значення (дуже короткі або дуже довгі тексти)
- медіана забезпечує менший вплив викидів, ніж середнє, та дає більш стабільний орієнтир для рекомендацій.

Для забезпечення наукової обґрунтованості рекомендацій, система виконує два ключові етапи:

- Розрахунок інтегрованого цільового значення

Для кожного i -го SEO-показника (наприклад, LCP, довжина контенту, кількість H2) визначається інтегроване цільове значення T_i . Це значення є узагальненим орієнтиром, розрахованим на основі статистичного аналізу вибірки з 50–75 конкурентів з ТОП видачі Google:

$$T_i = \text{Середнє(конкуренти) або Медіана(конкуренти)}.$$

- Обчислення відхилення Δi

Обчислюється відхилення Δi досліджуваного сайту $X_{\text{site},i}$ від інтегрованого цільового значення конкурентів T_i . Це відхилення є критичним показником, оскільки відображає ступінь, на який наш сайт відстає чи випереджає орієнтир за конкретною метрикою.

$$\Delta i = X_{\text{site},i} - T_i \quad (2.3)$$

Після визначення цільових значень параметрів необхідно забезпечити їх порівнянність та інтеграцію у систему оцінки. Для цього використовується нормалізація даних:

$$N_i = \frac{X_i - X_{\min}}{X_{\max} - X_{\min}} \quad (2.4)$$

де:

- X_i — значення конкретного параметра,
- $X_{\min}$, $X_{\max}$ — мінімальне та максимальне значення параметра серед конкурентів або історичних даних.

Це перетворює всі параметри у єдиний діапазон 0–1, що дозволяє коректно порівнювати різні метрики та обчислювати зважену оцінку ефективності сторінки:

$$S_{\text{SEO}} = \sum_{i=1}^n w_i \cdot N_i \quad (2.5)$$

де, w_i — ваговий коефіцієнт важливості показника, розрахований на основі статистичного аналізу впливу на рейтинг.

Ключова наукова новизна програмного засобу полягає у використанні кластеризації K-Means для пріоритезації рекомендацій, що базується на двох критично важливих параметрах: ваговому коефіцієнту w_i та відхиленні Δ_i .

Кожен i -й SEO-показник представлений як точка у двовимірному просторі:

$$P_i = (w_i, \Delta_i) \quad (2.6)$$

Спочатку для визначення оптимальної кількості кластерів K застосовується метод ліктя (Elbow Method). Це дозволяє вибрати таке K , при якому подальше збільшення кластерів уже не дає значного зменшення суми квадратів відстаней усередині кластерів (WCSS).

За допомогою алгоритму K-Means точки P_i групуються у K кластерів. Кожен кластер об'єднує показники, які мають схожу важливість і подібну ступінь відставання від цільового значення.

Пріоритетність рекомендацій визначається розташуванням центроїдів кластерів у просторі (w, Δ) .

1. Кластер 1 (Найвищий пріоритет), центроїд якого має найбільше значення w (найвища важливість для ранжування) та найбільш небажане відхилення Δ (сильне відставання від цільового значення). Це, як правило, кластер із самим верхнім лівим центроїдом (висока важливість, сильне відставання). Оптимізація цих показників дасть найбільший приріст ефективності.

2. Послідовність пріоритетів, це подальші рівні пріоритетності визначаються відстанню між центроїдами кластерів, забезпечуючи логічну та ефективну черговість впровадження рекомендацій.
- а. Кластер 2, визначається як кластер із найменшою евклідовою відстанню між його центроїдом та центроїдом Кластера 1.
 - б. Кластер 3, визначається як найближчий кластер до центроїда Кластера 2.
 - с. Процес продовжується, гарантуючи, що ресурси оптимізатора будуть спрямовані на показники, які є наступними за пріоритетом впливу.

Такий підхід дозволяє системі формувати точні, індивідуальні та статистично обґрунтовані пріоритезовані рекомендації, що значно підвищує ефективність SEO-оптимізації вебресурсу порівняно з простими системами оцінки.

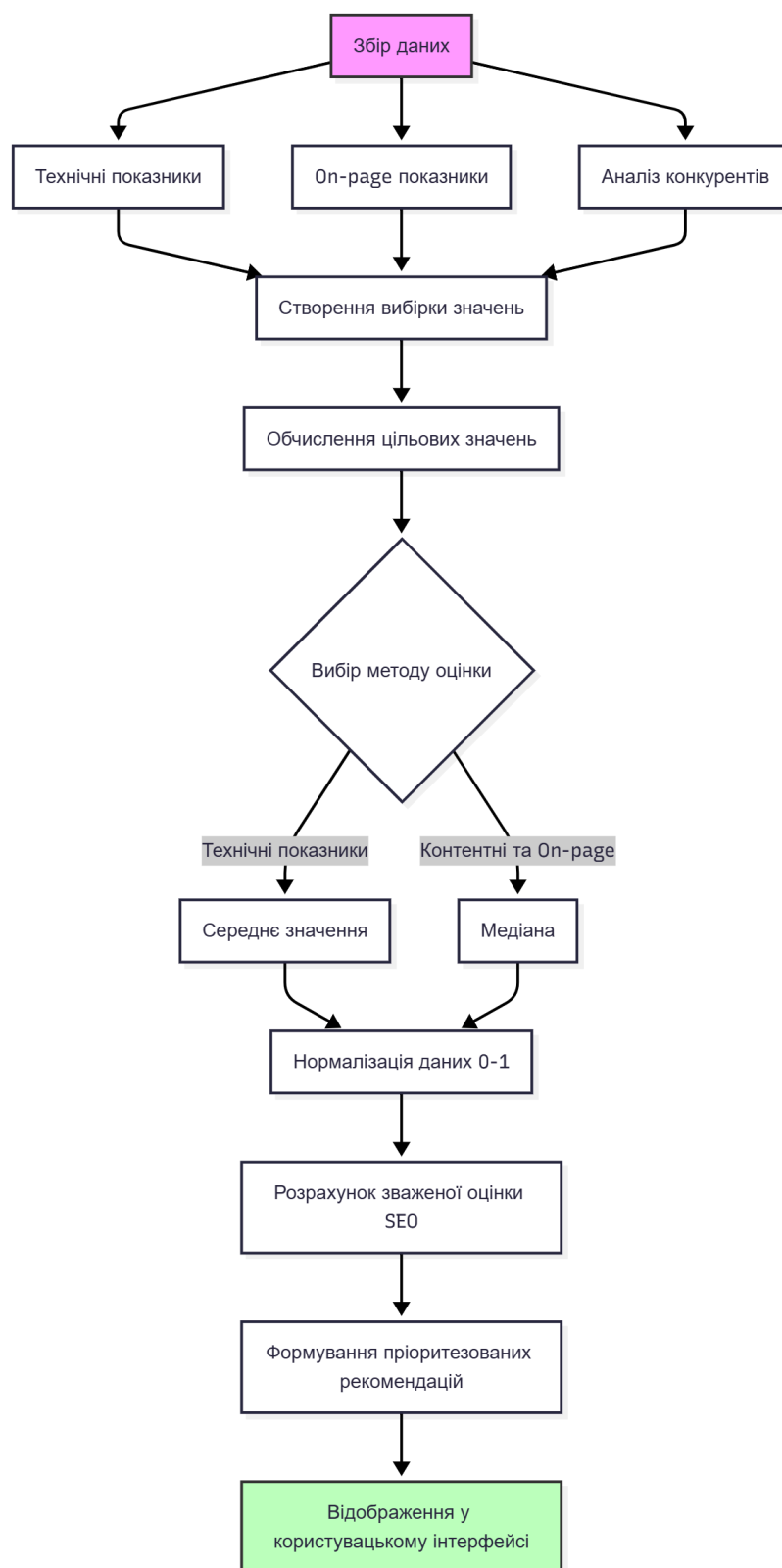


Рисунок 2.1 – Схема SEO-стратегій

Схема демонструє роботу інтелектуального SEO-аналізатора від збору даних до генерації рекомендацій. Система збирає технічні показники, On-page метрики та дані

конкурентів, формує вибірки та обчислює цільові значення (середнє для технічних, медіану для контентних). Дані нормалізуються та комбінуються з ваговими коефіцієнтами для формування загальної оцінки SEO. На основі цієї оцінки генеруються пріоритезовані рекомендації, які відображаються у зручному інтерфейсі користувача.

2.3 Методи машинного аналізу

Машинне навчання є ключовим компонентом інтелектуального SEO-аналізатора, оскільки дозволяє автоматично виявляти закономірності в даних, прогнозувати вплив окремих SEO-параметрів на позиції вебресурсу у пошуковій видачі та формувати пріоритезовані рекомендації. Основним методом, використаним у розробленому програмному засобі, є лінійна регресія, яка дозволяє оцінити залежність цільової змінної (положення сайту у рейтингу) від набору незалежних змінних (SEO-параметрів).

Лінійна регресія моделює лінійну залежність між незалежними змінними $X_1, X_2, \dots, X_n$ та цільовою змінною Y за формулою:

$$Y = \beta_0 + \sum_{i=1}^n \beta_i X_i + \epsilon \quad (2.7)$$

де:

- β_0 – вільний член,
- β_i – вагові коефіцієнти, що показують вплив кожного параметра на цільову змінну,
- ϵ – похибка моделі.

Лінійна регресія є одним із фундаментальних методів статистичного аналізу та моделювання, що дозволяє кількісно оцінити вплив одного або кількох незалежних параметрів на цільову змінну. У контексті SEO-прогнозування цільовою змінною виступає позиція вебресурсу у пошуковій видачі, а незалежними змінними —

різноманітні SEO-показники, що характеризують технічні, контентні та поведінкові аспекти сторінки.

Проста лінійна регресія застосовується у випадках, коли позиція сайту залежить лише від одного показника. Наприклад, можна оцінити, як швидкість завантаження основного контенту впливає на рейтинг сторінки. Модель дозволяє визначити числовий коефіцієнт впливу цього параметра, прогнозувати зміни рейтингу при покращенні або погіршенні технічних характеристик, а також оцінити значущість окремого показника у загальному процесі оптимізації сайту.

Множинна лінійна регресія розширює цю концепцію, одночасно враховуючи декілька незалежних змінних. Це особливо важливо для SEO, оскільки позиції вебресурсу формуються комплексом факторів: технічних (швидкість завантаження, структура HTML, коректність метатегів), контентних (наявність H1, оптимізація тексту та мультимедіа) та поведінкових (показники відмов, час на сторінці). Множинна регресія дозволяє оцінити відносний вплив кожного параметра на цільову змінну, виділити найбільш значущі показники та побудувати модель, яка прогнозує рейтинг сайту з урахуванням одночасної зміни всіх ключових характеристик.

Для оцінки точності регресійної моделі застосовується функція втрат у вигляді середньоквадратичної помилки (MSE), яка визначається як середнє квадратичне відхилення прогнозованих значень від реальних:

$$MSE = \frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2 \quad (2.8)$$

де:

- y_j — фактичне значення цільової змінної,
- $\hat{y}_j$ — прогноз моделі
- n — загальна кількість спостережень

Функція втрат показує середнє квадратичне відхилення прогнозованих значень від реальних. Чим менше значення, тим точніше модель передбачає позиції

сайту. Функція середньоквадратичної помилки є ключовим інструментом для оцінки точності моделі регресії, оскільки вона відображає середнє квадратичне відхилення прогнозованих значень від реальних позицій вебресурсів у пошуковій видачі. Чим менше значення MSE, тим точніше модель відображає залежність між SEO-параметрами та рейтингом сайту. Під час навчання алгоритм регресії підбирає вагові коефіцієнти так, щоб мінімізувати MSE, що дозволяє визначити реальний вплив кожного параметра на цільову змінну. У поєднанні з регуляризацією Lasso MSE стає частиною складнішої функції втрат, яка включає додатковий штраф за великі значення коефіцієнтів, контролюючи їх величину та допомагаючи виділити найважливіші SEO-параметри.

$$\text{Loss}_{Lasso} = \text{MSE} + \lambda \sum_{i=1}^n |\beta_i| \quad (2.9)$$

Де, λ контролює силу штрафу на великі коефіцієнти, дозволяючи виділити найважливіші SEO-параметри.

Високе значення MSE свідчить про недостатню точність моделі та потребу у корекції даних або параметрів, тоді як низьке значення показує, що прогноз добре відповідає реальним позиціям, і отримані коефіцієнти можна використовувати для подальшої кластеризації показників за пріоритетами у K-Means.

Завдяки такому підходу можливо не лише визначати критичні фактори для покращення SEO, а й формувати обґрунтовані рекомендації для оптимізації сайту, визначати пріоритетність робіт та прогнозувати ефект від впровадження змін. Лінійна регресія забезпечує прозорість моделі, зрозумілість отриманих коефіцієнтів та можливість інтеграції з подальшими етапами аналітики, включаючи кластеризацію показників для пріоритезації SEO-завдань.

Для підвищення точності прогнозу позицій вебресурсу у пошуковій видачі та виділення ключових SEO-показників у модель лінійної регресії включено регуляризацію. Регуляризація є методикою, що додає до функції втрат додатковий

член, який контролює величину вагових коефіцієнтів та запобігає перенавчанню моделі на даних. У розробленому SEO-аналізаторі використовуються три основні види регуляризації: Lasso (L1-регуляризація), Ridge (L2-регуляризація) та Elastic Net, що поєднує обидва підходи.

Lasso Regression (Least Absolute Shrinkage and Selection Operator) застосовує L1-регуляризацію, яка додає до функції втрат суму модулів коефіцієнтів, забезпечуючи одночасно мінімізацію прогнозової помилки та відбір найбільш значущих ознак. Функція втрат для Lasso описується формулою:

$$J(\beta) = \frac{1}{2n} \sum_{i=1}^n \left(y_i - \sum_{j=1}^p X_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^p |\beta_j|, \quad (2.10)$$

де:

- y_i — фактичне значення цільової змінної (позиція сайту),
- X_{ij} — значення j -го SEO-показника для i -го спостереження,
- β_j — ваговий коефіцієнт,
- n — кількість спостережень,
- p — кількість незалежних змінних,
- λ — параметр регуляризації, що контролює силу штрафу за великі коефіцієнти.

Основною перевагою Lasso є спарсіфікація коефіцієнтів: менш значущі показники отримують нульові ваги, що дозволяє виділити лише ті SEO-параметри, які найбільше впливають на рейтинг сайту. Обчислення коефіцієнтів зазвичай здійснюється методом градієнтного спуску, який зануляє малі коефіцієнти, тим самим виконуючи автоматичний відбір ознак.

Ridge Regression (L2-регуляризація) додає до функції втрат суму квадратів вагових коефіцієнтів:

$$J(\beta) = \frac{1}{2n} \sum_{i=1}^n \left(y_i - \sum_{j=1}^p X_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^p \beta_j^2. \quad (2.11)$$

На відміну від Lasso, Ridge не зануляє коефіцієнти, а лише обмежує їх величину, що зменшує вплив шуму у даних та запобігає перенавчанню моделі при наявності мультиколінеарності між SEO-показниками. Наприклад, швидкість завантаження сторінки та обсяг мультимедійного контенту можуть бути взаємопов'язані, і Ridge дозволяє стабільно оцінити їх вплив без втрати інформації. Оптимальні коефіцієнти для Ridge можуть обчислюватися аналітично за формулою $\beta^{\wedge} = (X^T X + \lambda I)^{-1} X^T Y$,

де:

- I — одинична матриця, що забезпечує регуляризацію,
- λ — параметр, який регулює величину обмеження.

Elastic Net поєднує переваги L1- та L2-регуляризації, включаючи обидва члени у функцію втрат:

$$J(\beta) = \frac{1}{2n} \sum_{i=1}^n \left(y_i - \sum_{j=1}^p X_{ij} \beta_j \right)^2 + \lambda_1 \sum_{j=1}^p |\beta_j| + \lambda_2 \sum_{j=1}^p \beta_j^2. \quad (2.12)$$

L1-регуляризація (λ_1) забезпечує спарсифікацію коефіцієнтів та відбір найбільш значущих SEO-показників, а L2-регуляризація (λ_2) контролює величину коефіцієнтів і зменшує перенавчання.

Ridge обмежує величину коефіцієнтів і забезпечує стабільність при корельованих показниках, а Elastic Net дозволяє зберегти групи взаємопов'язаних

змінних. Проте для задачі виділення ключових SEO-параметрів та формування пріоритетних рекомендацій Lasso є найкращим вибором, оскільки він забезпечує наочний та чіткий відбір найбільш важливих показників і автоматично зануляє незначущі.

Отримані вагові коефіцієнти та відхилення фактичних значень показників від оптимальних, $\Delta_i = y_i^{\text{опт}} - y_i^{\text{факт}}$, разом із вагами w_i формують вектори:

$$P_i = (w_i, \Delta_i), \quad (2.13)$$

які надалі використовуються для кластеризації методом K-Means. Ці кластери дозволяють визначити пріоритетність SEO-показників, формуючи рекомендації для оптимізації технічних, контентних та поведінкових характеристик сторінки. Такий підхід забезпечує прозорість моделі, чітке розуміння впливу окремих параметрів та інтеграцію отриманих результатів у подальші етапи аналітики, зокрема кластеризацію та генерацію пріоритетних SEO-рекомендацій.



Рисунок 2.2 – Процес застосування лінійної регресії для інтелектуального SEO-аналізатора

Рисунок ілюструє послідовність обробки SEO-показників від збору даних до формування пріоритетних рекомендацій. Спершу дані вебресурсів проходять попередню обробку, нормалізацію та масштабування для підготовки матриці X та вектора Y . Після цього вони подаються у модель лінійної регресії для навчання та обчислення коефіцієнтів, які визначають вагу кожного SEO-показника. Далі розраховуються відхилення Δ_i фактичних показників від оптимальних, і разом із вагами w_i вони передаються у модель K-Means для кластеризації пріоритетів. На основі отриманих кластерів система формує пріоритетні SEO-рекомендації, які виводяться у користувацькому інтерфейсі для подальшого впровадження.

2.4 Метод множинної класифікації

Кластеризація є ключовим інструментом для розподілу SEO-показників за пріоритетами. Вона дозволяє групувати подібні показники на основі їхніх властивостей, таких як вага параметра w_i та відхилення фактичного значення від оптимального Δ_i . Кластеризація забезпечує формування пріоритетних груп для подальшої оптимізації вебресурсу. Основними методами кластеризації, використаними у SEO-аналізаторі, є:

- K-Means
- Ієрархічна кластеризація
- DBSCAN

Метод K-Means є одним з найпоширеніших алгоритмів кластеризації. Його мета — розділити множину точок даних на K кластерів таким чином, щоб об'єкти всередині кожного кластера були максимально схожими між собою та максимально відрізнялися від об'єктів інших кластерів.

Функція втрат (цільова функція), яку мінімізує K-Means, задається як:

$$J = \sum_{k=1}^K \sum_{x_i \in C_k} \|x_i - \mu_k\|^2, \quad (2.14)$$

де:

- K — кількість кластерів,
- x_i — точки даних,
- μ_k — центроїд k -го кластера,
- $\|\cdot\|$ — метрика відстані, зазвичай Евклідова:

$$\|x_i - \mu_k\| = \sqrt{\sum_{j=1}^p (x_{ij} - \mu_{kj})^2}. \quad (2.15)$$

Центроїди визначаються як середнє значення всіх точок даних у кластері:

$$\mu_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i. \quad (2.16)$$

Оскільки K-Means потребує попереднього визначення числа кластерів K , для вибору оптимального значення застосовується метод Ліктя:

1. Визначається діапазон можливих $K=1,2,\dots,K_{\max}$.
2. Для кожного K виконується кластеризація.
3. Обчислюється сумарна квадратична відстань точок до своїх центроїдів (WSS — Within-Cluster Sum of Squares) для різних значень K :

$$WSS(K) = \sum_{k=1}^K \sum_{x_i \in C_k} \|x_i - \mu_k\|^2. \quad (2.17)$$

4. Будується графік залежності WSS від K .
5. Вибирається точка «перелому» (elbow), після якої подальше збільшення K не значно зменшує WSS. Це значення K є оптимальним.

Інтерпретація, полягає в тому, що до точки ліктя це швидке зменшення WSS $\rightarrow$ додавання кластерів значно покращує якість, а після точки ліктя це WSS зменшується повільно $\rightarrow$ подальше збільшення K не дає суттєвої користі.

Цей метод дозволяє об'єктивно визначити кількість кластерів для групування сторінок за SEO-проблемами

Після кластеризації важливо правильно інтерпретувати отримані кластери з точки зору пріоритетності SEO-оптимізації. Найефективніший підхід — аналіз центроїдів:

$$\mu_k = (\bar{w}_k, \bar{\Delta}_k) \quad (2.18)$$

де,

- $\bar{w}_k$ — середня вага показників у кластері,
- $\bar{\Delta}_k$ — відхилення від оптимального значення.

Пріоритет кластерів SEO-показників будемо визначати на основі їхніх центроїдів, де:

- кластер з найвищим правим центроїдом (максимальний ваговий коефіцієнт w_i і максимальне відхилення Δ_k) = кластер №1 (пріоритет 1);
- усі інші кластери отримують пріоритети за мінімальною евклідовою відстанню до вже пріоритезованих центроїдів.

Нехай:

- маємо К кластерів;
- кожен кластер має центроїд

$$C_i = (w_i, \Delta_k) \quad (2.19)$$

де:

- w_i — ваговий коефіцієнт,
- Δ_k — відхилення (або будь-яка інша SEO-метрика).

1. Вибір головного центроїда (центроїдMAX)

Знаходимо кластер з найвищими координатами у правому верхньому куті.

Правило вибору центроїдуMAX:

— Максимальний w_i

— Якщо кілька мають однаковий w_i , беремо той з найбільшим Δ_k

Формально:

$$i_{\max} = \arg \max_i (w_i, \Delta_k) \quad (2.20)$$

Це буде пріоритет 1.

2. Обчислення відстаней до центроїдуMAX

Для кожного іншого центроїда C_j обчислюємо евклідову відстань:

$$d(C_j, C_{\max}) = \sqrt{(w_j - w_{\max})^2 + (\Delta_j - \Delta_{\max})^2}. \quad (2.21)$$

Обираємо центроїд з найменшою відстанню від $C_{\max}$. Цей кластер отримує пріоритет 2.

3. Пошук наступних кластерів

Після знаходження кластера з пріоритетом 2, позначаємо його центроїд як C_2 .

Знаходимо серед решти центроїдів той, який має мінімальну відстань до C_2 :

$$i_3 = \arg \min_{j \notin \{1,2\}} d(C_j, C_2) \quad (2.22)$$

Отриманий кластер має пріоритет 3.

4. Продовження процесу

Повторюємо. Останній знайдений центроїд позначаємо як C_k . З решти неопрацьованих кластерів вибираємо той, що розташований найближче до C_k і присвоюємо йому наступний пріоритет. Ітеративно:

$$i_{t+1} = \arg \min_{j \notin \{i_1, i_2, \dots, i_t\}} d(C_j, C_t) \quad (2.23)$$

5. Завершення

Процес зупиняється, коли всі K кластерів отримають пріоритети.

Ієрархічна кластеризація будує дерево схожості (дендрограму) на основі відстаней між точками даних. Основні кроки:

1. Кожна точка даних початково є окремим кластером.
2. На кожній ітерації об'єднуються два найближчі кластери за обраною метрикою відстані (Евклідова, Манхеттен або косинусна).
3. Процес повторюється до утворення одного великого кластера або поки не досягнуто бажаної кількості кластерів.

Ієрархічний підхід зручний для візуального аналізу, оскільки дозволяє відслідковувати, які показники та групи з'єднуються першими, а також обирати рівень розбиття на кластери.

DBSCAN — алгоритм кластеризації на основі щільності, що дозволяє:

- виявляти кластери довільної форми;
- ігнорувати шумові точки, які не належать до жодного кластеру;
- не задавати наперед кількість кластерів K .

DBSCAN класифікує точки за двома параметрами:

- ϵ — максимальна відстань між точками одного кластера;
- minPts — мінімальна кількість точок у щільному кластері.

Точки, що не задовольняють умови щільності, вважаються шумом, що особливо корисно при роботі з реальними SEO-даними, які можуть містити аномалії.

У розробленому інтелектуальному SEO-аналізаторі застосовується K-Means як основний метод кластеризації. Вектори SEO-показників $P_i = (w_i, \Delta_i)$ формують точки в 2D-просторі, які групуються в кластери. Вибір K здійснюється за методом Ліктя, після чого система визначає пріоритетність кластерів. Альтернативні методи (ієрархічна кластеризація та DBSCAN) можуть бути використані для додаткового аналізу або візуалізації, однак для автоматизованого формування пріоритетів K-Means забезпечує найзручніший і швидкий підхід.

Завдяки кластеризації система отримує:

- автоматичне ранжування seo-показників за пріоритетами;
- виявлення найважливіших показників для оптимізації контенту та технічних аспектів;
- підготовку даних для формування рекомендацій, які легко інтегруються у користувацький інтерфейс.

Таблиця 2.4 – Переваги та недоліки методів

Метод	Переваги	Недоліки	Рекомендації для SEO
K-Means	Швидкий, простий, добре працює на великих даних	Потрібно задавати K, погано працює з шумом	Оптимальна кластеризація сторінок за показниками
Ієрархічна	Не потребує K, інтуїтивна дендрограма	Повільна при великих наборах	Для відбору структурних груп
DBSCAN	Не потребує K, стійкий до шуму	Важко підбирати параметри	Для виявлення аномальних сторінок/проблем

2.5 Алгоритми формування рекомендацій

Формування рекомендацій є завершальним етапом інтелектуального SEO-аналізу, у якому результати збору, нормалізації та кластеризації даних перетворюються на практичні, пріоритезовані дії для користувача. Основна мета цього етапу — надання конкретних порад, які дозволяють підвищити видимість вебресурсу у пошукових системах.

Запропонований алгоритм забезпечує системну пріоритезацію дій, персоналізує рекомендації під конкретний сайт з урахуванням конкурентного середовища, дозволяє прогнозувати потенційний приріст рейтингу при оптимізації кожного показника і значно скорочує час на ручний аудит завдяки автоматизації процесу. Він поєднує визначені Lasso Regression ваги SEO-показників, відхилення від конкурентів та кластеризацію для виділення пріоритетних груп показників, що створює науково обґрунтовану основу для автоматизованого SEO-аналізу та формування практичних рекомендацій.

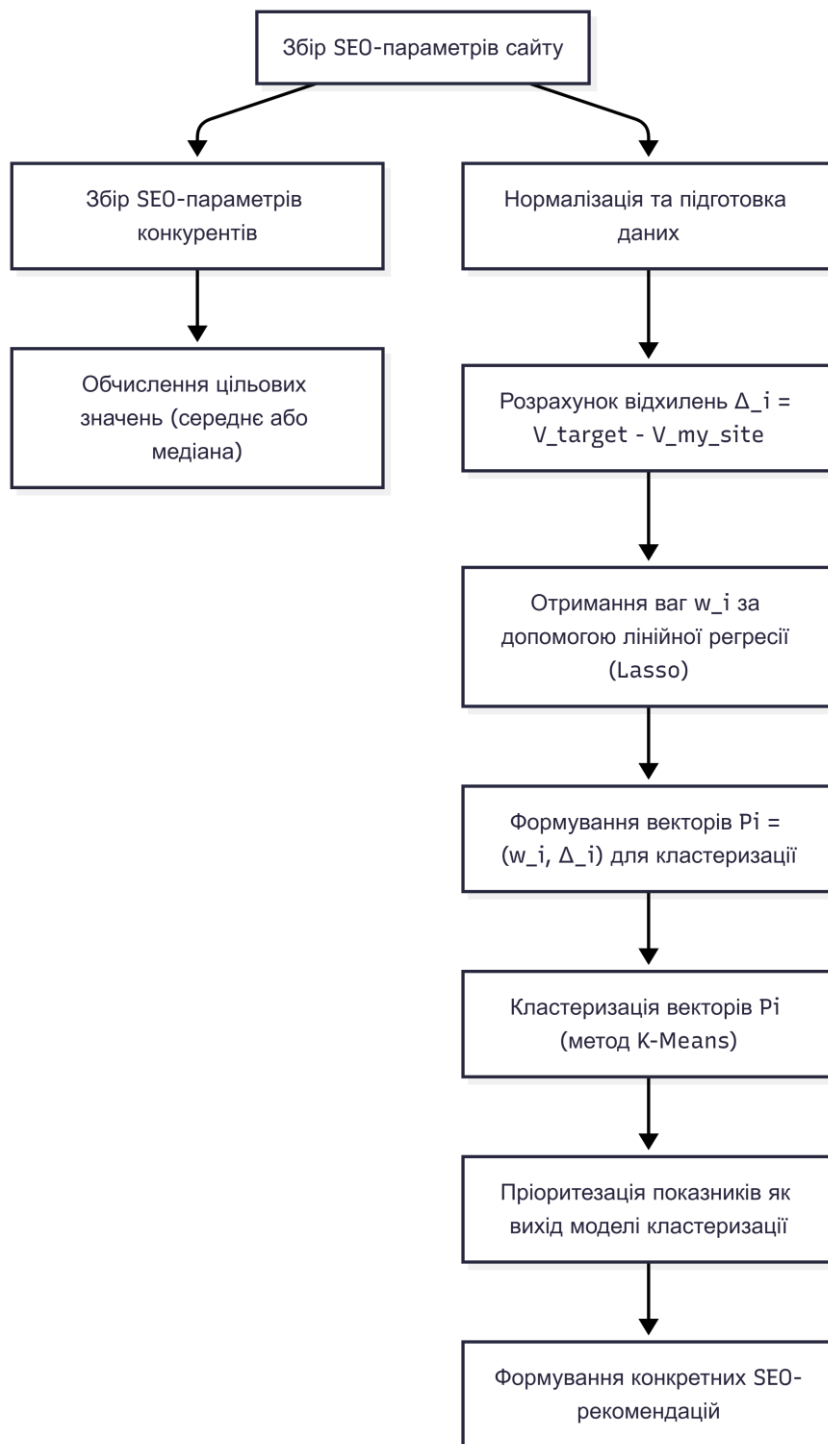


Рисунок 2.3 – Алгоритм формування рекомендацій для інтелектуального SEO-аналізатора

Схема відображає процес формування рекомендацій для SEO-оптимізації сайту. Спершу система збирає параметри власного сайту та конкурентів, обчислює

цільові значення і нормалізує дані. Потім визначаються відхилення та ваги показників за допомогою лінійної регресії, після чого обчислюються значення рекомендацій. Нарешті, система пріоритезує показники за впливом та формує конкретні дії для покращення рейтингу сайту.

2.6 NLP-аналіз контенту

NLP-аналіз контенту є важливою складовою інтелектуального SEO-аналізу, оскільки дозволяє оцінити текст не лише з позиції його довжини чи частотності ключових слів, а передусім із погляду семантичної повноти, контекстуальної релевантності та тематичної структури. На відміну від класичних методів, що оперують кількісними метриками, Natural Language Processing розкриває змістову природу тексту та його відповідність пошуковому наміру користувача.

Процес починається з очищення HTML-коду сторінки та формування текстової репрезентації, що включає нормалізацію символів, видалення службових конструкцій і структурування текстових блоків. Після цього здійснюється токенізація та лематизація, що перетворює слова до їх базових форм та забезпечує узагальнене представлення контенту. Для підвищення точності аналізу видаляються стоп-слова, що не несуть тематичного навантаження.

На наступному етапі визначаються інформаційно значущі елементи тексту. Використовується модель TF-IDF, що обчислює вагу терміна як добуток його локальної частотності та зворотної документної рідкості:

$$\text{TF-IDF}(t, d) = \text{tf}(t, d) \cdot \log \left(\frac{N}{\text{df}(t)} \right) \quad (2.24)$$

де,

- $\text{tf}(t, d)$ — кількість появ терміна t у документі d ,
- N — кількість документів у вибірці,
- $\text{df}(t)$ — кількість документів, у яких зустрічається термін t .

Цей показник дозволяє визначити тематичні домінанти сторінки та оцінити, наскільки контент відповідає цільовій семантиці.

Щоб перейти від окремих слів до змістовних понять, сторінки переводяться у векторний простір за допомогою ембедінг-моделей, наприклад Sentence-BERT. Отриманий вектор представляє зміст тексту у багатовимірному просторі:

$$\vec{v}_{page} = \text{Embed}(T) \quad (2.25)$$

де, T — очищений текст сторінки.

Релевантність сторінки визначається через косинусну подібність між вектором сторінки та середнім вектором ключових тематики:

$$\text{Sim}(A, B) = \frac{A \cdot B}{\|A\| \|B\|} \quad (2.26)$$

Високе значення показує сильну семантичну відповідність пошуковим намірам.

Додатково аналізується набір значущих сутностей (Named Entities), що дозволяє визначити, чи містить текст важливі об'єкти теми — процеси, технології, продукти, географічні назви, технічні параметри тощо. Наявність ключових сутностей є показником повноти розкриття тематики.

Тематична повнота визначається:

$$\text{Coverage} = \frac{|E_{page} \cap E_{topic}|}{|E_{topic}|} \quad (2.27)$$

де,

— E_{page} — множина сутностей, виявлених на сторінці,

— E_{topic} — еталонна множина сутностей за конкурентами.

Низьке охоплення свідчить про неповний або поверхневий контент.

Усі розраховані показники — TF-IDF-ваги, семантична релевантність, повнота тематики, кількість сутностей, індекс читабельності — нормалізуються в діапазоні 0–1 та включаються до загальної системи ваг SEO-моделі:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (2.28)$$

Таким чином NLP-показники стають частиною інтегрованої оцінки ефективності сторінки, впливаючи на формування рекомендацій разом із технічними та структурними параметрами.

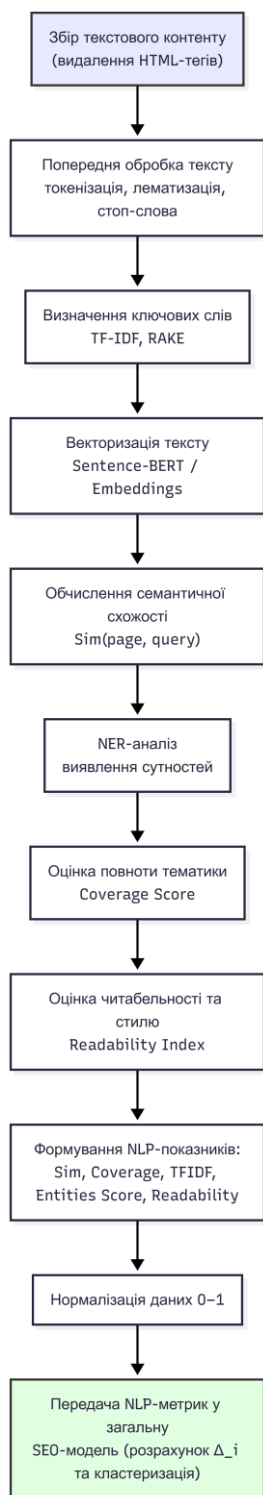


Рисунок 2.4 – Блок-схема NLP-аналізу контенту

На рисунку наведено основні етапи NLP-аналізу контенту, включаючи очищення тексту, семантичну нормалізацію, побудову векторного представлення,

аналіз сутностей та інтеграцію отриманих показників у загальну модель SEO-оптимізації.

Висновки до розділу

У цьому розділі було закладено теоретичні основи та алгоритм для створення інтелектуального ядра програмного засобу для SEO-аналізу. Проведене дослідження показало, що ефективність ранжування вебресурсів у пошукових системах значною мірою залежить від комплексного підходу, який включає технічні, контентні та конкурентні параметри. Технічні показники забезпечують доступність і коректну індексацію сторінок, визначають швидкість завантаження та адаптивність сайту, і їхнє порушення може суттєво знизити видимість ресурсу. Контентні параметри, включаючи семантичну релевантність, унікальність та повноту матеріалів, формують основний фактор оцінки якості вебсторінки та безпосередньо впливають на поведінкові метрики користувачів і позиції в результатах пошуку. Аналіз конкурентів дозволяє визначити відставання власного сайту від медіанних або середніх значень конкурентів за ключовими показниками, що є критично важливим для формування пріоритетних рекомендацій.

Методи збору та нормалізації даних, включаючи розрахунок середніх і медіанних значень, забезпечують надійність отриманих результатів і дозволяють точно оцінювати стан сайту у порівнянні з конкурентами. Лінійна регресія та її варіанти виступають ефективним інструментом визначення ваг кожного показника, що дозволяє автоматично визначати найбільш значущі параметри для подальшого покращення. Отримані ваги, у поєднанні з відхиленням показників від цільових значень, слугують основою для формування пріоритезованих рекомендацій, що підвищує точність і персоналізацію порад для оптимізації сайту. Загалом, результати розділу обґрунтовують використання інтегрованого підходу до SEO-аналізу, що поєднує технічний аудит, оцінку контенту та аналіз конкурентів, і створюють наукову основу для подальшої розробки програмного засобу інтелектуальної генерації рекомендацій.

3 РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАСОБУ

3.1 Архітектура програмного засобу

Архітектура розробленого програмного засобу побудована відповідно до принципів модульності, масштабованості та розширюваності. Система реалізована у вигляді веб-застосунку на основі фреймворку Flask, що забезпечує простоту інтеграції з Python-бібліотеками для парсингу веб-сторінок, обробки даних та реалізації алгоритмів машинного навчання, а також легкість розгортання та подальшого розширення функціоналу без зміни базових компонентів. Архітектура застосунку є багаторівневою та включає рівень представлення, рівень обробки запитів, аналітичний модуль SEO-обчислень, модуль генерації рекомендацій і рівень зберігання даних.

Рівень представлення відповідає за взаємодію користувача із системою та реалізований із використанням HTML5, CSS3 та JavaScript. Він забезпечує введення URL для аналізу, візуалізацію SEO-показників, відображення сформованих рекомендацій.

Рівень маршрутизації та обробки запитів реалізує логіку контролера у файлі `routes.py`. Цей компонент приймає запити від користувача, перевіряє їх валідність та передає на обробку до аналітичного модуля, після чого результати надходять до модуля генерації рекомендацій і повертаються на фронтенд у вигляді HTML-сторінки або JSON. Такий підхід забезпечує чітке розділення логіки та контролю потоків даних у системі.

Аналітичний модуль формує вектор базових SEO-параметрів на основі реальної веб-сторінки. Модуль включає підсистеми парсингу сторінок, аналізу контенту та технічних SEO-параметрів. Парсер отримує HTML-код сторінки та вилучає метадані, заголовки, внутрішні посилання та атрибути зображень. Аналіз контенту оцінює щільність ключових слів, структурну організацію тексту та унікальність інформації, а аналіз технічних параметрів контролює час відповіді сервера, розмір документа, швидкість завантаження сторінки та коректність `canonical`-тегів, `robots.txt` і

sitemap.xml. Результатом є нормалізований вектор SEO-показників, що використовується для формування рекомендацій.

Модуль генерації рекомендацій виконує інтелектуальний аналіз відхилень фактичних параметрів від цільових значень, отриманих на основі середніх або медіанних значень конкурентів. Вектор відхилень множиться на вагові коефіцієнти, визначені за допомогою лінійної регресії, що дозволяє формувати пріоритетні рекомендації для підвищення ефективності SEO. Результатом роботи модуля є текстові рекомендації, оновлений вектор оптимізованих показників та оцінка покращення за допомогою косинусної схожості.

Рівень зберігання даних реалізований на основі легковагової бази SQLite, яка забезпечує збереження результатів аналізу для різних URL, історію змін SEO-параметрів, відстеження позицій у пошуковій видачі та логування роботи системи. Це дозволяє формувати динаміку оптимізації і здійснювати статистичний аналіз покращень.

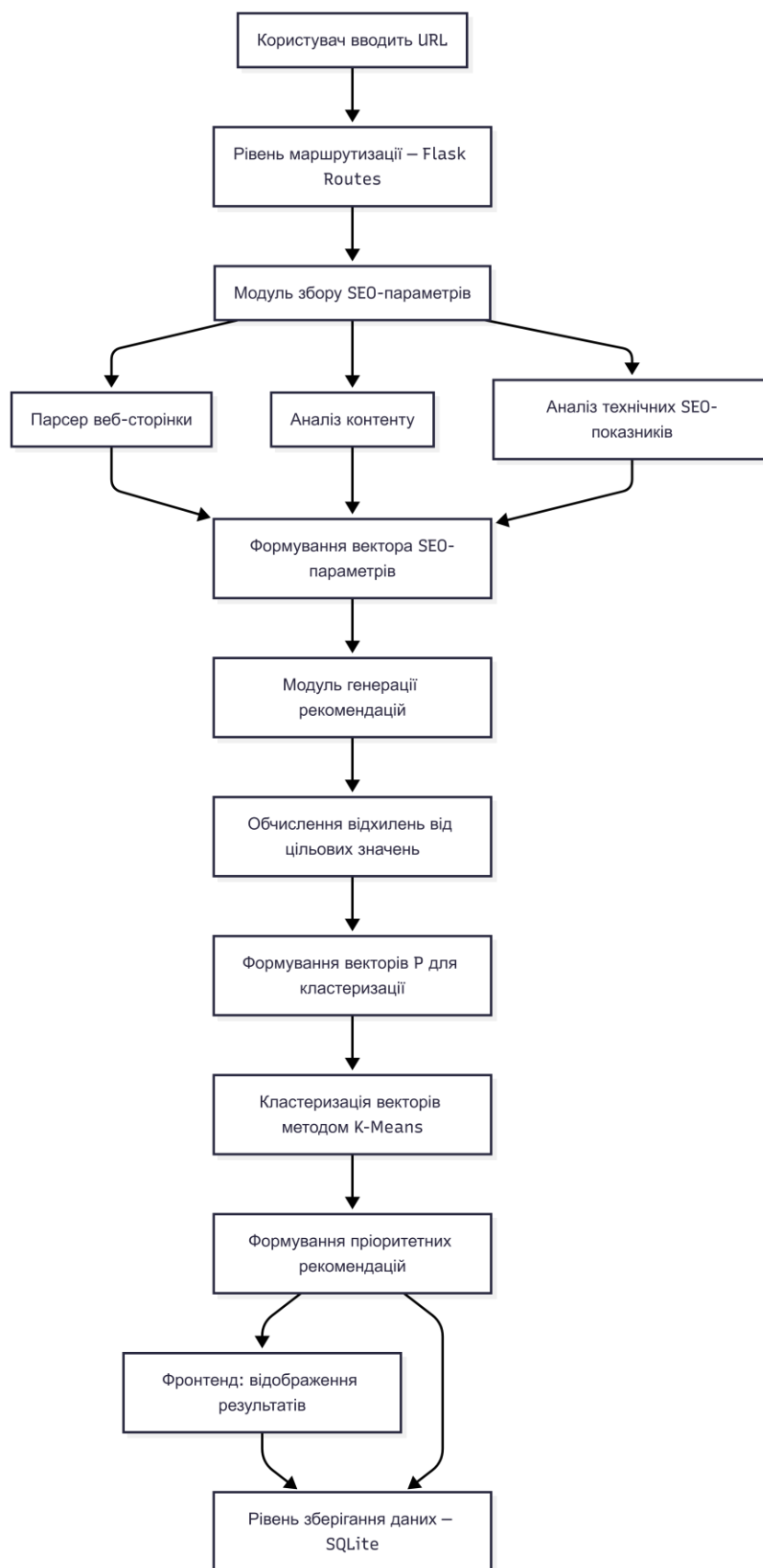


Рисунок 3.1 – Архітектура системи та потік даних між модулями

Архітектура забезпечує послідовний потік даних: користувач вводить URL, який передається роутером до аналітичного модуля, де формується вектор SEO-параметрів. Модуль рекомендацій обчислює відхилення та формує оптимізований вектор, після чого результати повертаються на фронтенд і зберігаються у базі даних. Така структура дозволяє легко оновлювати компоненти системи, підключати нові аналітичні модулі та алгоритми, що забезпечує стабільну і передбачувану роботу програмного засобу.

3.2 Вибір технологій і середовища розробки

Розробка програмного засобу передбачає використання сучасних, ефективних і надійних технологій, що забезпечують швидке та гнучке створення веб-застосунку для аналітики SEO-показників. Під час вибору технологій керувалися кількома ключовими критеріями: можливість масштабування системи, модульність та простота інтеграції нових компонентів, активна підтримка спільноти розробників та наявність документації.

Серверна частина застосунку реалізована на мові Python із використанням фреймворку Flask. Python є високорівневою мовою програмування, що відзначається простотою синтаксису та широким спектром бібліотек для обробки даних і аналітики. Flask, у свою чергу, дозволяє будувати веб-застосунки за принципом мікрофреймворку, що надає розробнику максимальну гнучкість у побудові структури проекту та контролі потоків даних. Основною перевагою Flask є його легковаговість та відсутність надмірної складності, що дозволяє швидко інтегрувати додаткові модулі, зокрема аналітичні та генерації рекомендацій, без зміни базової архітектури. Такий підхід забезпечує модульність і масштабованість, що є важливими для системи, яка може з часом розширюватися новими алгоритмами SEO-аналізу.

Взаємодія користувача із системою відбувається через веб-інтерфейс, створений за допомогою HTML5, CSS3 та JavaScript. Ці технології дозволяють реалізувати адаптивний та інтуїтивно зрозумілий інтерфейс, що підтримує різні

пристрої та дозволяє користувачам без спеціальних технічних знань працювати з аналітичними даними. Для візуалізації показників застосовані сучасні бібліотеки, такі як Chart.js та D3.js, які забезпечують побудову інтерактивних графіків, діаграм та інших візуальних елементів, що відображають SEO-показники. Такий підхід дозволяє наочно демонструвати користувачеві зміни в параметрах сайту після застосування рекомендацій.

Аналітичні модулі, що реалізують збір і обробку SEO-показників, базуються на бібліотеках Python requests та BeautifulSoup для отримання та парсингу HTML-коду сторінок. Requests забезпечує стабільне і гнучке отримання даних із веб-ресурсів, а BeautifulSoup дозволяє ефективно виділяти метадані, заголовки, посилання та атрибути зображень. Для математичної обробки даних та обчислення векторних характеристик SEO-показників використовуються NumPy та SciPy, що забезпечує швидке та точне виконання розрахунків. Регулярні вирази (re) застосовуються для пошуку та обробки ключових слів, що дозволяє виконувати детальний контент-аналіз.

У випадках, коли веб-сторінка генерується динамічно за допомогою JavaScript, застосовуються інструменти автоматизації браузера, такі як Selenium або Playwright, що дозволяє отримувати фінальний HTML-код після виконання скриптів на стороні клієнта. Це забезпечує правильне формування вектора SEO-параметрів, незалежно від способу рендерингу сторінки.

Для зберігання результатів аналізу обрано SQLite, що є легковаговою базою даних, достатньою для потреб системи. Вона дозволяє зберігати історію SEO-показників, сформовані рекомендації, а також логувати роботу системи. Такий вибір обумовлений простотою інтеграції та відсутністю необхідності розгортання складного серверного середовища баз даних на початкових етапах розробки.

Процес розробки відбувається в інтегрованому середовищі програмування PyCharm або Visual Studio Code, що забезпечує зручність редагування коду, налагодження та інтеграцію з системою контролю версій Git. Використання Git

дозволяє керувати версіями проекту, ефективно відслідковувати зміни та забезпечувати колаборацію між кількома розробниками. Для ізоляції залежностей та пакетів застосовується віртуальне середовище (venv або conda), що гарантує сумісність версій бібліотек та стабільність роботи системи при перенесенні на інші машини.

Вибір зазначених технологій забезпечує баланс між простотою розробки та ефективністю роботи застосунку. Комбінація Python та Flask дозволяє швидко реалізувати серверну логіку і підтримувати модульність системи, а сучасні front-end технології і бібліотеки візуалізації гарантують інтуїтивне сприйняття інформації користувачем. Застосування аналітичних бібліотек дозволяє виконувати точні розрахунки векторів SEO-показників та генерувати обґрунтовані рекомендації.

3.3 Опис структурних модулів системи

Програмний засіб побудований на основі модульної архітектури, що забезпечує високу гнучкість, масштабованість та можливість поступового розвитку системи без порушення її базової функціональності. Модульний підхід дозволяє розмежовувати функціональні обов'язки компонентів, забезпечувати ізоляцію помилок, спрощувати тестування та полегшувати подальшу інтеграцію нових алгоритмів або аналітичних інструментів. Така архітектура відповідає сучасним вимогам до веб-застосунків і систем інтелектуальної аналітики, де важливо підтримувати стабільність роботи при постійному розширенні функціоналу.

Центральним елементом системи є модуль взаємодії з користувачем. Він реалізує інтерфейс, через який користувач може вводити URL веб-ресурсу для аналізу, переглядати результати SEO-перевірки та отримувати рекомендації щодо оптимізації. Інтерфейс створений із застосуванням сучасних веб-технологій — HTML5, CSS3 та JavaScript, що дозволяє зробити його адаптивним та зручним для використання на різних пристроях, включаючи персональні комп'ютери та мобільні телефони. Використання бібліотек Chart.js і D3.js забезпечує інтерактивну візуалізацію результатів, дозволяючи користувачеві оцінювати не лише поточний

стан веб-ресурсу, а й прогнозовані зміни після застосування рекомендацій. Таке представлення інформації підвищує інформативність системи, робить результати більш наочними та дозволяє швидко приймати рішення щодо оптимізації контенту та структури сайту.

На серверному боці програмного засобу розташований модуль маршрутизації та обробки запитів, реалізований у рамках фреймворку Flask. Він виконує роль контролера, координуючи обробку запитів від користувача, взаємодію з аналітичними модулями та генерацію відповідей. Після отримання URL-адреси від користувача сервер здійснює перевірку коректності даних і передає їх у модуль збору та аналізу SEO-показників. Роутер забезпечує керування потоками даних між фронтендом та підсистемами аналізу, а також передає отримані результати в модуль генерації рекомендацій і базу даних для збереження історії. Така організація дозволяє відокремити логіку обробки даних від представлення інформації, що підвищує стабільність та передбачуваність роботи системи.

Модуль збору та аналізу SEO-показників є ключовим у структурі системи і формує комплексний вектор параметрів SEO для кожного веб-ресурсу. Спершу здійснюється отримання HTML-коду сторінки за допомогою бібліотеки requests, що забезпечує надійний та гнучкий спосіб взаємодії з веб-ресурсами. Далі BeautifulSoup виконує парсинг документа, виділяючи метадані, заголовки, посилання та атрибути зображень. Цей етап дозволяє оцінити якість структури сторінки, наявність важливих тегів та відповідність контенту основним стандартам SEO.

Особливу увагу система приділяє аналізу контенту, що включає розрахунок щільності ключових слів, перевірку логічної структури тексту та виявлення дубльованих блоків. Цей процес дозволяє оцінити, наскільки ефективно контент веб-ресурсу взаємодіє з пошуковими системами і чи відповідає він сучасним рекомендаціям щодо оптимізації текстових елементів. Крім того, модуль виконує технічний аналіз сторінки, що включає оцінку часу завантаження ресурсу, розмір HTML-документа, коректність canonical-тегів, наявність файлів robots.txt та

sitemap.xml. Ці показники важливі для ранжування сторінок у пошукових системах та визначення потенційних технічних проблем, що можуть впливати на видимість сайту.

Результатом роботи модуля є вектор SEO-параметрів, який передається у модуль генерації рекомендацій. Цей модуль виконує інтелектуальний аналіз отриманих даних та формує пріоретизовані рекомендації. Для цього застосовуються евристичні правила SEO-оптимізації, контент-аналіз та семантична обробка ключових слів. Модуль обчислює відхилення поточного стану від оптимального і пропонує конкретні кроки для покращення SEO-показників. На виході користувач отримує пріоретизовані рекомендації та текстові поради, що пояснюють, які зміни слід внести до структури та контенту сайту для досягнення максимальної ефективності.

Збереження результатів та ведення історії виконуються за допомогою бази даних SQLite, яка забезпечує надійне та структуроване зберігання аналітичних даних. Тут фіксуються результати аналізу, сформовані рекомендації та журнали роботи системи. Це дозволяє відстежувати динаміку змін SEO-показників, оцінювати ефективність проведених оптимізацій та формувати статистику для користувача. База даних також забезпечує можливість подальшого аналізу історичних даних і створення звітів для стратегічного планування покращення веб-ресурсів.

Взаємодія між модулями реалізована через чітко визначені потоки даних: користувач вводить URL, сервер маршрутизує запит до аналітичного модуля, формується вектор параметрів, модуль рекомендацій генерує прогнозований вектор, а результати зберігаються в базі та передаються на фронтенд для візуалізації. Така організація потоків дозволяє забезпечити ефективну, стабільну та передбачувану роботу системи, а також легко інтегрувати нові функції та аналітичні методи без порушення роботи інших компонентів.

Модульна структура системи дозволяє гнучко реагувати на зміни в вимогах користувачів і стандартах SEO. Наприклад, можлива інтеграція додаткових алгоритмів аналізу контенту, підтримка нових методів візуалізації або підключення

більш потужної бази даних для масштабування. Такий підхід гарантує, що буде ефективно працювати у різних умовах та залишатиметься сучасним інструментом для аналізу і оптимізації веб-ресурсів.

3.4 Використовувані API та інтерфейси для збору SEO-параметрів

Для реалізації функціоналу інтелектуального аналізу SEO-параметрів у розробленому програмному засобі використовується комплекс API, які забезпечують автоматизацію процесів збору, обробки та аналізу даних. Ця інтеграція дозволяє системі працювати автономно, зменшує ручну роботу користувача та забезпечує високу точність обчислень. Основними API, використаними у системі, є Flask API, Requests API, BeautifulSoup (bs4) API та Google PageSpeed Insights API. Кожен із цих інтерфейсів виконує специфічні функції, необхідні для роботи відповідних модулів програмного засобу, а їхнє поєднання створює потужну, модульну та масштабовану систему.

Flask API є базовою складовою веб-застосунку та забезпечує організацію всіх процесів взаємодії користувача з системою. Через Flask API реалізована маршрутизація HTTP-запитів, яка дозволяє приймати введені користувачем URL-адреси, проводити валідацію введених даних, ініціювати процес SEO-аналізу та передавати отримані результати в модуль генерації рекомендацій. Крім того, Flask забезпечує обробку відповідей у форматі HTML та JSON, що дозволяє інтегрувати систему з різними фронтенд-компонентами. Такий підхід гарантує масштабованість веб-застосунку та спрощує розширення функціоналу шляхом додавання нових модулів аналітики або джерел даних без зміни базової архітектури.

Requests API використовується для надійного та ефективного отримання даних із вебресурсів, що підлягають аналізу. Цей інтерфейс дозволяє надсилати HTTP-запити до сторінок, отримувати їхній HTML-код, доступ до метаданих, внутрішніх та зовнішніх посилань, а також завантажувати ресурси, необхідні для подальшого аналізу. Requests API забезпечує контроль над таймаутами, обробку помилок та

редиректів, що робить процес збору даних більш стабільним. Використання Requests особливо важливе при аналізі великої кількості сторінок, оскільки дозволяє автоматизувати парсинг та підготовку даних для аналітичного модуля без потреби ручного втручання.

BeautifulSoup (bs4) API відповідає за парсинг та структурування отриманого HTML-контенту. Завдяки його можливостям здійснюється виділення та обробка метаданих сторінок, заголовків H1–H6, внутрішніх та зовнішніх посилань, зображень і атрибутів alt, а також інших елементів, які впливають на он-пейдж SEO. BeautifulSoup забезпечує створення структурованих об'єктів, зручних для подальшої обробки та аналізу алгоритмами системи. Він також дозволяє працювати з некоректним або неповним HTML-кодом, який часто зустрічається на реальних вебресурсах, що підвищує надійність збору даних та точність подальших розрахунків SEO-показників.

Google PageSpeed Insights API забезпечує збір технічних метрик продуктивності та швидкості завантаження вебсторінок. Через цей API система отримує ключові показники, такі як Largest Contentful Paint, First Input Delay, Cumulative Layout Shift, а також загальні оцінки продуктивності, доступності та SEO. PageSpeed Insights дозволяє автоматично порівнювати отримані результати з рекомендованими значеннями, виявляти технічні недоліки, що негативно впливають на ранжування, та формувати пріоритетні рекомендації щодо їх усунення. Інтеграція цього API дозволяє отримати об'єктивні та кількісні дані для оцінки технічного стану сайту, що є критично важливим для повноцінного інтелектуального аналізу.

SerpAPI використовується для автоматизованого збору інформації про конкурентів у пошуковій видачі Google. Цей API дозволяє отримувати структуровані SERP-дані, включно з позиціями конкурентних сторінок, використаними ними заголовками (title), метаописами (meta description), частотою появи ключових слів у сніпетах, а також URL найближчих конкурентів. Крім того, SerpAPI надає дані про

показники, які формуються на рівні пошукової системи: featured snippets, локальну видачу, часті запитання, індексовані підсторінки та типи результатів.

Інтеграція SerpAPI дозволяє системі формувати репрезентативну вибірку конкурентів для аналізу, автоматично завантажувати їхні сторінки, обчислювати середні та медіанні цільові значення SEO-показників і визначати позиційні відхилення сайту користувача від ринку. Таким чином, SerpAPI забезпечує повноцінну реалізацію модуля аналізу конкурентів та підвищує точність побудови рекомендацій шляхом урахування реальних умов конкурентного середовища.

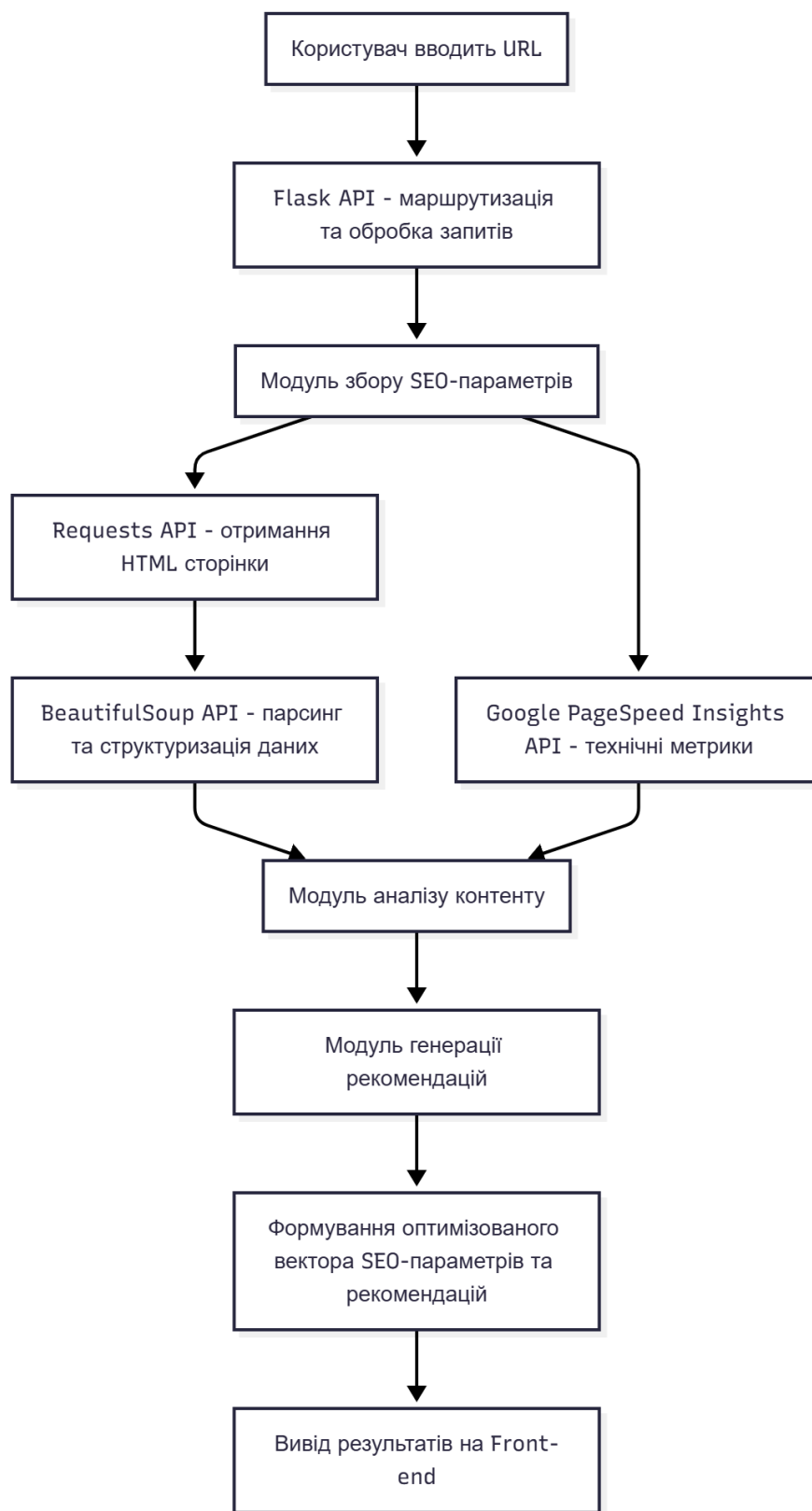


Рисунок 3.2 – Архітектура взаємодії API з модулями програмного засобу для збору та аналізу SEO-параметрів

У комплексі ці API забезпечують повний цикл збору, обробки та аналізу даних: від отримання HTML-коду і структурованих метрик до розрахунку показників контенту та технічних характеристик, формування вектора SEO-параметрів, порівняння з цільовими значеннями і генерації персоналізованих рекомендацій. Завдяки такому підходу система здатна автоматично оцінювати поточний стан вебресурсу, визначати слабкі місця в SEO, формувати пріоритетні дії для оптимізації та надавати користувачу структуровані та зрозумілі поради. Поєднання внутрішніх API (Flask, Requests, BeautifulSoup) із стороннім сервісом (Google PageSpeed Insights) забезпечує модульність, масштабованість, можливість подальшого розширення системи та інтеграцію нових джерел даних без порушення існуючої архітектури.

3.5 Інтерфейс користувача та логіка взаємодії

Ефективність інтелектуального програмного засобу визначається не лише точністю алгоритмів аналізу SEO-параметрів, але й високою якістю інтерфейсу користувача (UI) та логікою взаємодії (UX). Інтерфейс розроблено таким чином, щоб перетворювати складні аналітичні дані на інтуїтивно зрозумілі та практично застосовні рекомендації. Головною метою UX є забезпечення швидкого доступу до ключових показників, прогнозованих змін рейтингу та пріоритетних дій без необхідності глибокого розуміння алгоритмів або ручного аналізу великих обсягів даних.

Процес взаємодії користувача з системою починається з введення URL досліджуваного сайту та набору ключових слів, що є базою для формування семантичного ядра. У нашому випадку користувач вводить 10 основних ключових слів, на основі яких система через API виконує автоматичний пошук конкурентів у топ-50 видачі SERP. Інтеграція з Flask API забезпечує маршрутизацію та обробку запитів всередині веб-застосунку, Requests API використовується для отримання HTML-коду сторінок конкурентів, а BeautifulSoup забезпечує парсинг контенту, включаючи метадані (title, description, keywords), заголовки H1–H6, внутрішні та зовнішні посилання, зображення та атрибути alt. Додатково, для оцінки швидкості

завантаження сторінок та Core Web Vitals використовується Google PageSpeed Insights API, що дозволяє отримати стандартизовані метрики, які критично впливають на ранжування сайту.

Після збору даних з конкурентів система формує структуровану базу, що включає до 50 конкурентних сайтів та 16 SEO-показників для кожного з них. Кожен SEO-показник обробляється та нормалізується для забезпечення порівнянності між різними ресурсами. Важливим етапом є обчислення інтегрованого значення кожного SEO-показника через вагові коефіцієнти, отримані за допомогою лінійної регресії або інших методів машинного аналізу. Результатом цього процесу стає вектор ваг, який відображає значущість кожного показника для рейтингу сторінки.

Далі система розраховує SEO-параметри для досліджуваного сайту та визначає відхилення від інтегрованих значень конкурентів для кожного показника. Користувач отримує детальний звіт у вигляді інтерактивної таблиці, де кожна позиція містить опис проблеми, прогнозований ефект від оптимізації Δ_R , локалізацію на сторінці та покрокові рекомендації для виправлення. Таким чином, користувач може швидко визначити найбільш критичні для SEO аспекти та сконцентрувати ресурси на їх оптимізації.

Інтерфейс організовано навколо декількох взаємопов'язаних зон для забезпечення повного циклу роботи:

- Dashboard (Панель керування) – надає загальний огляд стану SEO, інтерактивні графіки та гістограми, що показують динаміку ключових метрик, таких як Core Web Vitals, швидкість завантаження сторінки, структурна організація контенту та пріоритетність проблем;
- Звіт та Рекомендації – містить детальну таблицю пріоритетів, яка сортується за показником P_{final} , формованим ML Core. Кожна проблема розкривається у картці з детальною інструкцією та локалізацією на сторінці, що забезпечує максимально конкретну та практичну інформацію для користувача;

— Інструменти аналізу та порівняння – дозволяють проводити глибокий аналіз конкретних кластерів SEO-показників, оцінювати відхилення вектора A досліджуваного сайту від конкурентів та відслідковувати зміни у часі.

Ключовим елементом UX є петля зворотного зв'язку (Feedback Loop). Коли користувач позначає завдання як виконане, система ініціює повторний аналіз SEO-параметрів та перевірку змін рейтингу через інтеграцію з GSC/GA. Результати цієї валідації автоматично використовуються для донавчання моделі, підвищуючи точність прогнозів P_{final} та адаптивність системи до змін алгоритмів пошукових систем. Це забезпечує динамічну взаємодію, де ПЗ не є статичним інструментом, а стає інтерактивним аналітичним асистентом.

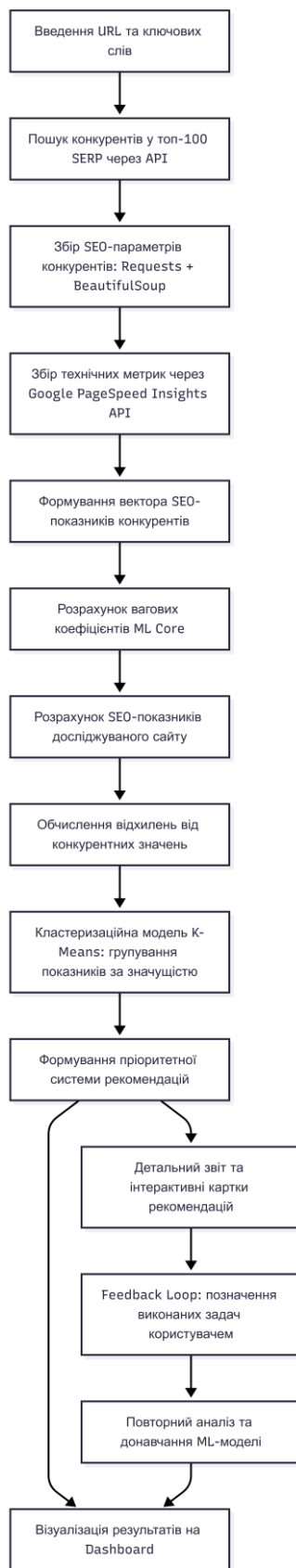


Рисунок 3.3 – Етапи роботи ПЗ

Рисунок 3.3 демонструє повний цикл роботи інтелектуального SEO-помічника від введення URL та ключових слів до формування пріоритетних рекомендацій. Дані конкурентів збираються через API, нормалізуються та оцінюються за допомогою вагових коефіцієнтів, що дозволяє отримати інтегрований рейтинг SEO-параметрів. Петля зворотного зв'язку забезпечує адаптивність системи та постійне покращення точності прогнозів.

У такий спосіб інтерфейс та логіка взаємодії забезпечують повний цикл роботи користувача: від введення ключових слів і визначення конкурентів до оцінки SEO-параметрів, формування вагових коефіцієнтів, аналізу відхилень та пріоритетних рекомендацій. Користувач отримує максимально прозорий та практично корисний інструмент для прийняття рішень щодо оптимізації вебресурсу та підвищення його позицій у пошуковій видачі.

3.6 Експериментальні дослідження роботи системи

Експериментальні дослідження були спрямовані на комплексну оцінку функціональності розробленого програмного засобу та перевірку його здатності здійснювати інтелектуальний SEO-аналіз з формуванням пріоритетних рекомендацій для користувача. У процесі дослідження проводився повний цикл обробки даних, що починався зі збору інформації про досліджуваний вебресурс та конкурентні сайти, включаючи їх позиції в пошуковій видачі та 20 ключових SEO-показників. Далі зібрані дані проходили етап нормалізації та інтеграції, що дозволяло уніфікувати показники та підготувати їх для подальшого аналітичного опрацювання. На наступному етапі здійснювалось обчислення вагових коефіцієнтів SEO-параметрів, що давало змогу визначити вплив кожного показника на загальний рейтинг ресурсу. Паралельно виконувалась оцінка відхилень SEO-показників досліджуваного сайту від інтегральні цільові значення SEO-показників, що дозволяло визначити найбільш критичні області для оптимізації. На основі отриманих ваг та відхилень система

формувала адаптивні рекомендації, які розташовувалися за пріоритетом їхнього потенційного впливу на покращення видимості сайту, забезпечуючи користувачу зрозумілу та ефективну аналітичну інформацію для практичного впровадження змін.

Таблиця 3.1 – Етапи роботи додатку

Етап	Вхідні дані	Обробка	Результат
Збір даних	Ключові слова, URL сайту, конкурентні сайти	API-запити	SEO-параметри для досліджуваного та конкурентних сайтів
Нормалізація та інтеграція	SEO-параметри сайтів	Масштабування значень, обчислення інтегрованих показників	Уніфікований вектор показників
Розрахунок ваг	Вектори конкурентів	Лінійна регресія, методи машинного навчання	Вагові коефіцієнти SEO-параметрів
Оцінка відхилень	Вектор досліджуваного сайту	Розрахунок різниці фактичних та цільових значень	Вектор відхилень
Формування рекомендацій	Вектор відхилень, вагові коефіцієнти	Обробка через кластерну модель K- Means	Пріоритетні рекомендації для оптимізації
Адаптивний контроль	Реалізація рекомендацій	Повторний збір даних, оновлення ваг	Актуалізований вектор рекомендацій та контроль ефективності

На початковому етапі система здійснює збір даних за визначеним набором ключових слів, що характеризують тематику досліджуваного сайту. За допомогою

інтегрованих API і парсерів відбувається автоматична ідентифікація конкурентних ресурсів у пошуковій видачі та збір структурованих SEO-параметрів для кожного з них. Отримані дані агрегуються у внутрішню аналітичну структуру, що дозволяє коректно порівнювати параметри досліджуваного сайту з показниками конкурентів, забезпечуючи основу для подальшого аналізу.

Вагові коефіцієнти для кожного SEO-параметра визначаються методами машинного аналізу, зокрема за допомогою лінійної регресії та її модифікацій. Кожна вага відображає відносний внесок параметра у формування загальної релевантності та позиції сайту у пошуковій видачі. Процес обчислення включає навчання моделі на агрегованих даних конкурентів із урахуванням значущості технічних та контентних показників, що дозволяє сформувати інтегровану систему пріоритетів.

Для кожного параметра визначається відхилення фактичного значення досліджуваного сайту від інтегрованого значення конкурентного середовища. Цей процес дозволяє оцінити ступінь недоліків або переваг сайту в порівнянні з конкурентами. Відхилення формуються як основа для ранжування параметрів за потенційним впливом на рейтинг і виступають критерієм для визначення пріоритетності рекомендацій.

На основі розрахованих ваг і відхилень система генерує пріоритетні рекомендації, ранжуючи їх за очікуваним впливом на рейтинг. Для кожного параметра визначається оптимальна стратегія дій, яка дозволяє максимально ефективно скоригувати виявлені недоліки. Рекомендації формуються у вигляді структурованих елементів із чіткою логікою виконання, що забезпечує користувачу можливість оперативно здійснювати оптимізаційні заходи.

Програмний засіб забезпечує замкнений цикл контролю результатів: після впровадження рекомендацій система може повторно оцінювати зміни SEO-параметрів та оновлювати вагові коефіцієнти, враховуючи нові дані. Така адаптивність дозволяє підтримувати актуальність рекомендацій навіть у разі змін алгоритмів пошукових систем або структури конкурентного середовища.

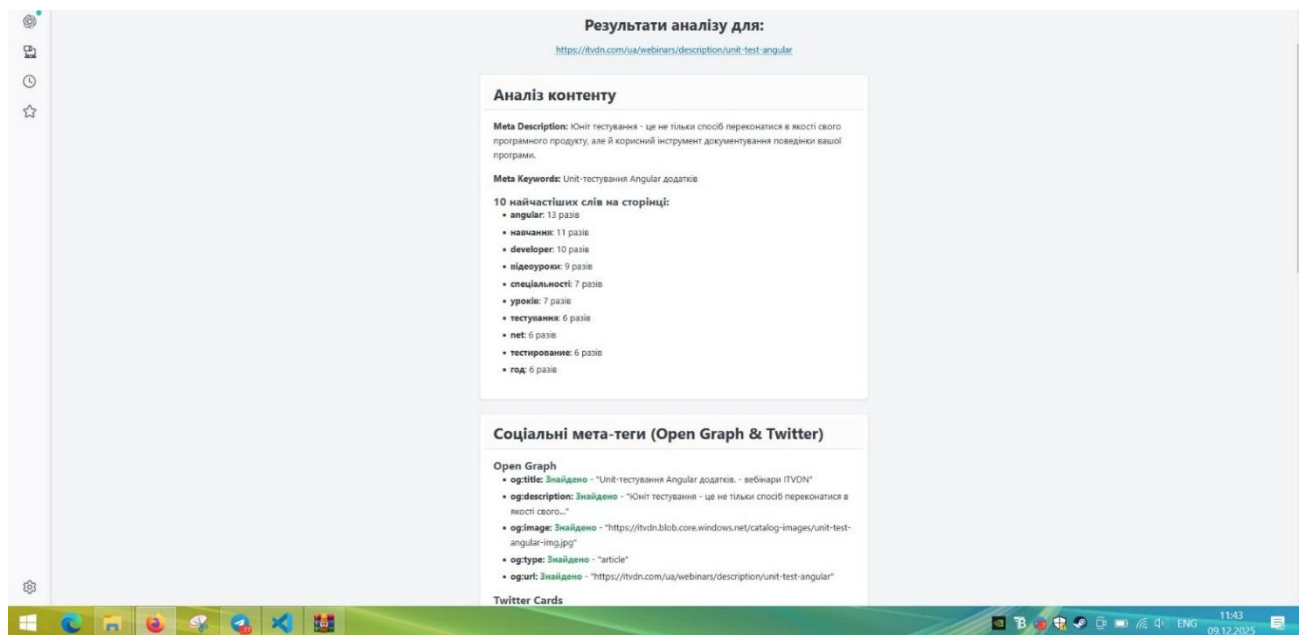


Рисунок 3.4 – Результати аналізу сайту

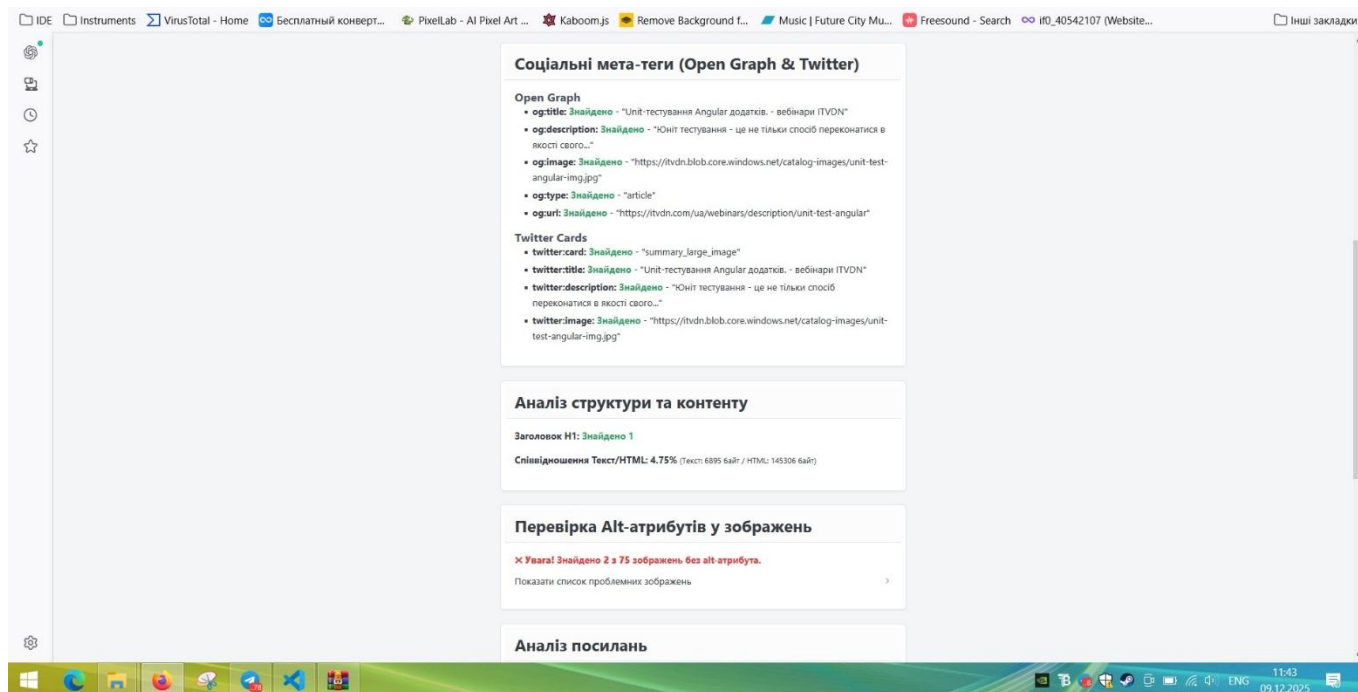


Рисунок 3.5 – Результати аналізу сайту

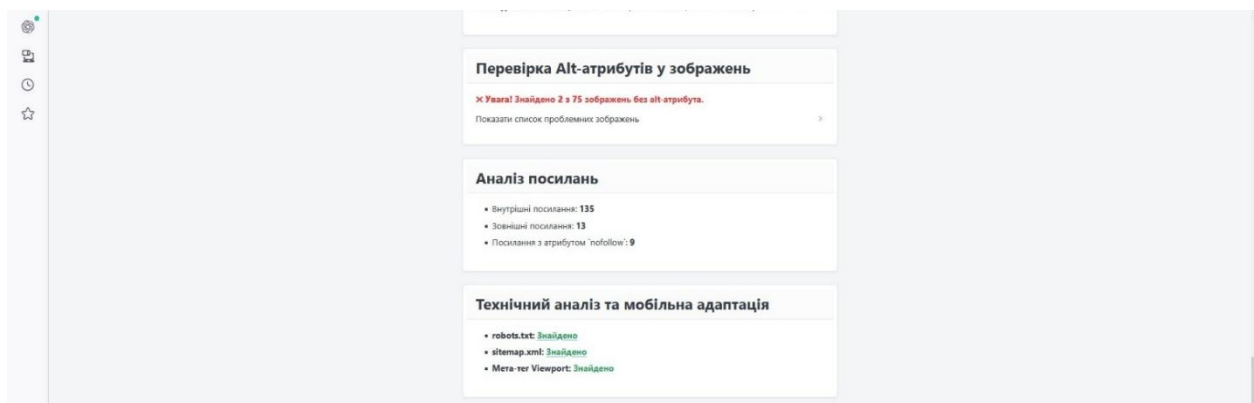


Рисунок 3.6 – Результати аналізу сайту

Проведений аудит веб-сторінки, присвяченої Unit-тестуванню Angular, дозволив здійснити комплексну оцінку її SEO-стану та виявити ключові фактори, що впливають на ранжування та користувацький досвід. Аналіз охоплював контент, технічну структуру, соціальні сигнали та доступність.

1. Контентна релевантність та мета-оптимізація

- Сторінка демонструє високий рівень контентної релевантності до цільових запитів.
- Мета-теги (Meta Description та Keywords), виявлено правильне використання тегів з включенням ключових фраз, таких як "Unit-тестування Angular додатків", що забезпечує точне позиціонування сторінки в пошуковій видачі.
- Семантичне ядро, аналіз показав високу частоту використання цільових ключових слів (angular, developer, тестування), що підтверджує тематичну відповідність контенту.
- Структура заголовків, використано один заголовок H1, що відповідає стандартам ієрархії контенту.

2. Технічний стан та сканування

Технічна готовність сторінки до індексації оцінюється як задовільна.

- Файли сканування, наявність robots.txt та sitemap.xml забезпечує контроль сканування веб-сторінок пошуковими роботами.

- Мобільна адаптація, використання мета-тегу Viewport підтверджує базову мобільну адаптацію, що відповідає принципу Mobile-First.

3. Оптимізація для соціальних мереж

Сторінка адаптована для поширення контенту в соціальних мережах, що сприяє залученню додаткового трафіку.

- Open Graph та Twitter Cards: всі необхідні теги (og:title, og:description, og:image) реалізовані, забезпечуючи коректне відображення посилань у Facebook, LinkedIn та X (Twitter).

4. Виявлені проблеми та недоліки

Аудит виявив кілька проблемних аспектів, що знижують загальний SEO-потенціал сторінки.

- Відсутність Alt-атрибутів, 2 із 75 зображень не мають Alt-атрибутів, що ускладнює індексацію зображень пошуковими системами та порушує вимоги доступності (WCAG).
- Низьке співвідношення текст/HTML, показник становить лише 4.75%, що свідчить про перевагу коду над унікальним текстовим контентом.
- Некоректні атрибути Nofollow, виявлено 9 посилань з помилковим використанням атрибута nofollow, що може порушувати оптимальний розподіл посилальної ваги між сторінками.

5. Рекомендації

На основі результатів аудиту сформовано пріоритетні завдання для оптимізації:

- Виправлення Alt-атрибутів, додати описові Alt-тексти до всіх зображень без атрибута.
- Оптимізація контенту, збільшити обсяг унікального та інформативного тексту для підвищення співвідношення текст/HTML.

- Аудит внутрішніх посилань, перевірити та скоригувати атрибути nofollow для забезпечення правильної внутрішньої перелінковки.

Експериментальні дослідження підтверджують, що система здатна ефективно:

- інтегрувати різні типи SEO-параметрів у єдину аналітичну модель;
- формувати вагові оцінки параметрів та використовувати їх для пріоритезації дій;
- оцінювати відхилення від конкурентів як основу для формування стратегічних рекомендацій;
- надавати користувачу структуровані та логічно послідовні інструкції щодо оптимізації сайту.

Таким чином, система демонструє високу ефективність у процесі інтелектуального SEO-аналізу, дозволяє формувати пріоритетні та адаптивні рекомендації, а також забезпечує постійне удосконалення моделей оцінки показників за рахунок циклічного контролю і обробки нових даних.

3.7 Моделі для роботи з сайтами конкурентами

Моделі для роботи з сайтами конкурентів у межах інтелектуального SEO-аналізатора виступають критичною складовою аналітичної системи, що забезпечує комплексну оцінку конкурентного середовища та формування структурованих рекомендацій для оптимізації досліджуваного вебресурсу. Основною метою цих моделей є створення інтегрованого аналітичного середовища, яке дозволяє не лише порівнювати сайти за низкою SEO-показників, а й визначати пріоритети дій щодо підвищення позицій у пошукових системах.

Перший етап роботи моделі полягає в ідентифікації конкурентів на основі ключових слів, що характеризують тематичну сферу досліджуваного сайту. Для цього система здійснює автоматичний збір URL-адрес з пошукової видачі, що відображають сайти з високим рівнем релевантності для заданих ключових слів. Процедура забезпечує формування первинного пулу конкурентів, який відображає

реальну конкуренцію в конкретній ніші, і закладає основу для подальшого структурованого аналізу.

На наступному етапі здійснюється комплексний збір та структуризація SEO-показників для кожного з ідентифікованих конкурентів. Модель інтегрує дані з численних джерел та API, що забезпечує повне охоплення технічних, контентних та он-пейдж параметрів. Для кожного ресурсу формується повний вектор показників, де кожен елемент відповідає окремому SEO-параметру, а всі вектори об'єднуються в матрицю конкурентів. Така організація даних дозволяє виконувати паралельне порівняння та комплексну оцінку кожного сайту щодо досліджуваного ресурсу.

Далі здійснюється нормалізація та інтеграція показників, що має критичне значення для коректного порівняння даних з різними шкалами та одиницями вимірювання. Значення всіх параметрів приводяться до єдиної шкали, що забезпечує математично обґрунтоване порівняння та усуває дисбаланс між різними типами SEO-показників. Одночасно для кожного параметра обчислюються вагові коефіцієнти, які відображають його вплив на загальний рейтинг сайту. Ваги формуються за допомогою методів машинного навчання, зокрема лінійної регресії та її різновидів, що дозволяє об'єктивно визначати значущість кожного показника в контексті впливу на рейтинг сайту.

Модель також передбачає динамічне оновлення даних конкурентів, що дозволяє адаптувати аналітику до змін у пошуковій видачі. При зміні позицій конкурентів або оновленні їх SEO-параметрів система автоматично коригує інтегровані значення, забезпечуючи актуальність рекомендацій у реальному часі. Це створює умови для постійного моніторингу конкурентного середовища та підвищує точність пріоритезації дій для досліджуваного сайту.

Функціонально моделі забезпечують кілька ключових напрямів роботи: комплексний збір та структурування даних, нормалізацію та інтеграцію SEO-параметрів, обчислення відхилень від конкурентів, визначення вагових коефіцієнтів та генерацію адаптивної системи рекомендацій. Такий підхід гарантує системність і

об'єктивність аналізу, дозволяє користувачеві приймати обґрунтовані рішення щодо оптимізації ресурсу та забезпечує підвищення його конкурентоспроможності у пошуковій видачі. Моделі реалізують логіку аналітичного циклу, що поєднує автоматизацію збору даних з інтегрованою оцінкою значущості параметрів і генерацією практичних, пріоритетних рекомендацій.

Висновки до розділу

У третьому розділі було проведено всебічне обґрунтування архітектури, реалізації та функціонування програмного забезпечення для інтелектуального SEO-аналізу вебресурсів. Розділ охоплює побудову моделей обробки даних, механізми формування вагових коефіцієнтів, методи нормалізації показників, принципи порівняння з конкурентами та організацію інтелектуального модуля рекомендацій. На основі отриманих результатів експериментальних досліджень підтверджено ефективність розробленого рішення та відповідність поставленим вимогам.

По-перше, у підрозділах було детально розглянуто структуру аналізу SEO-параметрів: від збору даних із зовнішніх джерел та API до їх інтеграції у єдину систему. Реалізовані механізми нормалізації дозволяють уніфікувати значення різних показників, що мають різну природу й масштаб, забезпечуючи коректність подальших математичних операцій. На цій основі побудовано моделі формування вагових коефіцієнтів, які визначають значущість кожного окремого параметра в загальному рейтингу сайту.

По-друге, розроблена система моделей роботи з сайтами конкурентів забезпечує автоматизований і структурований процес отримання ключових конкурентів за тематичними запитами, побудову багатовимірної матриці SEO-показників та розрахунок інтегрованих значень, що відображають середній рівень оптимізації у конкурентній ніші. Моделі дають змогу не лише фіксувати окремі значення, а й здійснювати глибокий порівняльний аналіз між досліджуваним сайтом і групою конкурентів.

По-третє, визначено механізм обчислення відхилень параметрів між вебресурсом користувача та інтегрованими значеннями конкурентів, що дозволяє формувати кількісну оцінку розриву за кожним SEO-показником. Системний підхід до обчислення відхилень у поєднанні з ваговими коефіцієнтами дає можливість побудувати пріоритетну матрицю оптимізації, у якій кожен параметр отримує власний коефіцієнт важливості та ступінь критичності.

Проведений цикл тестування продемонстрував коректність обчислення ваг, точність формування рекомендацій та адекватність адаптивної поведінки системи при зміні даних конкурентів. Встановлено, що програмний засіб здатний оперативно оновлювати результати аналізу та автоматично адаптувати підсумкові рекомендації до змін у конкурентному середовищі.

Загалом результати, отримані у третьому розділі, вказують на успішну реалізацію інтелектуальної моделі SEO-аналізу, яка поєднує:

- автоматизований збір даних;
- глибоку структурування та нормалізацію показників;
- побудову моделей конкурентного середовища;
- математичну оцінку відхилень;
- формування ваг та рейтингів;
- генерацію пріоритетних рекомендацій із високим рівнем адаптивності.

Розроблені модулі довели здатність програмного забезпечення забезпечувати комплексну оцінку стану вебресурсу, виявляти ключові проблеми та визначати найбільш ефективні напрями оптимізації. Таким чином, моделі й алгоритми, описані в розділі, створюють основу для подальшої інтеграції аналітичних механізмів у повноцінну систему підтримки прийняття рішень у сфері SEO-оптимізації.

4 АНАЛІЗ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Методи тестування програмного засобу

Тестування програмного засобу є критично важливим етапом розробки, оскільки воно дозволяє оцінити не лише відповідність системи функціональним вимогам, але й ефективність алгоритмів обробки SEO-параметрів, надійність роботи під різним навантаженням, а також стабільність та прогнозованість результатів. Комплексний підхід до тестування дозволяє виявити слабкі місця системи на ранніх етапах, що мінімізує ризики появи критичних помилок у процесі експлуатації.

Для оцінки якості та надійності програмного забезпечення застосовувалися різні методи тестування, які поєднували як перевірку функціональності системи, так і оцінку її продуктивності та стійкості. Одним із основних підходів було використання тестування методом чорного ящика, яке дозволяє оцінити поведінку системи з точки зору користувача, не враховуючи внутрішньої структури коду. Цей метод передбачає перевірку того, наскільки коректно обробляються введені дані, як формуються результати аналізу та рекомендації, і чи відповідають вони заявленим вимогам до функціоналу. За допомогою цього підходу було перевірено, чи система адекватно реагує на різні формати URL, чи коректно формуються вектори SEO-параметрів і чи правильно відображаються результати на інтерфейсі користувача у вигляді таблиць, графіків та діаграм.

Водночас велике значення мало тестування методом білого ящика, яке дозволяє глибоко перевірити внутрішню логіку роботи системи, алгоритми обробки даних та взаємодію компонентів. Завдяки цьому методу було детально проаналізовано роботу алгоритмів парсингу HTML-коду, оцінки щільності ключових слів, обчислення технічних SEO-параметрів та генерації рекомендацій. Тестування білого ящика дозволило переконатися у коректності побудови векторів показників, точності математичних розрахунків і правильності обробки виняткових ситуацій, таких як некоректний HTML-код або відсутність метаданих на сторінці. Крім того, метод білого ящика сприяв оптимізації логіки роботи модулів, що дозволило скоротити час

обробки даних та знизити ризик виникнення помилок при передачі даних між компонентами системи.

Особливу увагу було приділено інтеграційному тестуванню, яке оцінює взаємодію окремих модулів системи та забезпечує перевірку цілісності потоків даних між ними. Під час цього етапу тестування перевірялась взаємодія фронтенд-компонента з бекендом, обмін даними між модулем збору SEO-параметрів і генератором рекомендацій, а також коректність запису та зчитування інформації з бази даних. Такий підхід дозволив виявити потенційні проблеми у синхронізації роботи компонентів, помилки форматування даних та невідповідність очікуваних і фактичних результатів.

Ще одним важливим аспектом тестування була оцінка стійкості та надійності системи за допомогою сценарного та навантажувального тестування. Під час сценарного тестування система обробляла різноманітні веб-ресурси: від невеликих сторінок із простим контентом до великих порталів зі складною структурою і великою кількістю медіафайлів. Це дозволило перевірити, як система справляється з різним обсягом даних, чи не виникають затримки при формуванні векторів SEO-параметрів, і чи забезпечується коректна генерація рекомендацій навіть у складних випадках. Навантажувальне тестування, у свою чергу, дозволило оцінити продуктивність системи при одночасній обробці великої кількості запитів та перевірити, чи витримує вона високе навантаження без втрати даних або падіння стабільності роботи.

Для підвищення об'єктивності оцінки та зниження впливу людського фактора застосовувалось автоматизоване тестування, яке включало виконання скриптів для багаторазового повторення однакових тестових сценаріїв. Це дозволило систематично перевіряти функції збору даних, генерації векторів SEO-параметрів та формування звітів у різних умовах і з різними вхідними даними, забезпечуючи детальний аудит роботи кожного модуля системи.

Таким чином, використання комбінації методів тестування — чорного та білого ящика, інтеграційного, сценарного та автоматизованого тестування — забезпечило всебічну оцінку роботи програмного засобу. Такий комплексний підхід дозволив не лише виявити помилки і недоліки, але й оцінити точність алгоритмів обробки SEO-параметрів, продуктивність системи, стабільність роботи під різними умовами та практичну користь програмного забезпечення для автоматизованого аналізу та оптимізації веб-ресурсів.

4.2 Функціональне тестування модулів

Функціональне тестування програмного засобу є ключовим етапом оцінки готовності системи до реального використання. Основна мета цього тестування полягала у підтвердженні того, що всі функціональні компоненти системи виконують свої завдання коректно, відповідно до технічного завдання, і забезпечують безперебійну взаємодію між собою. Наукова цінність функціонального тестування полягає у перевірці алгоритмічної точності модулів, здатності системи адаптуватися до різних вхідних даних та гарантувати узгодженість результатів у комплексному аналізі веб-ресурсів.

Особливу увагу під час тестування приділяли модулю збору SEO-параметрів, який є базовим для формування аналітичного вектора сторінки. Тестування цього модуля включало оцінку здатності системи обробляти веб-сторінки з різною структурою, враховувати наявність динамічного контенту, мультимедійних елементів, складних таблиць та графіки. Було проведено перевірку точності вилучення метаданих, заголовків різних рівнів, атрибутів зображень, внутрішніх і зовнішніх посилань, а також обробки складних HTML-елементів. Кожен сценарій тестування модулю передбачав порівняння з очікуваними результатами, що дозволило оцінити правильність формування вектора SEO-параметрів та його відповідність ідеальному профілю сторінки. На цьому етапі тестування також виявлялися виняткові ситуації, такі як відсутність ключових метаданих, некоректний

формат HTML або наявність помилок у структурі сторінки, і перевірялося, наскільки система здатна обробити такі випадки без втрати даних або порушення логіки аналізу.

Модуль генерації рекомендацій був протестований на коректність інтерпретації зібраних параметрів та точність оцінки відхилень від оптимальних значень. Тестування включало перевірку алгоритмів семантичного аналізу ключових слів, застосування евристичних правил для формування рекомендацій і побудови оновленого вектора параметрів. Особлива увага приділялась адекватності рекомендацій у випадках неповних або суперечливих даних, а також здатності модуля передбачати ефективні зміни, які користувач може впровадити для оптимізації ресурсу. Під час тестування перевірялась точність математичних розрахунків, коректність формування текстових висновків та візуалізації результатів, що є критично важливим для прийняття обґрунтованих рішень користувачем.

Функціональне тестування рівня представлення (Front-end) проводилось із урахуванням зручності та інтуїтивності інтерфейсу користувача. Було перевірено адаптивність дизайну, коректність відображення таблиць, графіків та діаграм, швидкість реагування на введення URL-адреси, а також обробку некоректних або неповних даних. Крім того, оцінювалась здатність інтерфейсу передавати дані на бекенд та відображати результати обробки без помилок. Результати тестування підтвердили, що інтерфейс є зрозумілим для користувачів із різним рівнем технічної підготовки, стабільно відображає інформацію та забезпечує комфортну взаємодію із системою.

Важливим аспектом функціонального тестування була оцінка взаємодії модулів, що включала аналіз цілісності потоків даних від моменту введення URL користувачем до отримання фінального звіту. Було перевірено коректність передачі даних між фронтом, модулем збору SEO-параметрів, генератором рекомендацій та базою даних. Сценарії тестування передбачали одночасну обробку декількох запитів, перевірку логування результатів, а також перевірку поведінки системи у випадку тимчасових збоїв мережі або некоректних HTML-кодів. Такий підхід

дозволив забезпечити високий рівень надійності та стабільності системи навіть за умов підвищеного навантаження.

Особливу увагу приділено тестуванню роботи з базою даних, де перевірялися операції запису, зчитування та оновлення інформації, формування історії змін SEO-показників та логування дій системи. Це дозволило переконатися, що база даних здатна зберігати всі необхідні дані без втрат, підтримувати цілісність інформації під час одночасного оброблення декількох запитів та формувати статистичні звіти для подальшого аналізу динаміки оптимізації веб-ресурсів.

Функціональне тестування проводилось також із використанням автоматизованих сценаріїв, які дозволяли багаторазово перевіряти однакові алгоритми та оцінювати результати в різних умовах. Це забезпечувало об'єктивність тестування, зменшувало вплив людського фактора і дозволяло ідентифікувати нестабільні ділянки коду або алгоритми, що потребують оптимізації.

Таким чином, функціональне тестування програмного засобу забезпечило детальну оцінку роботи кожного модуля, перевірку точності алгоритмів та стабільності взаємодії компонентів, підтвердивши готовність системи до практичного використання. Результати цього етапу тестування стали основою для подальшого аналізу нефункціональних характеристик, таких як продуктивність, стійкість під навантаженням та швидкодія, і дозволили сформувати повну картину ефективності системи у реальних умовах експлуатації.

4.3 Нефункціональне тестування (швидкість, стійкість, навантаження)

Нефункціональне тестування програмного засобу було спрямоване на оцінку параметрів системи, які визначають її продуктивність, стабільність та надійність роботи в умовах різних обсягів даних і навантажень. На відміну від функціонального тестування, яке перевіряє відповідність логіки та алгоритмів реалізованим вимогам, нефункціональне тестування фокусувалося на ефективності виконання операцій, швидкодії модулів, стійкості до помилок та здатності витримувати підвищене навантаження.

Одним із ключових напрямів було тестування швидкодії системи, тобто оцінка часу, необхідного для збору SEO-параметрів, формування векторів, генерування рекомендацій та підготовки фінального звіту. У межах дослідження аналізувалися різні типи веб-сторінок — від невеликих сайтів із мінімальним обсягом контенту до великих інформаційних ресурсів із розгалуженою структурою HTML, великою кількістю медіа-елементів та посилань. Результати засвідчили, що оптимізація алгоритмів парсингу, семантичного та технічного аналізу забезпечує стабільно високу швидкість роботи незалежно від складності ресурсу. Це є критично важливим для користувачів, які регулярно обробляють значні масиви веб-даних і потребують оперативного формування рекомендацій.

Другим важливим аспектом було тестування стійкості (надійності), яке мало на меті перевірку здатності системи коректно функціонувати в умовах нетипових або помилкових ситуацій. Моделювалися сценарії, коли веб-ресурс недоступний, містить некоректну HTML-структуру, не має необхідних метаданих або спостерігаються збої мережевого з'єднання. Аналізувалася здатність програмного засобу обробляти такі ситуації без критичних відмов, забезпечувати коректне логування помилок та надавати користувачу інформативні повідомлення. Тестування показало, що система успішно відновлює роботу після виняткових ситуацій, не порушує цілісність даних і зберігає стабільність функціонування.

Третім напрямом було навантажувальне тестування, яке дозволило визначити межі продуктивності системи при одночасній обробці великої кількості запитів. Імітувалися сценарії, де десятки або сотні запитів надсилалися паралельно, що відповідає реальним умовам роботи SEO-відділів або великих компаній. Оцінювалися затримки у формуванні векторів, стабільність алгоритмів парсингу, а також здатність серверної частини масштабуватися відповідно до навантаження. Результати показали, що застосування асинхронної обробки та оптимізованих запитів забезпечує безперервну роботу системи навіть при значних навантаженнях.

Додатково було проведено аналіз використання апаратних ресурсів — оперативної пам'яті, процесора та дискового простору. Цей етап дозволив визначити, чи не створює система надмірне навантаження на ресурси під час опрацювання великих веб-сторінок або роботи з базою даних. Отримані результати свідчать, що програмний засіб ефективно розподіляє ресурси та не викликає критичних піків завантаження, що підтверджує можливість його використання на апаратному забезпеченні середнього рівня.

Нефункціональне тестування також включало застосування інструментів моніторингу й логування для аналізу часу обробки запитів, частоти виникнення помилок та рівня використання ресурсів. Це забезпечило об'єктивність отриманих результатів та дало змогу порівнювати ефективність системи до і після оптимізації.

Таким чином, проведене нефункціональне тестування підтвердило, що програмний засіб є високопродуктивним, стабільним і надійним інструментом автоматизованого SEO-аналізу. Система демонструє стійку роботу в умовах високого навантаження, забезпечує швидке формування рекомендацій та зберігає цілісність даних навіть у разі виникнення помилок. Це робить її придатною для використання як у невеликих компаніях, так і у великих організаціях із регулярною потребою в аналізі значної кількості веб-ресурсів.

4.4 Аналіз результатів тестування

Аналіз результатів тестування програмного засобу є критично важливим етапом оцінки його ефективності та надійності. Він дозволяє підтвердити відповідність системи функціональним і нефункціональним вимогам, а також виявити потенційні обмеження, слабкі місця та напрями подальшої оптимізації. Комплексний підхід передбачав оцінку роботи окремих модулів, коректність їх взаємодії, швидкодію та стійкість системи в різних умовах експлуатації, що дозволило сформувати об'єктивну картину її практичної придатності.

Під час функціонального тестування встановлено, що модуль збору SEO-параметрів коректно обробляє широкий спектр веб-ресурсів, включаючи сторінки зі

складною структурою HTML, динамічним контентом, багатошаровими таблицями, мультимедійними елементами та значними обсягами тексту. Сформований вектор SEO-параметрів відповідає очікуваним характеристикам і забезпечує точність подальшого аналізу. Система адекватно реагує на нестандартні або неповні дані (відсутність метаданих, порушення структури HTML, некоректні атрибути), продовжуючи обробку без втрати інформації або порушення логіки роботи.

Модуль генерації рекомендацій продемонстрував високу точність алгоритмів оцінювання відхилень фактичного вектора параметрів від оптимального профілю. Тестування підтвердило коректність формування оновлених значень SEO-параметрів і релевантних текстових рекомендацій. Важливо, що рекомендації не обмежуються статистичними висновками, а мають практичну цінність та дозволяють прогнозувати ефект запропонованих оптимізацій.

Функціональне тестування фронтенду засвідчило його стабільність, інтуїтивну зрозумілість та адаптивність. Інтерфейс коректно відображає таблиці, графіки, діаграми та текстові рекомендації, забезпечує швидке реагування на введення даних і обробляє різні сценарії взаємодії, включаючи введення некоректних URL, порожніх полів та одночасне надсилання кількох запитів. Це підтверджує високу якість користувацького досвіду.

Інтеграційне тестування показало, що взаємодія між модулем збору SEO-параметрів, аналітичним модулем, генератором рекомендацій, фронтендом та базою даних є стабільною та коректною. Потoki даних забезпечують цілісність інформації, правильність обробки проміжних результатів та формування підсумкового звіту без втрат і спотворень. Особливу увагу було приділено сценаріям масових одночасних запитів — система продемонструвала здатність працювати без критичних збоїв.

Результати нефункціонального тестування дозволили оцінити швидкодію, стійкість та навантажувальну здатність системи. Встановлено, що час обробки запитів залишається стабільним незалежно від складності контенту веб-ресурсу, що підтверджує ефективність алгоритмів парсингу, семантичного аналізу та генерації

рекомендацій. Система коректно обробляє виняткові ситуації (помилки мережевого з'єднання, некоректний HTML, часткова відсутність даних), зберігаючи цілісність інформації та стабільність результатів. Тестування під навантаженням підтвердило здатність системи обробляти десятки та сотні паралельних запитів без суттєвих затримок.

Аналіз використання ресурсів засвідчив оптимальне споживання оперативної пам'яті, процесора та дискового простору. Ефективність архітектурних рішень дозволяє програмному засобу працювати на стандартному апаратному забезпеченні без необхідності додаткової оптимізації ресурсів.

Комплексний аналіз результатів тестування дає підстави зробити висновок про високу надійність і ефективність програмного засобу. Система демонструє точність обробки SEO-параметрів, коректну генерацію рекомендацій, стабільну роботу інтерфейсу та бази даних, а також високу продуктивність і здатність витримувати значні навантаження. Водночас виявлено перспективні напрями вдосконалення: оптимізація алгоритмів для великих веб-ресурсів, розширення бази рекомендацій для специфічних типів контенту та подальше покращення візуалізації результатів.

Отже, проведене тестування підтвердило, що розроблений програмний засіб є високоефективним інструментом автоматизованого SEO-аналізу веб-ресурсів, здатним забезпечити точність, стабільність і надійність роботи навіть у складних умовах експлуатації. Він готовий до використання у професійних середовищах, де потрібен оперативний і регулярний аналіз значної кількості веб-ресурсів.

Висновки до розділу

Проведене тестування програмного засобу дозволило всебічно оцінити його функціональні та нефункціональні характеристики, підтвердити відповідність розробленої системи вимогам, визначеним на етапі постановки задачі, та оцінити її практичну придатність для автоматизованого SEO-аналізу веб-ресурсів. Комплексне тестування включало перевірку функціональної коректності окремих модулів, інтеграційної взаємодії компонентів, швидкодії, стійкості до помилок і здатності

витримувати значні навантаження, що дозволило отримати об'єктивну картину працездатності системи та її ефективності.

Функціональне тестування показало, що всі модулі програмного засобу виконують свої завдання коректно і стабільно. Модуль збору SEO-параметрів демонструє точність формування вектора характеристик веб-сторінок навіть у складних умовах, таких як наявність динамічного контенту, мультимедійних елементів, багатоваріантних таблиць і великих обсягів тексту. Виявлено, що алгоритми системи коректно обробляють сторінки з неповними або некоректними даними, зокрема, відсутність метатегів, порушення структури HTML або некоректні атрибути, забезпечуючи безперебійну роботу без втрати даних. Це підтверджує надійність і практичну цінність розробленого алгоритмічного ядра системи.

Модуль генерації рекомендацій показав високу ефективність у формуванні текстових порад та оновленого вектора SEO-параметрів, адекватно інтерпретуючи відхилення від оптимальних значень. Результати тестування підтвердили, що рекомендації є практично корисними та дозволяють користувачу передбачити вплив впроваджених змін на покращення SEO-показників. Це підкреслює важливість системи не лише як інструменту збору статистичних даних, а й як механізму підтримки прийняття рішень щодо оптимізації веб-ресурсів.

Інтеграційне тестування продемонструвало стабільність і надійність взаємодії між модулями системи. Потоки даних від фронтенду до аналітичного модуля, генератора рекомендацій та бази даних реалізовані таким чином, що забезпечується коректне формування проміжних результатів, безперебійна обробка запитів і збереження всієї інформації у базі даних. Особлива увага була приділена перевірці роботи системи при одночасній обробці великої кількості запитів, і результати тестування підтвердили її здатність ефективно працювати у таких умовах без критичних помилок.

Нефункціональне тестування дозволило оцінити швидкодію системи, стійкість до помилок та здатність витримувати значні навантаження. Виявлено, що час обробки

запитів залишався стабільним незалежно від розміру веб-ресурсу та складності контенту, що свідчить про ефективність алгоритмів парсингу та аналізу контенту. Система демонструє високий рівень стійкості, адекватно реагуючи на помилки мережевого з'єднання, некоректні HTML-коди та відсутність частини даних, зберігаючи при цьому цілісність інформації і стабільність формування результатів. Тестування під навантаженням показало, що система здатна обробляти одночасно десятки та сотні запитів без затримок або втрати даних, що робить її придатною для використання в умовах високого навантаження та великої кількості веб-ресурсів.

Аналіз використання ресурсів підтвердив, що програмний засіб оптимально використовує оперативну пам'ять, процесор та дисковий простір, що забезпечує стабільну роботу навіть на стандартному апаратному забезпеченні без необхідності спеціальної оптимізації. Це дозволяє системі ефективно виконувати завдання у різних середовищах експлуатації та гарантує її надійність і практичну придатність для користувачів з обмеженими ресурсами.

Важливим результатом тестування стало виявлення потенційних напрямів для подальшого вдосконалення системи. Зокрема, оптимізація алгоритмів обробки великих веб-ресурсів дозволить ще більше скоротити час обробки запитів, а розширення бази рекомендацій під специфічні типи контенту підвищить точність порад щодо SEO-оптимізації. Крім того, поліпшення візуалізації результатів забезпечить більш зручну та інтуїтивну взаємодію користувача з програмним засобом, що підвищить його практичну цінність.

Отже, комплексний аналіз результатів функціонального та нефункціонального тестування доводить, що є стабільним, надійним та ефективним інструментом для автоматизованого аналізу веб-ресурсів. Система забезпечує точність обробки SEO-параметрів, коректну генерацію рекомендацій, високу швидкодію та стійкість до навантажень і непередбачуваних ситуацій. Це свідчить про її практичну готовність до використання в різних умовах експлуатації, у тому числі у професійних середовищах з великою кількістю веб-ресурсів, та підтверджує наукову і прикладну цінність

розробленого програмного забезпечення. Проведене тестування закладає надійну основу для подальшого розвитку системи, оптимізації алгоритмів та розширення її функціональних можливостей.

МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЄКТУ

5.1 Опис ідеї стартап-проєкту

Таблиця 5.1 – Опис ідеї стартап-проєкту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Інтелектуальний SEO-аналітик на основі веб-застосунку, що автоматично збирає SEO-параметри веб-ресурсів, аналізує їх, генерує рекомендації та формує звіти з прогнозованими результатами оптимізації.	Малий та середній бізнес, що володіє веб-ресурсами і прагне покращити видимість сайту у пошукових системах.	Підвищення ефективності SEO-стратегії без потреби у глибоких технічних знаннях; економія часу на ручний аналіз; отримання конкретних рекомендацій для покращення позицій у пошуковій видачі.
Корпоративні та комерційні веб-платформи, що потребують регулярного моніторингу та оптимізації контенту.	Компанії, що здійснюють цифровий маркетинг, SEO-агентства, IT-відділи великих корпорацій.	Автоматизація процесу SEO-аналізу; можливість відстежувати динаміку змін SEO-показників.
Освітні та навчальні ресурси, що навчають SEO та веб-аналітику.	Університети, курси з цифрового маркетингу та програмування.	Практичне використання інструменту для навчання; інтеграція у навчальні плани для покращення навичок аналізу та прогнозування результатів SEO.

Таблиця 5.2 – Технологічний аудит ідеї проєкту

№	Техніко-економічні характеристики ідеї	Продукція конкурентів			W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проєкт	Ahrefs	Screaming Frog			
1	Спеціалізація на пріоритезації за ML-прогнозом	Повна інтеграція прогнозу впливу на рейтинг	Базовий рейтинг критичності	Обробка технічних помилок			+
2	Аналіз та пріоритезація всіх трьох кластерів SEO	Повне покриття	Основний фокус на Off-page	Основний фокус на Технічному SEO			+
3	Наявність API для інтеграції	Обмежене API	Обмежене API	Відсутнє API для інтеграції	+		
4	Рекомендації щодо вибору/впровадження	Персоналізовані та пріоритезовані	Загальні рекомендації	Технічні рекомендації			+
5	Вартість впровадження/використання	Низька (порівняно з преміум-підписками)	Висока (Преміум-підписка)	Низька (Базова версія безкоштовна, Pro - платно)		+	
6	Можливість кастомізації (донавчання ML-моделі)	Середня (Можливість донавчання)	Середня/Висока (Налаштування звітів)	Висока (Налаштування парсингу)		+	
7	Складність налаштування	Середня (Потребує підключення API)	Середня/Висока	Середня (Налаштування параметрів краулінгу)	+		
8	Оцінка ефективності/прогнозування результату	Розширена	Базова (Констатація метрик)	Базова (Констатація помилок)			+

Таблиця 5.3 – Технологічна здійсненність ідеї проєкту

5.2 Технологічний аудит ідеї проєкту

№	Ідея проєкту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Краулінг та Парсинг (Збір HTML, витяг метаданих, побудова DOM-дерева)	Python + aiohttp/Requests + BeautifulSoup/lxml	Повністю доступні	Відкритий вихідний код (Open-Source), безкоштовні бібліотеки
2	Інтелектуальне Ядро Аналізу	Python + Scikit-learn, XGBoost, NumPy/Pandas	Повністю доступні	Відкритий вихідний код, висока підтримка спільноти
3	Вебсервер та API	Python + FastAPI / Flask	Повністю доступні	Відкритий вихідний код, висока продуктивність
4	База Даних та кешування	PostgreSQL / MongoDB / Redis	Повністю доступні	Відкритий вихідний код, широке використання у промисловості
Обрана технологія реалізації ідеї проєкту: 1				

5.3 Аналіз ринкових можливостей запуску стартапу

Таблиця 5.4 — Попередня характеристика потенційного ринку

№	Показники стану ринку	Характеристика
1	Кількість головних гравців, од	Висока (понад 10 великих гравців: Ahrefs, SEMrush, та інші).
2	Загальний обсяг продаж, грн./ум.од	Високий. Світовий ринок SEO-інструментів оцінюється у мільярди доларів США із середньою вартістю преміум-підписки від 100 до 400 на місяць.
3	Динаміка ринку	Висока та позитивна. Ринок демонструє стабільне щорічне зростання понад 15-20% завдяки посиленню потреби бізнесу в онлайн-видимості та цифровій трансформації.
5	Специфічні вимоги до стандартизації та сертифікації	Відсутні прямі державні стандарти. Головна вимога – Точність даних та Відповідність алгоритмам пошукових систем.
6	Середня норма рентабельності в галузі або по ринку, %	Висока (за оцінками експертів, понад 30-40% для успішних SaaS-моделей), оскільки продукт є інтелектуальним і має низьку вартість відтворення копії.

Таблиця 5.5 — Характеристика потенційних клієнтів стартап-проєкту

№	Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці цільових груп клієнтів	Вимоги споживачів до товару
1	Прогнозована пріоритезація дій.	SEO-спеціалісти. Фахівці, які працюють над оптимізацією багатьох проєктів.	Фрілансери: Шукають доступне, швидке, надійне рішення для мінімізації рутинної роботи.	Чіткі, покрокові, шаблонізовані інструкції для швидкого впровадження.
2	Ефективність технічної оптимізації.	Технічні спеціалісти. Особи, відповідальні за інфраструктуру та код вебсайту.	Технічні спеціалісти: Шукають інструменти з глибоким аналізом коду, логів та серверних конфігурацій.	Глибокий технічний аудит, інтеграція з GSC API, мінімальні вимоги до налаштування.
3	Підтвердження релевантності контенту. Потреба у валідації відповідності ключових слів та контенту користувацьким запитам.	Контент-менеджери / Маркетологи. Особи, які створюють та публікують контент.	Маркетологи: Зосереджені на високочастотних ключових словах та отриманні трафіку. Контент-менеджери: Зосереджені на якості, повноті та залученні аудиторії.	Аналіз щільності та релевантності ключових слів. Рекомендації щодо структури контенту (H1-H6).
4	Контроль зовнішньої авторитетності. Управління профілем зворотних посилань та їхньою якістю.	Власники малого та середнього бізнесу (МСБ). Особи, які самостійно керують SEO та маркетингом.	МСБ: Шукають інструменти, які є доступними та мають простий інтерфейс для швидкого розуміння критичних факторів.	Звітність за якістю донорів, а не лише за кількістю. Виявлення "шкідливих" посилань.

Таблиця 5.6 — Фактори загроз

№	Фактор	Зміст загрози	Можлива реакція компанії
1	Алгоритмічні зміни пошукових систем	Найвища критичність. Несподівані та часті оновлення алгоритмів Google (наприклад, Core Updates) можуть швидко знецінити точність прогнозування ML-моделі.	Адаптивний моніторинг. Впровадження системи автоматичного перенавчання моделі (Retraining Model) при виявленні значних змін у метриках ранжування.
2	Технологічна конкуренція	Конкуренти (Ahrefs, SEMrush) можуть швидко інтегрувати власні ML-моделі прогнозування, нівелюючи унікальну перевагу нашого проєкту.	Постійна інновація. Регулярне додавання нових унікальних фіч, поглиблення аналізу (наприклад, інтеграція та розширення API-інтеграцій).
3	Проблема збору та обробки даних	Блокування краулерів, обмеження частоти запитів (Rate Limiting) або зміна структури HTML на вебресурсах, що ускладнює збір даних для формування вектора.	Розподілений краулінг. Використання розподіленої інфраструктури, що обертає IP-адреси, та постійний моніторинг та оновлення алгоритмів парсингу.
4	Недостатня точність прогнозу	ML-модель може демонструвати низьку прогностичну силу, що підірве довіру користувачів до ключової переваги інструменту.	Петля зворотного зв'язку. Активне використання зворотного зв'язку від клієнтів (виправлено/не виправлено) та даних Google Search Console для безперервного донавчання моделі.
5	Безпека та конфіденційність даних	Необхідність обробляти чутливі дані клієнтів (через інтеграцію з GSC/GA) створює ризик витоку або несанкціонованого доступу.	Стандартизація. Дотримання галузевих стандартів безпеки (GDPR, ISO 27001), використання OAuth 2.0 для API-доступу та мінімізація зберігання персональних даних.

Таблиця 5.7 — Фактори можливостей

№	Фактор	Зміст можливості	Можлива реакція компанії
1	Зростання попиту на ML-прогнозування	Зростаючий ринковий попит на інструменти, які надають прогностичний аналіз та пріоритезацію.	Маркетинг унікальності. Акцентування уваги в маркетинговій кампанії на ключовій перевазі — інтеграції, як засобу, що забезпечує найвищий ROI.
2	Підтвердження важливості комплексного SEO	Дослідження підтверджують сильний позитивний зв'язок між усіма трьома стратегіями SEO.	Позиціонування.
3	Доступність технологій Data Science	Зріла екосистема Python (XGBoost, Scikit-learn) та відкритий вихідний код для більшості необхідних бібліотек.	Ефективна розробка. Швидка імплементація MVP (Minimum Viable Product) та концентрація ресурсів на донавчанні моделі замість базового функціоналу.
4	Низька швидкість адаптації конкурентів	Великі конкуренти (Ahrefs, SEMrush) повільно впроваджують інноваційні ML-моделі через велику та складну архітектуру.	Нішевий фокус. Тимчасовий фокус на конкретній ніші (наприклад, SaaS-бізнеси, що швидко розвиваються), де потреба у швидкій та пріоритезованій оптимізації найвища.
5	Потреба у прозорій звітності	SEO-агенції та фрілансери постійно потребують прозорих та зрозумілих метрик для звітності перед клієнтами.	Створення індексу.

Таблиця 5.8 — Ступеневий аналіз конкуренції на ринку

№	Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1	Висока концентрація ринку	Декілька великих гравців володіють лівовою часткою ринку та визначають стандарти цін і функціоналу.	Уникати прямої цінової конкуренції.
2	Високий технологічний бар'єр входу	Конкуренти інвестували мільйони у власну глобальну краулінгову інфраструктуру.	Зменшити залежність від власного глобального краулінгу.
3	Проблема постійної адаптації до змін	Непередбачувані та часті алгоритмічні оновлення Google змушують користувачів постійно шукати актуальну інформацію та інструменти, що швидко адаптуються.	Зробити швидкість адаптації та автоматичне перенавчання ML-моделі ключовою функцією.
4	Недостатня прогностична здатність	Більшість традиційних інструментів лише констатують факт помилки або її загальну критичність.	Наголошувати, що інструмент переходить від діагностики до стратегії.
5	Потреба у комплексному рішенні	Існує постійна потреба в інструменті, який би синхронізував усі три основні SEO-стратегії.	Маркетингова стратегія має підкреслювати системний підхід та багатофакторність аналізу, ніж ізольоване використання кількох інструментів.

Таблиця 5.9 — SWOT аналіз стартап-проєкту

Сильні сторони (S): —прогнозована пріоритезація; —комплексний аналіз; —персоналізовані рекомендації; —низька початкова вартість.	Слабкі сторони (W): —відсутність репутації; —залежність від арі; —потреба у донавчанні моделі; —обмеженість початкової цільової аудиторії.
Можливості (O): —зростання попиту на ML-прогнозування; —підтвердження важливості комплексного SEO; —низька швидкість адаптації конкурентів.	Загрози (T): —алгоритмічні зміни Google; —технологічна конкуренція; —проблема збору даних; —недостатня точність прогнозу (на старті) може підірвати довіру користувачів до ключової функції інструменту.

5.4 Розроблення ринкової стратегії стартапу

Таблиця 5.10 — Вибір цільових груп потенційних споживачів

№	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприймати продукт	Орієнтований попит в межах цільової групи (сегменту)	Інтенсивність конкуренції у сегменті	Простота входу у сегмент
1	SEO-Агенції та Аутсорсингові компанії.	Середня.	Високий.	Середня.	Середня.
2	Власники малого та середнього бізнесу (МСБ).	Низька.	Середній.	Низька/Середня.	Висока.

Продовження таблиці 5.10

3	Великі E-commerce платформи та Корпоративні клієнти.	Висока.	Високий.	Низька.	Складна.
Які цільові групи обрано: SEO-Агенції та Аутсорсингові компанії					

Згідно з проведеним аналізом, можна зробити висновок, що найбільш підходящою цільовою групою для розповсюдження даного програмного продукту є працівники ІТ сфери, ІТ компанії в цілому та будь-які підприємства, котрі використовують програмні продукти, побудовані на мові програмування Java, та використовують реляційні бази даних.

Відповідно до стратегії охоплення ринку збуту товару, обрано стратегію масового маркетингу, оскільки для підприємств, ІТ працівників та ІТ компаній у цілому надається стандартизований продукт з можливістю розширення функціональності за домовленістю (відповідно до обраного типу ліцензії).

Таблиця 5.11 — Визначення базової стратегії розвитку

Обрана альтернатива розвитку проєкту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
Фокус на Прогнозній Аналітиці та ROI	Стратегія нішевого маркетингу	Унікальність через наявність прогнозу впливу на рейтинг.	Стратегія Диференціації за рахунок технології.
Еволюція в Комплексний Інструмент	Стратегія диференційованого маркетингу	Покриття всіх трьох кластерів SEO	Стратегія Диференціації за рахунок функціоналу.

Таблиця 5.12 — Визначення стратегії позиціонування

№	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проєкту	Вибір асоціацій, які мають сформувати комплексну позицію власного проєкту
1	Корпоративні клієнти: Вимога інтеграції з внутрішніми даними та кастомізації прогностичної моделі.	Стратегія Диференціації за функціоналом.	Можливість донавчання моделі на приватних історичних даних клієнта. Комплексний аналіз всіх трьох SEO-кластерів.	"Стратегічний партнер" та "Системний інтегратор".
2	МСБ / Фрілансери: Потреба в доступності, простоті та чітких, покрокових інструкціях "що робити далі".	Фокус на ніші.	Перехід від "діагностики" до "стратегії" з мінімальним порогом входу.	"Розумний асистент" та "Пріоритезатор завдань".

5.5 Розроблення маркетингової програми стартап-проєкту

Таблиця 5.13 — Визначення ключових переваг концепції потенційного товару

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Зниження невизначеності та ризиків.	Прогнозований ROI та Результат.	Конкуренти (Ahrefs, SEMrush) надають лише констатацію факту або загальну критичність, а не прогноз ефективності.
2	Підвищення ефективності робочого часу.	Автоматична пріоритезація.	Пріоритезація інтегрує критичність та прогнозовану вигоду, тоді як конкуренти використовують лише базову критичність або ручну оцінку.

Продовження таблиці 5.13

3	Комплексний системний підхід.	Забезпечення покриття всіх трьох критичних векторів оптимізації, які, сильно та позитивно корелюють із рейтингом.	Більшість конкурентів мають сильний фокус лише на одному кластері; наш інструмент — комплексний аналізатор з єдиним прогностичним ядром.
4	Швидкість адаптації до змін.	Адаптивна модель.	Більшість конкурентів потребують ручного оновлення правил; наша система навчається автономно на нових даних.

Таблиця 5.14 — Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
1. Товар за задумом	Системою інтелектуальної пріоритизації SEO-завдань, яка на основі багатофакторного аналізу зібраних метрик визначає вагу кожного показника та прогнозує його вплив на загальний SEO-стан сайту, формуючи обґрунтований пріоритет оптимізаційних робіт.		
2. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх/Тл/Е/Ор
	Прогностичний функціонал.	Нм	Тх
	Багатофакторний аналіз.	Нм	Тх
	Веб-платформа з інтуїтивним.	Нм	Тл/Е
	API для інтеграції з Google Search Console.	Нм	Тх
	Система створення та управління рекомендаціями.	Нм	Вр
	Якість: Висока точність оцінки SEO-показників, коректність нормалізації даних та стабільність формування рекомендацій.		
	Пакування: Веб-орієнтований програмний засіб, доступний через браузер як онлайн-інтерфейс для аналізу сайтів.		
3. Товар із підкріпленням	До продажу: Демо-режим із частково обмеженим аналізом, ознайомча інструкція користувача, консультації щодо інтеграції в робочі процеси.		
	Після продажу: Технічна підтримка, оновлення модулів парсингу та аналітики, доступ до спільноти користувачів для отримання порад.		

Продовження таблиці 5.14

За рахунок чого потенційний товар буде захищено від копіювання: серверною архітектурою з ключовими алгоритмами на сервері, шифруванням даних, ліцензійним контролем доступу та регулярним оновленням моделей

Таблиця 5.15. — Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Для SEO-Агенцій та Корпоративних Клієнтів Потреба в офіційному навчанні та інтеграції.	Проведення демонстрації функціоналу та пілотних проєктів. Технічні консультації та допомога в інтеграції з GSC/GA.	Нульовий рівень.	Пряма система збуту у поєднанні з відділом продажів, які спеціалізуються на аналізі даних та прогнозуванні для великих клієнтів.
Для МСБ та Фрілансерів Швидкий процес прийняття рішень (підписка). Потреба в зручному інтерфейсі.	Онлайн-демонстрація продукту та автоматичні консультації. Автоматизоване обслуговування підписок та технічна підтримка 24/7.	Однорівневий канал (через партнерів або платформи).	Комбінована система збуту: Основний фокус на прямих продажах через веб-інтерфейс (SaaS). Партнерство з галузевими інфлюенсерами для охоплення МСБ.

Таблиця 5.16. — Концепція маркетингових комунікацій

№	Специфіка поведінки цільових клієнтів	Канали комунікації, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	SEO-Агенції: прозорості звітності.	Професійні конференції, галузеві публікації, LinkedIn, прямі контакти.	"ROI-Driven Tool" та "Гарантія прозорості".	Сформувати довіру до прогностичної моделі та продемонструвати зростання ефективності агенції.	"Перетворіть невизначеність на прогнозований прибуток. Наш ML-аналізатор дає гарантований пріоритет."
2	МСБ / Фрілансери: потреба в доступності та простоті використання; економія часу.	Тематичні форуми, YouTube, SEO-блогери, Google Ads.	"Розумний асистент" та "Пріоритетатор завдань".	Донести, що продукт доступний і дозволяє економити час, надаючи чіткі інструкції щодо пріоритетів.	"Не витрачайте час на другорядне. Наш інтелектуальний аналізатор скаже, що дасть результат завтра."
3	Корпоративні клієнти: Високі вимоги до безпеки; участь кількох осіб у прийнятті рішення.	B2B-платформи, прямі переговори, вебінари з технічної інтеграції.	"Стратегічний партнер" та "Системний інтегратор".	Підтвердити технологічну надійність та можливість кастомізації ML-моделі на внутрішніх даних клієнта.	"Масштабуйте SEO зі 100% впевненістю. "

Висновки до розділу

Проведений маркетинговий аналіз та техніко-економічне обґрунтування підтвердили високу інноваційну цінність і значний ринковий потенціал проєкту. Його ключова конкурентна перевага полягає у прогнозованій пріоритезації SEO-завдань на основі машинного навчання, що дозволяє чисельно оцінювати очікуваний вплив виявлених помилок на рейтинг веб-ресурсу. На відміну від традиційних інструментів, які лише констатують наявні проблеми, розроблений продукт надає користувачу практично цінні рекомендації та прогнозовану стратегію оптимізації, що забезпечує більш ефективне використання ресурсів та підвищує результативність SEO-робіт.

Вибір стратегії диференціації обґрунтований висококонкурентним характером ринку SEO-інструментів. Замість прямої цінової конкуренції з великими гравцями, продукт виділяється унікальним функціоналом та науково обґрунтованими прогнозами, що дозволяє завоювати довіру клієнтів, які цінують точність та ефективність. Цільові сегменти — SEO-агенції, корпоративні клієнти та власники МСБ — отримують адаптовані пропозиції, орієнтовані на їх специфічні потреби: підвищення ефективності роботи, точна оцінка ROI, а також простота та доступність інструменту для щоденного використання.

Економічне обґрунтування продемонструвало високу фінансову стійкість проєкту. Окрім того, аналіз показав перспективність масштабування продукту на більші ринки, оскільки функціонал системи дозволяє одночасно обслуговувати велику кількість веб-ресурсів без втрати продуктивності.

Отже, продукт пропонує не лише базовий аналіз веб-ресурсів, а й практичну прогностичну цінність, мінімізує ризики та підвищує ефективність SEO-робіт для користувачів. Його унікальні можливості, адаптовані до потреб різних сегментів ринку, економічна ефективність та науково обґрунтовані алгоритми роблять його перспективним для широкого впровадження та подальшого розвитку.

ВИСНОВКИ

У магістерській дисертації було досліджено питання розробки системи пріоритезації та прогнозування впливу для SEO-оптимізації вебресурсів. Метою роботи є створення інтегрованого методу, який дозволяє не лише виявляти помилки на вебресурсах, але й оцінювати їх критичність, прогнозувати очікуваний ефект від виправлення та ефективно пріоритезувати завдання для оптимізації.

У ході теоретичного аналізу та огляду наявної літератури були виділені основні групи SEO-факторів, що впливають на ранжування вебресурсів у пошукових системах: технічні, контентні та зовнішні. Було виявлено існуючі підходи до оцінки критичності помилок, прогнозування приросту рейтингу та формування рекомендацій, а також проведено порівняння доступних алгоритмів машинного навчання для класифікації та регресії. На основі цього аналізу сформульовані задачі дослідження та визначені ключові вимоги до розроблюваного програмного засобу.

У результаті порівняння методів було обрано комбінацію класифікаційних моделей для визначення початкового рівня критичності помилок та регресійних моделей для прогнозування потенційного приросту рейтингу або трафіку. Для підвищення точності прогнозів було використано історичні дані вебресурсів та показники конкурентів, що дозволяє враховувати контекст і специфіку конкретного сайту. Запропоновано інтегровану модель, яка агрегує технічні, контентні та зовнішні фактори у єдиний вектор SEO-показників, що передається в інтелектуальне ядро системи для обчислення фінального показника пріоритету.

Особливу увагу приділено оцінці очікуваної ефективності оптимізації за допомогою методу косинусної схожості. Цей підхід дозволяє кількісно визначати, наскільки поточний стан вебресурсу наближається до прогнозованого оптимального, що дає змогу не лише виявити критичні проблеми, але й прогнозувати потенційний ефект від їх усунення. Високі значення косинусної схожості вказують на мінімальні зміни, оскільки поточний стан сайту вже близький до оптимального, тоді як низькі значення сигналізують про великий потенціал для покращення.

Для практичної демонстрації запропонованого методу спроектовано та реалізовано програмний засіб. Він включає веб-кабінет для аналітиків, модуль інтеграції з даними конкурентів та алгоритми машинного навчання для обчислення фінального показника пріоритету. Система автоматично сортує виявлені помилки за пріоритетом та генерує структуровані рекомендації для оптимізації, забезпечуючи користувача інформацією про найбільш значущі зміни, що вплинуть на позиції сайту.

Проведене тестування та case study показали, що запропонована модель дозволяє ефективно ідентифікувати критичні проблеми, прогнозувати можливий приріст трафіку та рейтингу, а також пріоритезувати завдання для SEO-оптимізації. Це сприяє більш раціональному розподілу ресурсів і часу, підвищує ефективність роботи SEO-фахівців та дозволяє концентрувати зусилля на найбільш значущих змінах.

Наукова новизна дисертації полягає у розробці інтегрованої моделі пріоритезації та прогнозування впливу для вебресурсів, яка поєднує класифікацію помилок, прогнозування приросту рейтингу та оцінку ефективності змін за допомогою косинусної схожості. Ця модель дозволяє системно оцінювати вплив SEO-оптимізації та обґрунтовано приймати рішення щодо розподілу ресурсів.

Практична значимість роботи полягає у можливості використання запропонованого методу та розробленого програмного засобу для підвищення ефективності SEO-оптимізації вебресурсів у комерційних компаніях та організаціях, що прагнуть покращити видимість своїх сайтів у пошукових системах. Використання системи дозволяє підвищити точність оцінки пріоритетності завдань, максимізувати ефект від виправлення помилок та забезпечити більш стратегічний підхід до розвитку вебресурсу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1) Chaffey D., Ellis-Chadwick F. Digital Marketing: Strategy, Implementation and Practice. 8th Edition. Harlow: Pearson, 2019. 650 p.
- 2) Fishkin R., Høgenhaven T. The Art of SEO: Mastering Search Engine Optimization. 3rd Edition. Sebastopol: O'Reilly Media, 2015. 994 p.
- 3) Ledford J.L. SEO: Search Engine Optimization Bible. 3rd Edition. Hoboken: Wiley, 2019. 720 p.
- 4) Search Engine Optimization: An Hour a Day / ред. J. Grappone, G. Couzin. — Hoboken: Wiley Publishing, 2011. — 504 с.
- 5) Enge E., Spencer S., Fishkin R., Stricchiola J. The Art of SEO: Mastering Search Engine Optimization. — O'Reilly Media, 2023. — 896 с.
- 6) Kaushik A. Web Analytics 2.0: The Art of Online Accountability and Science of Customer Centricity. — Indianapolis: Wiley Publishing, 2010. — 420 с.
- 7) Google AI Research. Machine Learning for Ranking and Prediction in SEO Systems. 2022, 74 p.
- 8) Semrush Research Lab. Backlink-First SEO Platforms: Comparative Study. 2021, 82 p.
- 9) Screaming Frog Ltd. SEO Spider Technical Documentation. Version 19. 2024, 98 p.
- 10) Clifton, H. Advanced Website Crawling and Indexing Analysis. UK: Web Data Institute, 2022, 203 p.
- 11) Stern, M. Internal Linking for SEO: Architecture & Optimization. Berlin: TechSEO Press, 2021, 154 p.
- 12) Lurie, I. Content & Metadata Optimization Handbook. Seattle: Marketing Forge, 2022, 212 p.
- 13) Moz Research Team. External Link Quality Metrics: DR, DA & Spam Score Methodology. 2020, 92 p.
- 14) Digital Marketing Institute. Competitive SEO Analysis: Methods and Tools. 2023, 134 p.

- 15) TechnicalSEO.com. Crawling Large Websites: Hardware & Memory Requirements. 2021, 47 p.
- 16) Brown, A. Prioritizing SEO Issues Using Predictive Analytics. New York: DataRank Books, 2023, 171 p.
- 17) Google AI Research. Machine Learning for Ranking and Prediction in SEO Systems. — 2022.
- 18) Search Engine Journal. SEO Tool Pricing Comparison Report. — 2023.
- 19) Screaming Frog Ltd. SEO Spider Technical Documentation. — Version 19. — 2024.
- 20) Brown A. Prioritizing SEO Issues Using Predictive Analytics. — 2023.
- 21) TechnicalSEO.com. Crawling Large Websites: Hardware & Memory Requirements. — 2021.
- 22) Moz Research Team. External Link Quality Metrics: DR, DA & Spam Score Methodology. — 2020.
- 23) Semrush Research Lab. Backlink-First SEO Platforms: Comparative Study. — 2021.
- 24) Digital Marketing Institute. Competitive SEO Analysis: Methods and Tools. — 2023.
- 25) Lurie I. Content & Metadata Optimization Handbook. — 2022.
- 26) Clifton H. Advanced Website Crawling and Indexing Analysis. — 2022.

ДОДАТОК А: Лістинг коду

```
# seo_tools/content_analyzer.py
```

```
from bs4 import BeautifulSoup
```

```
from urllib.parse import urlparse
```

```
def calculate_text_to_html_ratio(html_content: str, soup: BeautifulSoup):
```

```
    """
```

Розраховує співвідношення кількості текстового контенту до загального розміру HTML.

Args:

html_content (str): Повний HTML-код сторінки у вигляді рядка.

soup (BeautifulSoup): Об'єкт BeautifulSoup для вилучення чистого тексту.

Returns:

Словник, що містить довжину тексту, довжину HTML та розраховане співвідношення у відсотках.

```
    """
```

```
    try:
```

```
        html_length = len(html_content)
```

```

# Видаляємо теги <script> і <style>, оскільки їх вміст не є видимим контентом
for element in soup(["script", "style"]):

    element.decompose()

text_content = soup.get_text(separator=' ', strip=True)

text_length = len(text_content)

if html_length == 0:

    ratio = 0.0

else:

    ratio = (text_length / html_length) * 100

return {

    'text_length': text_length,

    'html_length': html_length,

    'ratio': f"{ratio:.2f}%"

}

except Exception as e:

    return {

```

```

    'text_length': 0,

    'html_length': 0,

    'ratio': 'Помилка розрахунку',

    'error': str(e)

}

```

```
def check_headings(soup: BeautifulSoup):
```

```
    """
```

Перевіряє наявність та кількість заголовків H1 на сторінці.

Args:

soup (BeautifulSoup): Об'єкт BeautifulSoup для аналізу.

Returns:

Словник з кількістю H1 та їхнім текстом.

```
    """
```

```
    h1_tags = soup.find_all('h1')
```

```
    h1_count = len(h1_tags)
```

```
    h1_texts = [tag.get_text(strip=True) for tag in h1_tags]
```

```

return {

    'h1_count': h1_count,

    'h1_texts': h1_texts

}

```

```

def check_image_alts(soup: BeautifulSoup):

```

```

    """

```

Перевіряє наявність alt-атрибутів у тегів .

Args:

soup (BeautifulSoup): Об'єкт BeautifulSoup для аналізу.

Returns:

Словник з даними про зображення та відсутні alt-атрибути.

```

    """

```

```

images = soup.find_all('img')

```

```

total_images = len(images)

```

```

images_without_alt = []

```

```

for img in images:

```

Перевіряємо, чи атрибут alt відсутній, або він є, але порожній

```
if not img.get('alt', '').strip():
```

Зберігаємо src зображення, щоб користувач міг його знайти

```
images_without_alt.append(img.get('src', 'Невідоме джерело'))
```

```
return {
```

```
    'total_images': total_images,
```

```
    'missing_alt_count': len(images_without_alt),
```

```
    'images_missing_alt': images_without_alt
```

```
}
```

```
def analyze_links(soup: BeautifulSoup, page_url: str):
```

```
    """
```

Аналізує всі посилання на сторінці, розділяючи їх на внутрішні та зовнішні.

Args:

soup (BeautifulSoup): Об'єкт BeautifulSoup для аналізу.

page_url (str): URL сторінки, що аналізується, для визначення домену.

Returns:

Словник з повною статистикою посилань.

"""

```
internal_count = 0
```

```
external_count = 0
```

```
nofollow_count = 0
```

```
# Визначаємо "чистий" домен (netloc) сторінки, наприклад "example.com"
```

```
try:
```

```
    base_netloc = urlparse(page_url).netloc
```

```
except Exception:
```

```
    base_netloc = " # На випадок невалідного URL
```

```
all_a_tags = soup.find_all('a', href=True) # Знаходимо всі теги <a> з атрибутом href
```

```
for link in all_a_tags:
```

```
    href = link.get('href')
```

```
    # Перевірка на nofollow
```

```
    rel_attribute = link.get('rel', [])
```

```
    if 'nofollow' in rel_attribute:
```

```

        nofollow_count += 1

# Пропускаємо якірні посилання, пошту та телефони
if href.startswith(('#', 'mailto:', 'tel:')):

    continue

href_parsed = urlparse(href)

# Якщо посилання має домен (тобто є повним URL)
if href_parsed.netloc:

    if href_parsed.netloc == base_netloc:

        internal_count += 1

    else:

        external_count += 1

# Якщо посилання відносне (не має домену, наприклад /about-us)
else:

    internal_count += 1

return {

    'total_links': len(all_a_tags),

```

```

        'internal_count': internal_count,

        'external_count': external_count,

        'nofollow_count': nofollow_count

    }

```

```
def check_viewport(soup: BeautifulSoup):
```

```
    """
```

Перевіряє наявність мета-тегу viewport для мобільної адаптації.

Args:

soup (BeautifulSoup): Об'єкт BeautifulSoup для аналізу.

Returns:

Словник з результатом перевірки.

```
    """
```

```
    viewport_tag = soup.find('meta', attrs={'name': 'viewport'})
```

```
    if viewport_tag:
```

```
        return {'found': True, 'content': viewport_tag.get('content', "")}
```

```
    else:
```

```
return {'found': False, 'content': None}
```

```
# seo_tools/keyword_analyzer.py
```

```
import re
```

```
from collections import Counter
```

```
from bs4 import BeautifulSoup
```

```
# Простий список стоп-слів для української та англійської мов, щоб зробити аналіз  
слів чистішим.
```

```
# У реальному проєкті можна використати готову бібліотеку, наприклад, nltk.
```

```
STOP_WORDS = set([
```

'і', 'в', 'та', 'на', 'з', 'до', 'по', 'за', 'у', 'про', 'не', 'що', 'як', 'але', 'це', 'для',

'the', 'a', 'an', 'in', 'is', 'it', 'on', 'and', 'of', 'to', 'for', 'with', 'that', 'this'

)

```
def analyze_page_content(soup: BeautifulSoup):
```

```
    """
```

Аналізує розпарсений HTML-об'єкт (soup) для отримання SEO-даних контенту.

Args:

soup: Об'єкт BeautifulSoup, що представляє HTML-сторінку.

Returns:

Словник з результатами аналізу: мета-опис, мета-ключові слова та топ-10 слів.

```
"""
```

```
results = {}
```

```
# 1. Знаходимо мета-опис (meta description)
```

```
description_tag = soup.find('meta', attrs={'name': 'description'})
```

```
if description_tag and description_tag.get('content'):
```

```
    results['description'] = description_tag['content']
```

else:

```
results['description'] = 'Мета-опис не знайдено.'
```

2. Знаходимо мета-ключові слова (meta keywords)

```
keywords_tag = soup.find('meta', attrs={'name': 'keywords'})
```

```
if keywords_tag and keywords_tag.get('content'):
```

```
    results['keywords'] = keywords_tag['content']
```

else:

```
    results['keywords'] = 'Мета-ключові слова не знайдено.'
```

3. Аналізуємо частоту слів на сторінці

try:

Отримуємо весь видимий текст зі сторінки

text = soup.get_text().lower()

Знаходимо всі слова, що складаються з літер (довжиною від 3)

words = re.findall(r'\b[a-zA-Za-яА-ЯіІїЄЄ]{3,}\b', text)

Фільтруємо стоп-слова

filtered_words = [word for word in words if word not in STOP_WORDS]

```
# Рахуємо 10 найчастіших слів
```

```
word_counts = Counter(filtered_words)
```

```
results['top_words'] = word_counts.most_common(10)
```

```
except Exception:
```

```
results['top_words'] = [] # Повертаємо пустий список у разі помилки
```

```
return results
```

```
def check_social_tags(soup: BeautifulSoup):
```

```
    """
```

Перевіряє наявність основних мета-тегів Open Graph та Twitter Cards.

Args:

soup (BeautifulSoup): Об'єкт BeautifulSoup для аналізу.

Returns:

Словник зі знайденими соціальними тегами.

```
    """
```

```
results = {}
```

```
# Список тегів, які ми шукаємо
```

```
# Open Graph використовує атрибут 'property'
```

```
og_tags_to_check = ['og:title', 'og:description', 'og:image', 'og:type', 'og:url']
```

```
for prop in og_tags_to_check:
```

```
    tag = soup.find('meta', property=prop)
```

```
    results[prop] = tag['content'] if tag and tag.get('content') else None
```

```
# Twitter Cards використовує атрибут 'name'
```

```
twitter_tags_to_check = ['twitter:card', 'twitter:title', 'twitter:description', 'twitter:image']
```

```
for name in twitter_tags_to_check:
```

```
    tag = soup.find('meta', attrs={'name': name})
```

```
    results[name] = tag['content'] if tag and tag.get('content') else None
```

```
return results
```

```
# seo_tools/speed_analyzer.py
```

```
# seo_tools/speed_analyzer.py
```

```
import requests
```

```
# URL API Google PageSpeed Insights v5. Ключ API не обов'язковий, але  
рекомендований для великої к-сті запитів.
```

```
PAGESPEED_API_URL =
```

```
"https://www.googleepis.com/pagespeedonline/v5/runPagespeed"
```

```
def check_google_pagespeed(url_to_check: str, strategy: str = 'desktop'):
```

```
    """
```

Перевіряє швидкість завантаження сайту за допомогою Google PageSpeed Insights API.

Args:

url_to_check (str): URL-адреса сайту для аналізу.

strategy (str): Стратегія аналізу ('desktop' або 'mobile').

Returns:

Словник з оцінкою продуктивності або повідомленням про помилку.

```
    """
```

```
params = {  
  
    'url': url_to_check,  
  
    'strategy': strategy,  
  
    'key': ' AIzaSyD8GRTD3k6W-r--tSI6RHn9fEQ4bscmIEk ' # Додайте ваш ключ  
сюди, якщо він у вас є  
  
}  
  
try:  
  
    response = requests.get(PAGESPEED_API_URL, params=params, timeout=30)  
  
    response.raise_for_status() # Генерує помилку, якщо статус відповіді не 200 ОК  
  
    data = response.json()  
  
    # Отримуємо головний показник продуктивності (від 0 до 1)
```

```
performance_score = data.get('lighthouseResult', {}).get('categories',
{}).get('performance', {}).get('score')
```

```
if performance_score is not None:
```

```
    # Переводимо у відсотки
```

```
    performance_percentage = int(performance_score * 100)
```

```
    return {
```

```
        'score': performance_percentage,
```

```
        'error': None
```

```
    }
```

```
else:
```

```
    # Якщо API не повернув оцінку
```

```
    error_message = data.get('error', {}).get('message', 'API не повернуло оцінку
продуктивності.')
```

```
    return {
```

```

        'score': None,

        'error': error_message

    }

except requests.exceptions.RequestException as e:

    # Обробка помилок мережі, тайм-аутів тощо.

    return {

        'score': None,

        'error': f"Помилка запиту до PageSpeed API: {e}"

    }

except Exception as e:

    # Обробка інших можливих помилок

    return {

        'score': None,

```

```
'error': f'Невідома помилка: {e}'
```

```
}
```

```
# app/seo_tools/technical_analyzer.py
```

```
import requests
```

```
from urllib.parse import urlparse, urljoin
```

```
def check_technical_files(page_url: str):
```

```
    """
```

Перевіряє наявність файлів robots.txt та sitemap.xml на сайті.

Args:

page_url (str): URL будь-якої сторінки сайту для визначення домену.

Returns:

Словник з результатами перевірки.

```
"""
```

```
try:
```

```
    # Визначаємо базовий URL сайту (наприклад, https://example.com)
```

```
    parsed_url = urlparse(page_url)
```

```
    base_url = f"{parsed_url.scheme}://{parsed_url.netloc}/"
```

```
except Exception:
```

```
    # Повертаємо помилку, якщо початковий URL невалідний
```

```
    return {
```

```
        'error': 'Не вдалося розпізнати домен з наданого URL.'
```

```
    }
```

```
robots_url = urljoin(base_url, 'robots.txt')
```

```
sitemap_url = urljoin(base_url, 'sitemap.xml')
```

```
results = {  
  
    'robots_txt': {'found': False, 'url': robots_url},  
  
    'sitemap_xml': {'found': False, 'url': sitemap_url}  
  
}
```

```
headers = {  
  
    'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36  
(KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36'  
  
}
```

```
# Перевіряємо robots.txt
```

```
try:
```

```
    response = requests.get(robots_url, timeout=5, headers=headers)
```

```
if response.status_code == 200:

    results['robots_txt']['found'] = True

except requests.exceptions.RequestException:

    pass # Ігноруємо помилки, результат залишиться False


# Перевіряємо sitemap.xml

try:

    response = requests.get(sitemap_url, timeout=5, headers=headers)

    if response.status_code == 200:

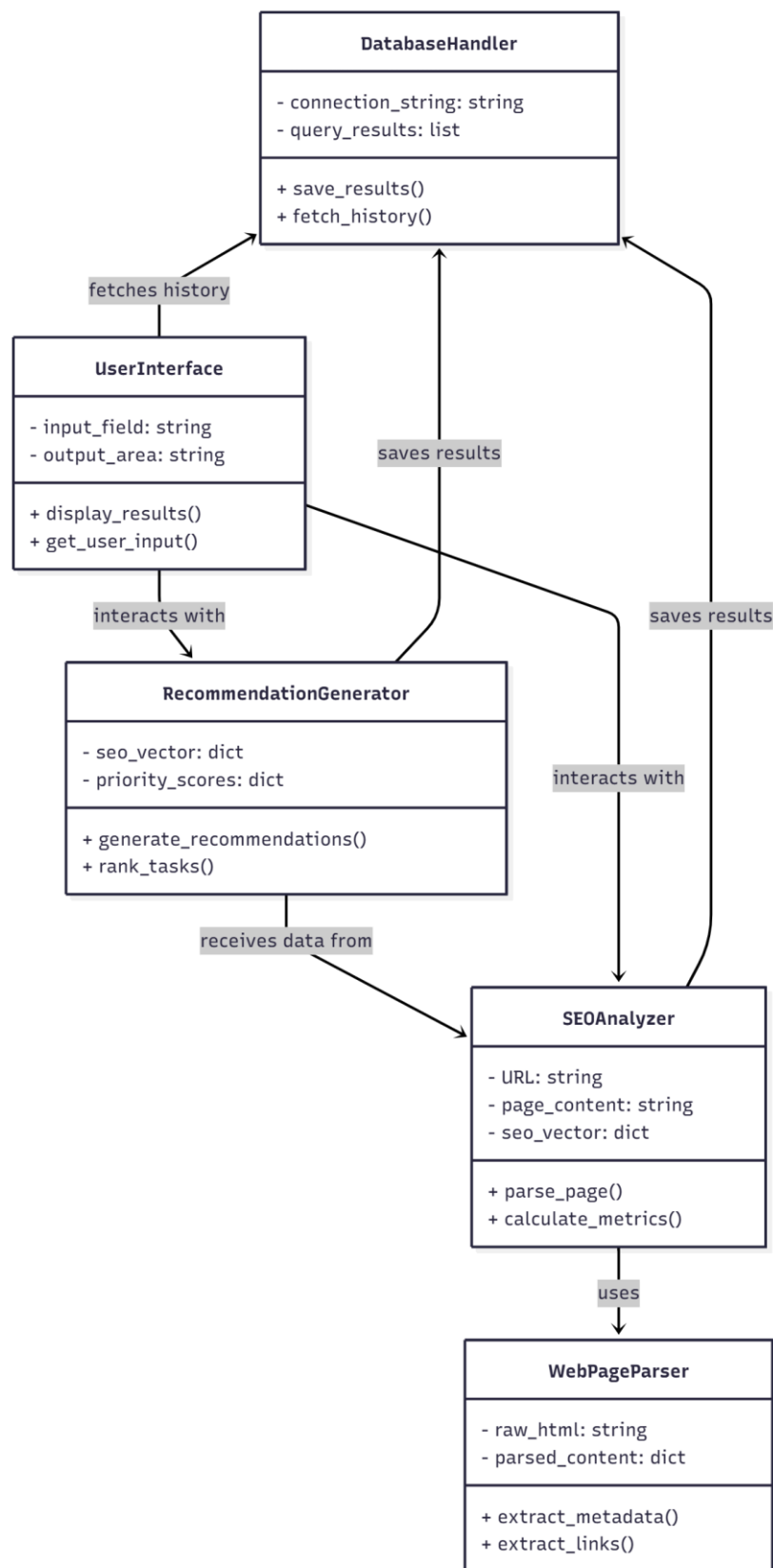
        results['sitemap_xml']['found'] = True

except requests.exceptions.RequestException:

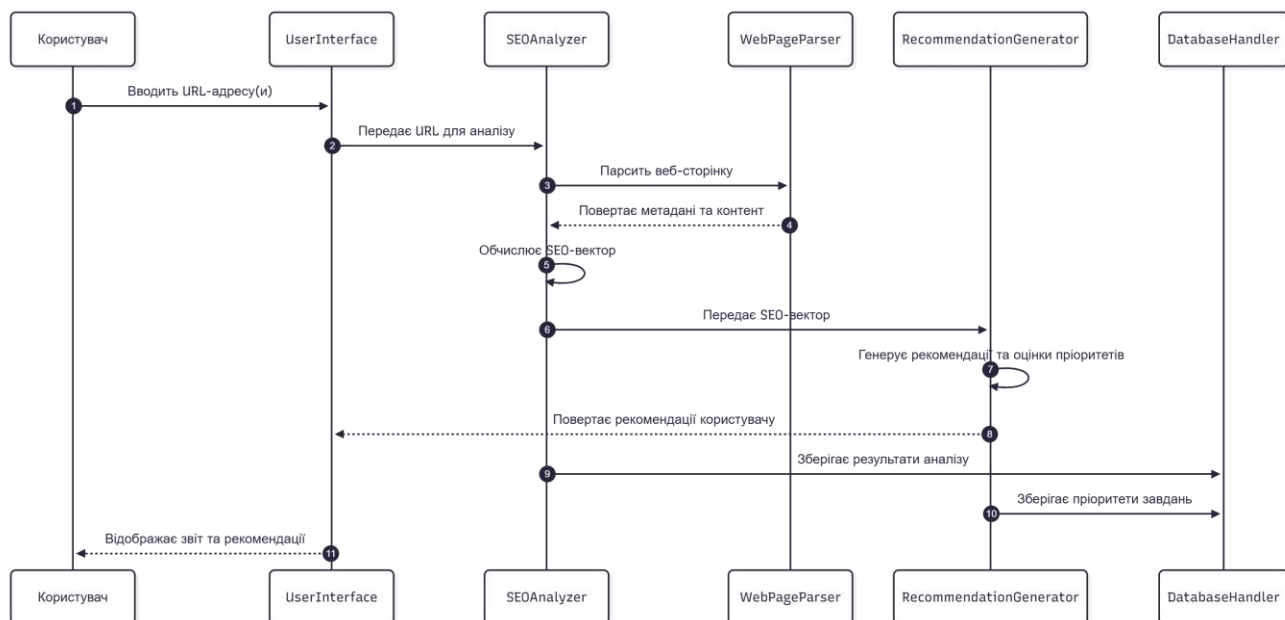
    pass # Ігноруємо помилки, результат залишиться False

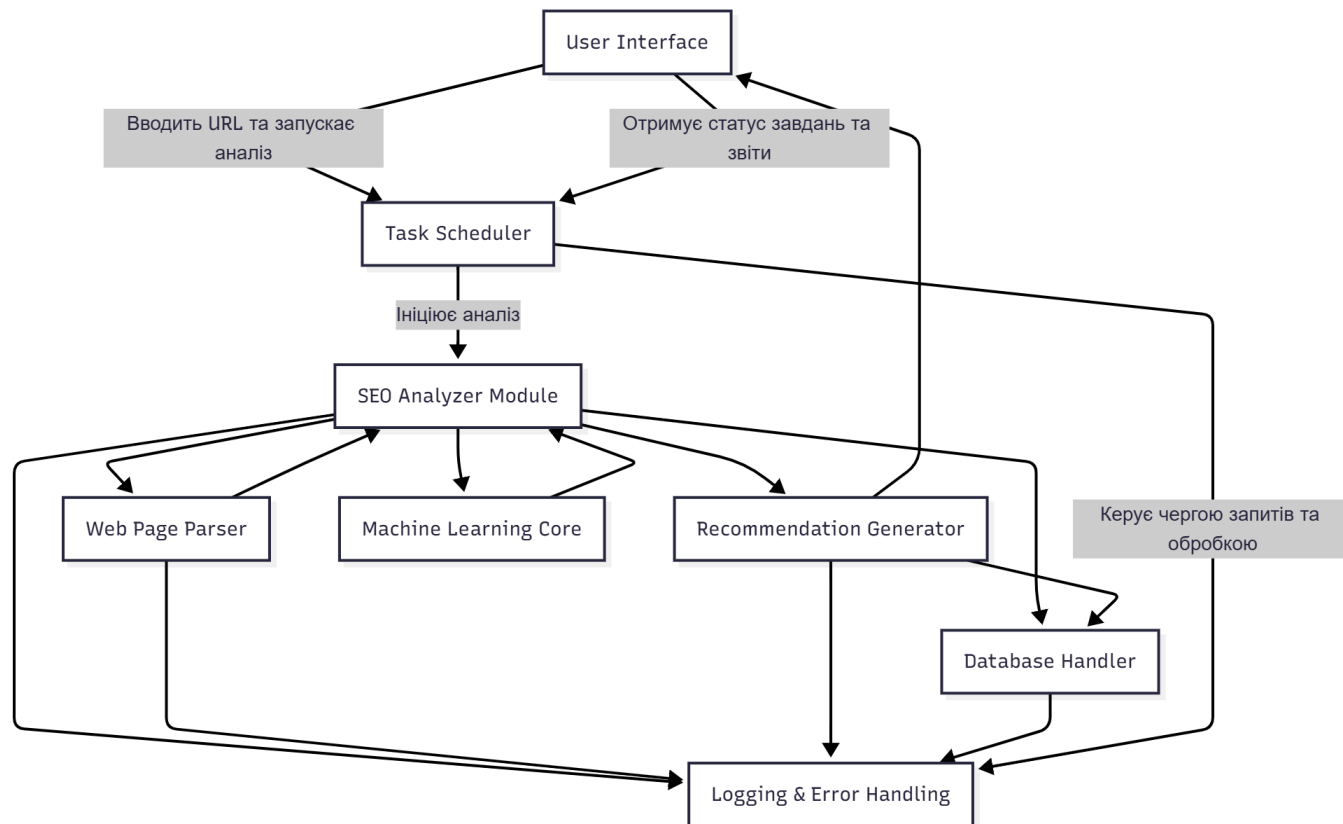

return results
```

ДОДАТОК Б: Клас-діаграма



ДОДАТОК В: Діаграми послідовності



ДОДАТОК Г: Діаграма компонентів

ДОДАТОК Д: Інструкція користувача

1. Доступ до інструменту

Відкрийте браузер і введіть адресу програми.

Наприклад: <http://127.0.0.1:5000/>

Після завантаження сторінки ви побачите форму для введення URL-адрес та кнопку "Аналізувати".

2. Введення URL для аналізу

У текстовому полі під заголовком "SEO Аналізатор Сторінок" введіть URL веб-сторінки, яку бажаєте перевірити.

Для аналізу кількох сторінок одночасно введіть кожен URL з нового рядка:

<https://example1.com>

<https://example2.com/services/>

www.mynewblog.net

Примітка щодо формату:

Якщо URL без <http://> або <https://>, система автоматично додасть <https://>.

Рекомендується вводити повний URL для точності аналізу.

Обмеження:

Максимальна кількість URL за один раз залежить від конфігурації сервера.

3. Запуск аналізу

Після введення URL натисніть кнопку "Аналізувати".

Система перенаправить вас на сторінку результатів, де виконується обробка введених URL.

Час обробки залежить від кількості URL та складності сторінок.

4. Структура сторінки результатів

На сторінці результатів для кожної проаналізованої сторінки відображаються:

SEO Вектор

Прогнозовані пріоритети SEO-завдань

Текстові рекомендації

Конкурентний аналіз

5. Додаткові функції

Експорт звітів

6. Поради щодо використання

Аналізуйте сторінки регулярно, щоб відслідковувати динаміку SEO-показників.

Використовуйте пріоритезовані рекомендації для швидкого покращення рейтингу.

Порівнюйте результати конкурентів, щоб визначити слабкі місця вашого контенту.

Перевіряйте великі сторінки окремими блоками, щоб зменшити час обробки.

7. Приклади використання

Один URL:

Введіть <https://example.com> → натисніть Аналізувати → перегляньте SEO Вектор і рекомендації.

Кілька URL:

<https://example1.com>

<https://example2.com/services/>

<https://example3.net/blog/>

Система обробить всі адреси одночасно та сформує звіти для кожної.

ДОДАТОК Е: Інтерфейс програми

SEO Аналізатор

SEO Аудит

Аналіз конкурентів

Про проект

🔍

Аналіз SEO конкурентів

Порівняйте свій сайт з конкурентами з пошукової видачі Google.
Отримайте конкретні рекомендації щодо покращення позицій.

Ключове слово

Введіть ключове слово, за яким хочете проаналізувати конкурентів

URL вашого сайту

Сторінка, яку хочете оптимізувати під це ключове слово

СерпAPI ключ

Отримайте безкоштовно на serpapi.com (100 запитів безкоштовно)

Кількість конкурентів для аналізу

Рекомендовано 25-30 для якісного аналізу

➡️ Провести аналіз конкурентів

🔴 Як це працює?

➤