МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ
Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему: **«ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ АВТОМАТИЧНОГО SEO-АУДИТУ ВЕБСАЙТІВ»**

Виконав: студент 2 курсу, групи 8.1264
спеціальності 126 Інформаційні системи та технології
(шифр і назва спеціальності)
освітньої програми Інформаційні системи та штучний інтелект
(назва освітньої програми)
С.Д. Левіт
(ініціали та прізвище)
Керівник доцент кафедри комп'ютерних наук, к.т.н Матвіїшина Н.В.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)
Рецензент завідувач кафедри фундаментальної та прикладної математики, професор, д.т.н. Гребенюк С.М.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Запоріжжя 2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет математичний

Кафедра комп'ютерних наук

Рівень вищої освіти магістр

Спеціальність 126 Інформаційні системи та технології
(шифр і назва)

Освітня програма Інформаційні системи та штучний інтелект

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерних наук,
д.т.н., професор

_____ Шило Г.М.
(підпис)

" 05 " 03 2025 р.

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Левіту Станіславу Дмитровичу
(прізвище, ім'я та по-батькові)

1. Тема роботи Використання штучного інтелекту для автоматичного SEO-аудиту вебсайтів

керівник роботи Матвіїшина Надія Вікторівна, к.т.н., доцент
(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від « 28 » лютого 2025 року № 326-с

2. Строк подання студентом роботи 03.12.2025

3. Вихідні дані до роботи 1. Постановка задачі.
2. Перелік літератури, інформаційних джерел

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Постановка задачі.
2. Основні теоретичні відомості що використання SEO-аудиту.
3. Створення прототипу та аналіз результатів

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
презентація

6. Консультанти розділів роботи

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

7. Дата видачі завдання 03.03.2025

К А Л Е Н Д А Р Н И Й П Л А Н

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	28.03.2025	
2.	Збір вихідних даних.	20.04.2025	
3.	Обробка методичних та теоретичних	14.05.2025	
	джерел.		
4.	Розробка першого та другого розділу.	13.09.2025	
5.	Розробка третього розділу.	11.10.2025	
6.	Оформлення та нормоконтроль		
	кваліфікаційної роботи магістра.	27.11.2025	
7.	Захист кваліфікаційної роботи.	17.12.2025	

Студент _____ С.Д.Левіт
(підпис) (ініціали та прізвище)

Керівник роботи _____ Н.В.Матвіїшина
(підпис) (ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер _____ О.Г.Спиця
(підпис) (ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота магістра «Використання штучного інтелекту для автоматичного SEO-аудиту вебсайтів»: 63 с., 13 рис., 12 джерел, 6 додатків.

АВТОМАТИЗАЦІЯ SEO-АУДИТУ, АВТОМАТИЗОВАНЕ ВИЯВЛЕННЯ SEO-ПОМИЛОК, АНАЛІЗ ВЕБСАЙТІВ НА ОСНОВІ ШІ, ЗБІР ТА ОБРОБКА SEO-ДАНИХ, ІНТЕГРАЦІЯ АІ У ВЕБ-АНАЛІТИКУ, ТЕХНІЧНИЙ SEO-АУДИТ, ШТУЧНИЙ ІНТЕЛЕКТ У ЦИФРОВОМУ МАРКЕТИНГУ.

Об’єкт дослідження – процес пошукової оптимізації та технічного аудиту вебсайтів.

Предмет дослідження – методи та моделі застосування штучного інтелекту для автоматизації SEO-аудиту.

Мета роботи: розробити концепцію та прототип системи автоматизованого SEO-аудиту вебсайтів із використанням технологій штучного інтелекту для підвищення ефективності аналізу та оптимізації.

Методи дослідження – емпіричні (спостереження, експеримент) та теоретичні (аналіз, індукція).

Вперше спроектовано систему SEO-аудиту з використанням методів машинного навчання та створено її спрощений прототип. За допомогою прототипу виконано SEO-аудит вебсторінки, після чого проведено її оптимізацію згідно з отриманими рекомендаціями. Зафіксовано позитивний вплив внесених змін на видимість сторінки у пошуковій системі Google.

Отримані результати підтверджують ефективність застосування ШІ для автоматизації SEO-аудиту: це дозволяє зменшити час на аналіз для досвідчених SEO-фахівців та допомагає новачкам швидше опанувати ключові принципи оптимізації, оскільки система чітко визначає основні недоліки сторінки з погляду SEO.

SUMMARY

Master's Qualifying Theses «Using AI for Automatic SEO-audit of Websites»: 63 pages, 13 figures, 12 references, 6 annexes.

SEO-AUDIT AUTOMATION, AUTOMATED SEO-ERROR DETECTION, AI-BASED WEBSITE ANALYSIS, SEO DATA COLLECTION AND PROCESSING, AI INTEGRATION INTO WEB ANALYTICS, TECHNICAL SEO-AUDIT, ARTIFICIAL INTELLIGENCE IN DIGITAL MARKETING.

Object of the study – the process of search engine optimization and technical audit of websites.

Subject of the study – methods and models for applying artificial intelligence to automate SEO audits.

Aim of the study: to develop a concept and prototype of an automated SEO website audit system using artificial intelligence technologies to increase the efficiency of analysis and optimization.

Methods of research – empirical (observation, experiment) and theoretical (analysis, induction).

For the first time, a SEO-audit system using machine learning methods has been designed, and a simplified prototype of this system has been created. Using this prototype, a SEO audit of a webpage was conducted, followed by optimization of the page according to the obtained recommendations. A positive impact on the webpage's visibility in the Google search engine was recorded.

The results confirm the effectiveness of applying AI for automating SEO audits: it allows experienced SEO specialists to reduce the time needed for analysis and helps beginners improve their skills and gain new knowledge, as the system clearly identifies the key shortcomings of a webpage from an SEO perspective.

ЗМІСТ

Завдання на кваліфікаційну роботу……..2
Реферат 4
Summary 4
Вступ 8
1 Теоретичне обґрунтування SEO-аудиту та роль інструментів машинного навчання 11
1.1 Загальна теорія SEO 11
1.2 Рекомендації щодо SEO від Google 13
1.3 Вплив штучного інтелекту на цифровий маркетинг 16
1.4 Найпопулярніші SEO-інструменти 18
1.5 Висновки до розділу 1 24
2 Планування, архітектура та технологічний стек інтелектуальної інформаційної системи seo-аудиту 26
2.1 Загальні дії перед переходом до реалізації системи SEO-аудиту 26
2.2 Вибір технологій, архітектурний підхід та стек інструментів28_Toc122294030
2.3 Опис процесу розробки 31
2.4 Висновки до розділу 2 37
3 Реалізація прототипу 38
3.1 Загальна структура проєкту 38
3.2 Створення прототипу 40
3.3 Тестування прототипу 43
3.4 Аналіз відповідей 45
3.5 Можливі покращення прототипу 46
3.6 Оцінка результату 47
Висновки 51
Перелік посилань 52

Додаток А Програмний код на мові Python у файлі main.py 53
Додаток Б Програмний код на мові Python у файлі routes.py 54
Додаток В Програмний код на мові Python у файлі crawler.py 55
Додаток Г Програмний код на мові Python у файлі model_runner.py 57
Додаток Д Приклад оголошення змінної для збереження ключа API у файлі .env та перелік бібліотек для встановлення у файлі requirements.txt 60
Додаток Е Програмний код на мові Python у файлі output_fixer.py 61

ВСТУП

На сьогодняшній день цифрові технології стрімко розвиваються і одним із значущих факторів ефективності онлайн-присутності бізнесу є якісна пошукова оптимізація (SEO – з англ. Search Engine Optimization) [8]. Через високу конкуренцію в цифровому середовищі існує необхідність постійного моніторингу поведінкових, технічних так контентних показників вебресурсів. Традиційний якісний SEO-аудит потребує багато ресурсів, як часових, так і людських, а також вимагає від спеціалістів ґрунтовних знань. Саме тому зростає потреба у створенні систем, які допожуть автоматизувати аналіз та виконати комплексну оцінку сайтів при мінімальному втручанні з боку люди. Розвиток штучного інтелеку (ШІ) вже відкрив та продовжує відкривати нові можливості для підвищення ефективності SEO процесів. Наприклад вже можливо інтегрувати алгоритми машинного навчання у процес аудиту, щоб забезпечити глибший аналіз даних та виявити закономірності, щоб формувати рекомендаціх на основі поведінкових або контентих факторів.

Сьогодні існує на ринку низка інструментів, які допомагають провести SEO-аудит – Ahrefs [3], SEMrush [12], Screaming Frog [7], Serpstat [10] та інші. Однак більшість з відомих та вже згаданих інструментів працюють за фіксованими алгоритмами перевірки, та не забезпечують інтелектуальний аналіз, що враховує взаємозв’язки між факторами ранжування та контекст. Досліження використання штучного інтелекту у цифровому маркетингу стосується здебільша окремих аспектів, зокрема генерації контенту, кластеризації запитів, семантичного аналізу або прогозування поведінки користувачів. В той самий час питання комплексної автоматизації SEO-аудиту на базі штучного інтелекту залишається недостатньо опрацьованим і має потребу в подальшому розвитку.

Завдяки штучному інтелекту підходи до цифрового маркетингу поступово трансформуються. Завдяки алгоритмам машиного навчання можливо автоматизувати аналітику, створити персоналізований контент, оптимізувати рекламні кампанії та прогнозувати тенденції ринку. У SEO штучний інтелект зазвичай використовують здебільша як інструмент для генерації текстового або візуального контенту, інколи для генерації програмного коду для часткової автоматизації рутинних задач, наприклад для обробки файлів з даними, таблиць, тощо, але це все ще не повна автоматизація.

Тому метою дослідження є розробити концепцію та прототип системи автоматизованого SEO-аудиту вебсайтів із використанням технологій штучного інтелекту для підвищення ефективності аналізу та оптимізації вебресурсів. Відповідно сформоване наступне завдання:

- проаналізувати існуючі підходи та інструменти проведення SEO-аудиту;
- дослідити можливості застосування штучного інтелекту для автоматизації SEO-процесів;
- розробити архітектуру системи автоматизованого аудиту;
- реалізувати програмний прототип, що виконує аналіз основних SEO-параметрів вебсайтів;
- оцінити ефективність запропонованого рішення порівняно з традиційними методами.

Об’єкт дослідження – процес пошукової оптимізації та технічного аудиту вебсайтів. Предмет дослідження – методи та моделі застосування штучного інтелекту для автоматизації SEO-аудиту.

Уперше запропоновано комплексний підхід до автоматизації SEO-аудиту, який поєднує машинне навчання, обробку природної мови та аналітику технічних параметрів. Розроблено архітектуру системи, здатної самостійно виявляти SEO-проблеми та формувати рекомендації. Практичне значення дослідження полягає у створенні прототипу інтелектуальної

системи, яка може бути інтегрована у роботу SEO-агенцій, відділів цифрового маркетингу або вебстудій.

Запропоноване рішення дозволяє скоротити час проведення аудиту, зменшити людський фактор у процесі оцінки та підвищити точність виявлення проблем, що впливають на позиції вебсайтів у пошукових системах.

1 ТЕОРЕТИЧНЕ ОБҐРУНТУВАННЯ SEO-АУДИТУ ТА РОЛЬ ІНСТРУМЕНТІВ МАШИННОГО НАВЧАННЯ

1.1 Загальна теорія SEO

Пошукова оптимізація (англ. Search Engine Optimization, далі – SEO) – це комплекс заходів, спрямованих на підвищення видимості вебресурсу у результатах органічної видачі пошукових систем, таких як Google, Bing чи Yahoo.

Метою SEO є забезпечення стабільного потоку цільового трафіку без використання платних рекламних каналів, шляхом покращення відповідності сайту запитам користувачів та вимогам пошукових алгоритмів.

SEO є однією з ключових складових цифрового маркетингу, оскільки впливає на репутацію бренду, рівень залученості користувачів і рентабельність бізнесу в онлайновому середовищі. Ефективна оптимізація поєднує технічні аспекти веброзробки, лінгвістичний аналіз, аналітику поведінки користувачів і маркетингову стратегію. У структурі SEO прийнято виділяти три ключові напрями:

On-page SEO (внутрішня оптимізація) – комплекс робіт, спрямованих на покращення елементів сторінки:

- оптимізація метаданих (title, description, alt);
- правильна структура заголовків (h1–h6);
- внутрішня перелінковка;
- релевантність текстового контенту;
- швидкість завантаження сторінки та адаптивність дизайну.

Off-page SEO (зовнішня оптимізація) – дії, спрямовані на підвищення авторитету сайту за межами самого ресурсу:

- побудова профілю зворотних посилань (backlink profile);

- згадки бренду у змі та соцмережах;
- крауд-маркетинг;
- робота з онлайн-рейтингами та відгуками.

Technical SEO (технічна оптимізація) – забезпечення коректної індексації сайту пошуковими ботами:

- налаштування файлів robots.txt, sitemap.xml;
- усунення дублікатів сторінок і редиректів;
- правильне використання канонічних тегів (canonical);
- оптимізація структури url;
- впровадження структурованих даних (schema.org).

У професійній практиці SEO використовується широкий набір термінів, що описують як технічні, так і поведінкові аспекти. Нижче наведено одні з найуживаніших понять:

- SERP (Search Engine Results Page) – сторінка результатів пошукової системи;
- CTR (Click-Through Rate) – відсоток кліків на посилання у SERP відносно кількості показів;
- crawl – процес сканування вебсторінок пошуковими ботами;
- indexing – додавання вебсторінок до бази даних пошукової системи;
- keyword – ключове слово або фраза, що характеризує зміст сторінки;
- backlink – зворотне посилання з іншого сайту;
- anchor text – текст, що є клікабельною частиною посилання;
- bounce rate – частка користувачів, які залишають сторінку без взаємодії;
- organic traffic – трафік, отриманий із безкоштовних результатів пошуку;
- canonical tag – тег, що визначає основну версію сторінки для уникнення дублювання контенту;

– core web vitals – набір показників Google, що оцінюють зручність користування сайтом (LCP, FID, CLS).

Ця термінологія є фундаментом для формування системи автоматичного SEO-аудиту, оскільки саме ці параметри підлягають аналітичному моніторингу.

SEO-аудит – це систематичний процес аналізу вебсайту з метою виявлення технічних, контентних і структурних недоліків, що впливають на його видимість у пошукових системах. Аудит дозволяє оцінити поточний стан сайту, визначити напрямки для покращення та сформувати рекомендації для подальшої оптимізації.

Основні етапи SEO-аудиту:

– технічний аналіз: перевірка доступності сторінок для індексації, коректності редиректів, швидкості завантаження, мобільної адаптивності, SSL-сертифікатів тощо;
– контентний аналіз: оцінка якості текстів, унікальності, релевантності ключових слів, структури заголовків, метаданих і внутрішньої перелінковки;
– аналіз зовнішніх факторів: перевірка профілю зворотних посилань, авторитету домену (Domain Authority, Trust Flow) і репутації сайту в мережі;
– поведінковий аналіз: аналіз показників взаємодії користувачів із сайтом – часу перебування, глибини перегляду, відмов;
– висновки і рекомендації: формування звіту з конкретними діями для підвищення позицій у SERP.

1.2 Рекомендації щодо SEO від Google

Google, як провідна пошукова система у світі, визначає стандарти якості контенту та технічної оптимізації вебсайтів через власні рекомендації

– Google Search Essentials (раніше відомі як Webmaster Guidelines). Ці документи містять офіційні вимоги до створення, індексації та ранжування вебресурсів. Їхнє дотримання є необхідною умовою для коректного відображення сайту у пошуковій видачі та запобігання санкціям з боку пошукових алгоритмів. Мета цих гайдлайнів – забезпечити, щоб у топі результатів пошуку опинялися сайти, які надають корисний, достовірний і доступний користувачеві контент. Google наголошує, що будь-які маніпуляції з метою штучного підвищення позицій (наприклад, прихований текст, купівля посилань, автоматична генерація контенту низької якості) можуть призвести до зниження рейтингу або видалення сайту з індексу.

Офіційні рекомендації Google умовно поділяються на три групи:

а) Технічні вимоги (Technical Requirements):

Вони описують, як забезпечити доступність сайту для сканування та індексації пошуковими ботами. Основні принципи:

- перевірка доступності сайту для Googlebot;
- правильне використання robots.txt;
- створення та оновлення sitemap.xml;
- адаптивний дизайн (mobile-first indexing);
- швидке завантаження сторінок (Core Web Vitals);
- використання безпечного протоколу HTTPS.

б) Рекомендації щодо контенту (Content Guidelines):

Основна увага приділяється цінності контенту для користувача. Гугл рекомендує:

- створювати унікальний, достовірний і зрозумілий контент;
- розкривати тему повно й логічно, без надмірного “спаму” ключовими словами;
- використовувати структуру заголовків (h1–h3) для логічної організації тексту;
- регулярно оновлювати інформацію;

– забезпечувати релевантність між пошуковими запитами та змістом сторінок.

в) Політика проти спаму (Spam Policies):

Визначає дії, які вважаються маніпулятивними або такими, що порушують принципи чесної оптимізації. До них належать:

– прихований текст і посилання;
– автоматично створений неякісний контент;
– клоакінг (показ різного контенту користувачеві й пошуковику);
– дубльований контент;
– участь у посилальних фермах і pbn-мережах;
– використання шкідливого пз або фішингових елементів.

Важливим компонентом гайдлайнів Google є принцип E-E-A-T (Experience, Expertise, Authoritativeness, Trustworthiness – досвід, експертність, авторитетність, довіра). Ця концепція визначає, як алгоритми оцінюють якість контенту та надійність джерела.

– experience (досвід): контент має створюватися авторами, які мають реальний досвід у темі. Expertise (експертність): тексти повинні містити точну, підтверджену інформацію;
– authoritativeness (авторитетність): сайт має бути визнаним джерелом у своїй ніші (наявність посилань, згадок, позитивної репутації);
– trustworthiness (довіра): сторінки мають бути безпечними (HTTPS), містити достовірні контакти, прозору політику конфіденційності, коректну інформацію про автора.

Алгоритми Google, зокрема Helpful Content Update, дедалі більше орієнтовані на виявлення контенту, створеного для людей, а не для пошукових систем. Це означає, що якість, глибина розкриття теми та користь для читача стають вирішальними чинниками SEO-успіху.

Показники Core Web Vitals є частиною технічних вимог Google, що вимірюють якість взаємодії користувача із сайтом. До них належать:

- LCP (Largest Contentful Paint) – час відображення основного елемента сторінки (має бути ≤2,5 с);
- FID (First Input Delay) – затримка між дією користувача та реакцією сайту (≤100 мс);
- CLS (Cumulative Layout Shift) – стабільність візуального контенту під час завантаження (≤0,1).

Високі показники цих метрик позитивно впливають на позиції сайту в результатах пошуку, оскільки вони свідчать про комфорт користувацького досвіду. Для розробки системи автоматизованого SEO-аудиту гайдлайни Google відіграють роль еталонного стандарту.

На їх основі можна формувати:

- чек-листи перевірки технічних параметрів сайту (індексація, структура URL, HTTPS, mobile-friendly);
- модулі аналізу контенту, що оцінюють унікальність, релевантність і природність ключових слів;
- алгоритми виявлення спаму;
- оцінку відповідності сторінок принципу E-E-A-T;
- автоматизовані звіти Core Web Vitals.

Таким чином, у контексті теми даної кваліфікаційної роботи гайдлайни Google виступають методологічною основою для побудови логіки системи аудиту, визначення параметрів аналізу та формування критеріїв оцінки SEO-якості вебресурсів.

1.3 Вплив штучного інтелекту на цифровий маркетинг

Штучний інтелект (ШІ) став одним із ключових чинників цифрової трансформації сучасного бізнесу. Його впровадження докорінно змінює підходи до планування, реалізації та оцінки маркетингових стратегій.

У цифровому маркетингу ШІ використовується для автоматизації аналітики, прогнозування поведінки споживачів, персоналізації контенту, управління рекламою та оптимізації взаємодії з клієнтами.

Основна цінність штучного інтелекту полягає у здатності швидко обробляти великі обсяги даних (Big Data), виявляти закономірності та робити висновки, які людина не може сформулювати інтуїтивно. Це дозволяє маркетологам приймати дано-орієнтовані рішення (data-driven decisions) і переходити від реактивного до прогнозного управління маркетинговими процесами.

Основні напрями застосування ШІ в цифровому маркетингу:

а) персоналізація контенту та рекомендацій.

Алгоритми машинного навчання аналізують поведінку користувачів, історію переглядів, кліки, місце розташування та інтереси, створюючи індивідуальні рекомендації. Прикладом є системи, які формують персоналізовану стрічку новин, пропонують товари чи контент, що найбільше відповідає очікуванням користувача.

б) прогнозна аналітика (Predictive Analytics).

Завдяки ШІ компанії можуть передбачати майбутню поведінку користувачів, наприклад, ймовірність відтоку клієнтів, зміни попиту чи ефективність рекламних кампаній. Це дозволяє оптимізувати бюджети й мінімізувати ризики.

в) автоматизація маркетингових кампаній.

Платформи типу Google Ads або Meta Ads Manager активно використовують алгоритми машинного навчання для автоматичного підбору цільової аудиторії, ставок і форматів реклами. Це дає змогу підвищити ROI (Return on Investment) за рахунок динамічної адаптації стратегії.

г) чат-боти та голосові асистенти.

Технології обробки природної мови (Natural Language Processing, NLP) дозволяють створювати системи автоматизованої підтримки користувачів, які здатні вести діалог, розпізнавати наміри та надавати релевантні відповіді.

Такі рішення знижують навантаження на персонал і підвищують рівень залучення клієнтів.

д) Аналіз контенту та репутації.

Алгоритми ШІ застосовуються для виявлення тональності відгуків (sentiment analysis), відстеження згадок бренду у соцмережах, аналізу конкурентів і трендів ринку. Це дозволяє компаніям реагувати на зміни у сприйнятті бренду в режимі реального часу.

е) Генерація контенту.

Використання моделей типу GPT, Claude чи Gemini спрощує створення текстових матеріалів – від заголовків до повноцінних статей і рекламних оголошень. ШІ може формувати оптимізований контент із урахуванням пошукових запитів, що робить його цінним інструментом для SEO та контент-маркетингу.

1.4 Найпопулярніші SEO-інструменти

Сучасний SEO-ринок характеризується високою конкуренцією між аналітичними платформами, які забезпечують збір, систематизацію та інтерпретацію великих обсягів даних. Найвідоміші з них – Ahrefs, SEMrush, Serpstat, Moz Pro [2], Screaming Frog, Google Search Console [1] і Google Analytics [11].

Їхня основна функціональність охоплює технічний аудит, аналіз ключових слів, моніторинг позицій, аудит зворотних посилань, оцінку конкурентного середовища та генерацію звітів. З розвитком штучного інтелекту ці інструменти поступово інтегрують елементи машинного навчання для автоматизації аналізу, прогнозування та оптимізації.

Ahrefs – одна з найпотужніших платформ для роботи з посиланнями та аналізу контенту. Компанія активно застосовує машинне навчання (ML) для

обробки великих обсягів даних про зворотні посилання (понад 30 трлн URL) та оцінки авторитетності сторінок (URL Rating) і доменів (Domain Rating).

AI-моделі в Ahrefs використовуються для:

– кластеризації сторінок за темами і типами контенту;

– виявлення спам-посилань на основі поведінкових патернів;

– прогнозування трафіку з урахуванням динаміки serp.

У 2024 році Ahrefs (рис. 1.1) представив «Content Gap AI Model», який аналізує конкурентний контент і пропонує теми для створення нових сторінок з вищим потенціалом ранжування.

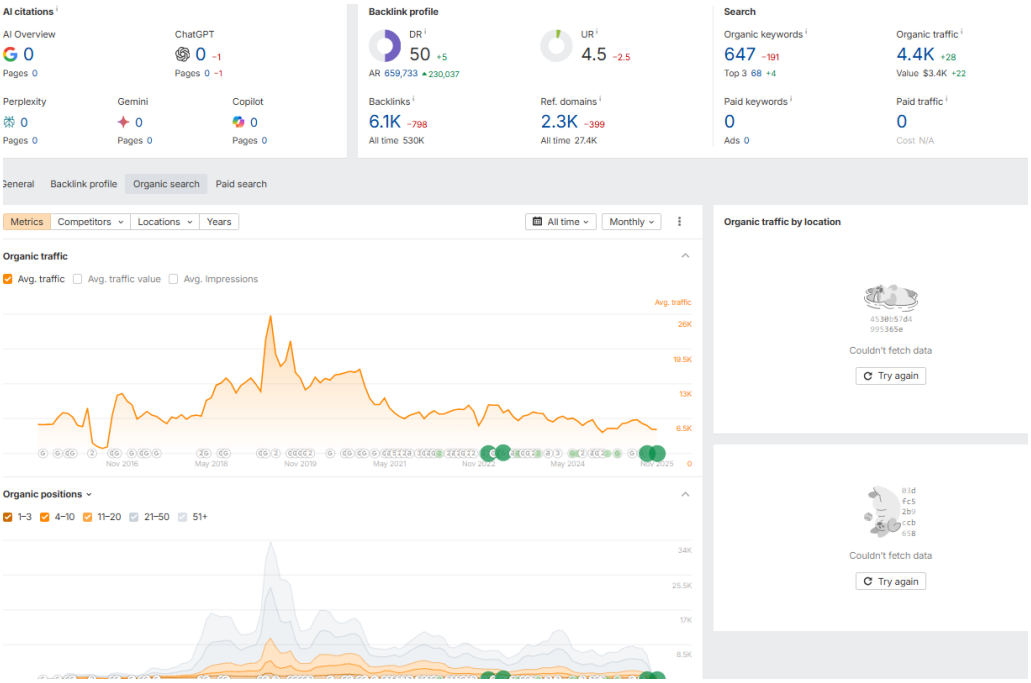


Рисунок 1.1 – Ahrefs

SEMrush (рис. 1.2) – комплексна платформа цифрового маркетингу, де ШІ відіграє все більшу роль у модулях:

– keyword intent detection – алгоритми ШІ визначають намір користувача (інформаційний, комерційний, транзакційний), допомагаючи формувати стратегію контенту;

– AI writing assistant – інструмент, інтегрований із GPT-моделями, оптимізує тексти під SEO, зберігаючи природність мовлення;

– traffic analytics – модуль прогнозує обсяги трафіку конкурентів на основі машинного навчання, поєднуючи дані із сотень тисяч джерел;
– backlink audit – автоматично оцінює ризики посилань і пропонує, які з них варто видалити або дізавовити.

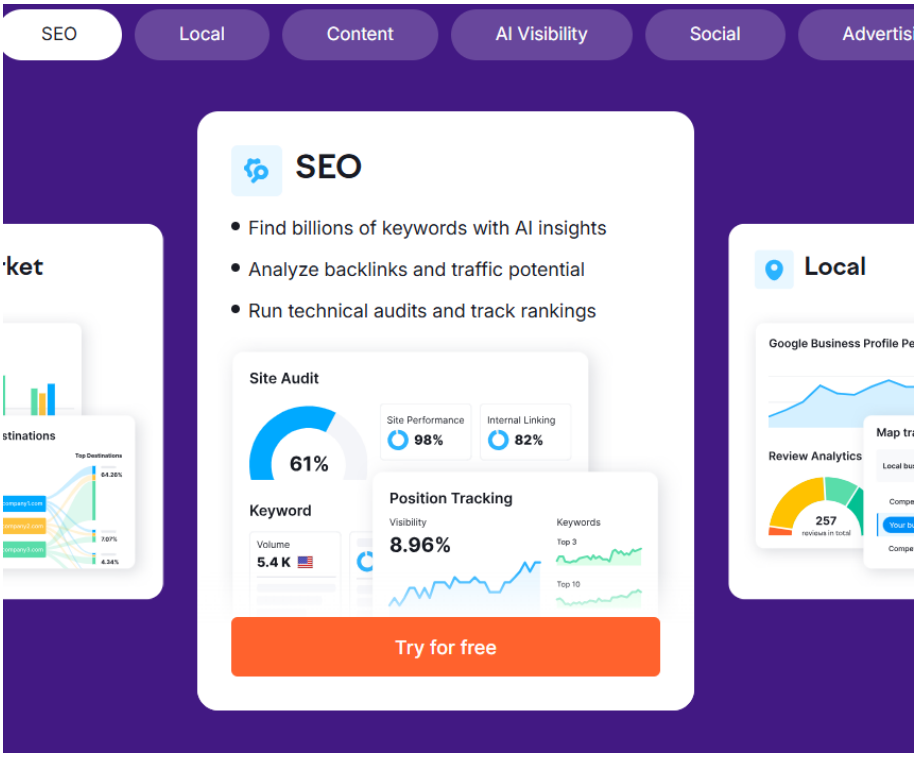


Рисунок 1.2 – Semrush

Таким чином, SEMrush інтегрує ШІ як у контент-аналіз, так і в аналітику ризиків, що суттєво скорочує ручну роботу SEO-фахівців.

Serpstat (рис. 1.3) позиціонує себе як «SEO-платформу для зростання бізнесу на основі даних». Використання ШІ у Serpstat охоплює такі напрями:

– AI clusterization tool – автоматично групує ключові слова за семантичними кластерами, використовуючи векторні представлення слів (word embeddings);
– AI text analysis – інструмент для оцінки контенту сторінок конкурентів і рекомендацій щодо щільності ключових слів;

– SERP features prediction – модель прогнозує ймовірність появи сторінки у фрагментах пошуку (featured snippets, People Also Ask);
– rank tracking AI – алгоритми виявляють аномальні зміни позицій (SERP volatility) і допомагають зрозуміти, чи пов’язані вони зі змінами алгоритмів Google.

Загально Serpstat визнається спеціластами як один з найкращих інструментів для аналізу саме ключових слів, фраз тощо.



Рисунок 1.3 – Serpstat

Screaming Frog SEO Spider (рис. 1.4) – класичний десктопний сканер сайтів, який із версії 18 інтегрує елементи штучного інтелекту у вигляді:

– автоматичного розпізнавання патернів проблем – наприклад, сторінок-дублікатів, повторюваних метаданих чи некоректних canonical-посилань;
– AI content analysis integration – модуль, що через API може передавати контент у GPT-моделі для оцінки читабельності, релевантності ключових слів і рекомендацій щодо оновлення;

– у поєднанні з Google Search Console та GA 4 Screaming Frog може виконувати напівавтоматичний технічний аудит за допомогою зовнішніх ШІ-сервісів (через API).

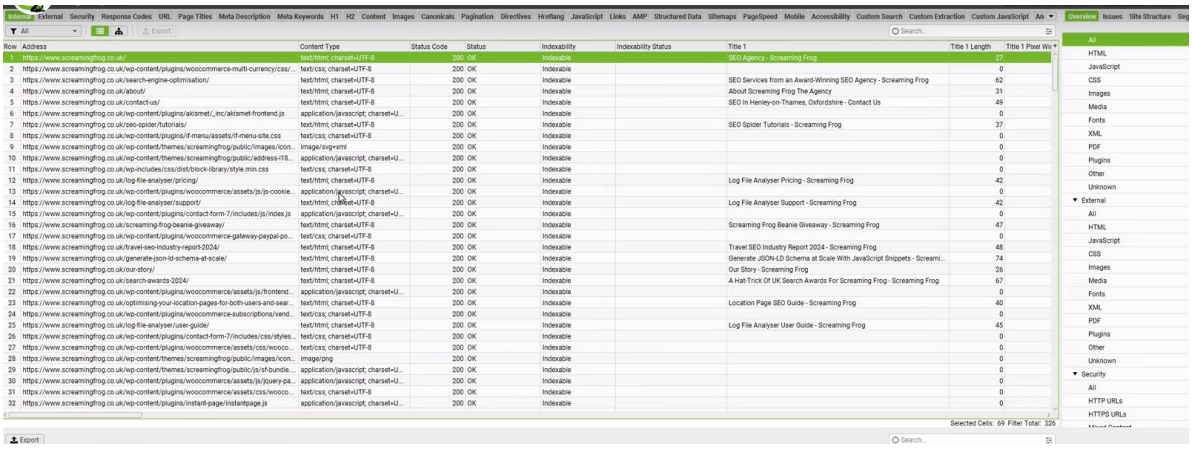


Рисунок 1.4 - Screaming Frog

Moz Pro (рис. 1.5) історично відома своїм Domain Authority (DA) – показником, що базується на машинному навчанні. У 2023 році Moz оновила свій алгоритм із використанням графових нейронних мереж (GNN) для точнішої оцінки якості доменів. Інші ШІ-компоненти:

– spam score AI – прогнозує ймовірність санкцій через неякісні посилання;
– keyword explorer AI – генерує групи ключових слів на основі латентно-семантичного аналізу;
– On-Page optimization AI – оцінює релевантність сторінок щодо запиту, використовуючи обчислювальні моделі близькості термінів.

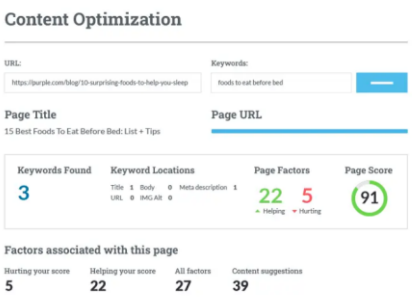


Рисунок 1.5 - Moz Pro

Google Search Console (рис. 1.6) та Google Analytics

Хоча ці інструменти не позиціонуються як «AI-продукти», у них реалізовано низку алгоритмів машинного навчання:

– anomaly detection in Search Console – автоматичне сповіщення про відхилення в індексації чи CTR;
– smart insights у GA 4 – модуль на базі TensorFlow, який прогнозує конверсії, ідентифікує поведінкові тренди користувачів;
– core web vitals prediction model (рис. 1.7) – на основі історичних даних оцінює, які сторінки можуть втратити позиції через погіршення швидкодії.

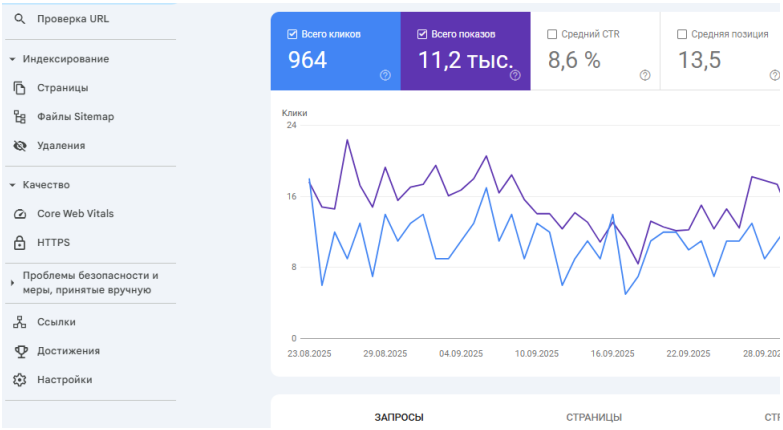


Рисунок 1.6 - Google Search Console

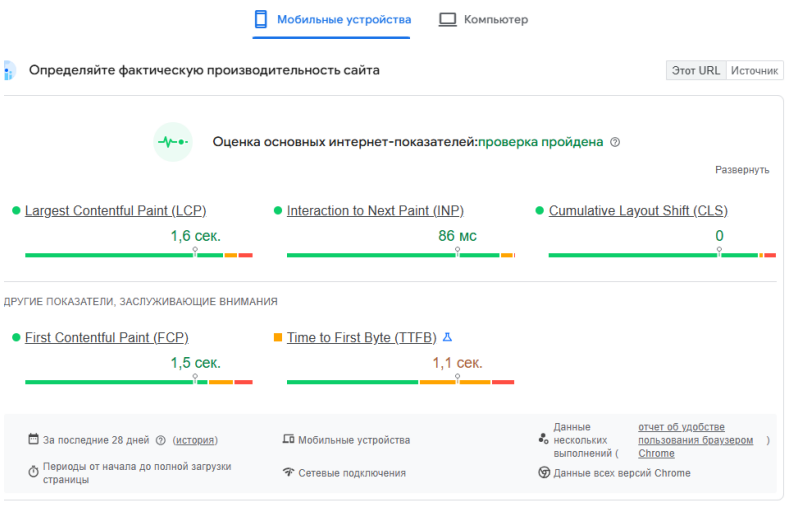


Рисунок 1.7 - Google Core Web Vitals

1.5 Висновки до розділу 1

У цьому розділі здійснено комплексний теоретичний аналіз основних аспектів пошукової оптимізації (SEO) та впливу штучного інтелекту на сучасний цифровий маркетинг.

На основі проведеного дослідження встановлено, що SEO залишається ключовим інструментом підвищення видимості вебресурсів у пошукових системах, забезпечуючи зростання органічного трафіку та покращення конкурентоспроможності бізнесу в онлайн-середовищі. Визначено, що ефективна SEO-стратегія базується на поєднанні трьох основних компонентів: технічної оптимізації, контентного наповнення та стратегії побудови профілю посилань, які мають бути реалізовані відповідно до вимог та гайдлайнів пошукових систем.

Проаналізовано офіційні рекомендації Google щодо SEO, зокрема принципи E-E-A-T (Experience, Expertise, Authoritativeness, Trustworthiness), важливість коректної індексації, оптимізації швидкодії (Core Web Vitals) та створення якісного, оригінального контенту. Ці рекомендації визначають базові орієнтири для побудови етичних та довгострокових стратегій оптимізації, де головна увага приділяється користувацькому досвіду, а не маніпуляціям алгоритмами пошуку.

Розглянуто вплив штучного інтелекту на сферу цифрового маркетингу. Встановлено, що ШІ докорінно змінює підходи до аналітики, персоналізації контенту, прогнозування поведінки користувачів та автоматизації рекламних процесів. У контексті SEO технології машинного навчання дозволяють обробляти великі масиви даних, визначати закономірності у пошукових запитах і покращувати релевантність контенту за допомогою алгоритмів природньої обробки мови (NLP).

Окрему увагу приділено аналізу сучасних SEO-інструментів, таких як Ahrefs, SEMrush, Serpstat, Screaming Frog, Moz, Google Search Console тощо. Доведено, що більшість провідних платформ уже інтегрують елементи

штучного інтелекту – від автоматичної кластеризації ключових слів і аналізу намірів користувачів до прогнозування динаміки трафіку та виявлення технічних проблем. Це свідчить про поступовий перехід від класичного аналітичного підходу до інтелектуальної автоматизації SEO-аудиту.

Загалом, результати розділу підтверджують, що штучний інтелект є ключовим драйвером еволюції SEO-технологій. Його інтеграція забезпечує новий рівень точності, швидкості та адаптивності аналітичних процесів. Водночас, повна автоматизація SEO-аудиту поки що недосяжна, оскільки інтерпретація даних, прийняття стратегічних рішень та розуміння контексту потребують участі фахівця.

Отже, у результаті проведеного аналізу:

- узагальнено теоретичні основи SEO та сучасну термінологію;
- систематизовано вимоги Google до якісного контенту та оптимізації;
- визначено основні напрями впливу ШІ на цифровий маркетинг;
- проаналізовано найпоширеніші SEO-інструменти й рівень інтеграції в них технологій ШІ.

Отримані результати створюють теоретичне підґрунтя для подальшої розробки моделі автоматизації SEO-аудиту вебсайтів на базі штучного інтелекту, що стане предметом наступних розділів дослідження.

2 ПЛАНУВАННЯ, АРХІТЕКТУРА ТА ТЕХНОЛОГІЧНИЙ СТЕК ІНТЕЛЕКТУАЛЬНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ SEO-АУДИТУ

2.1 Загальні дії перед переходом до реалізації системи SEO-аудиту

Після теоретичного опрацювання сучасних методів SEO, аналізу гайдлайнів Google та визначення потенціалу використання штучного інтелекту, виникає потреба перейти до практичної розробки системи автоматизації SEO-аудиту. На цьому етапі важливо сформувати чіткий план дій, що визначає логіку переходу від теорії до практичного вирішення поставлених задач. Загальні дії охоплюють комплекс підготовчих, організаційних та технологічних кроків, які створюють підґрунтя для побудови програмного продукту.

а) визначення функціональних вимог системи.

На першому етапі необхідно окреслити конкретні завдання, які має виконувати система: аналіз технічного стану сайту, оцінка оптимізації контенту, кластеризація ключових слів, перевірка посилальних профілів, виявлення критичних помилок тощо. Чітке формулювання вимог дозволяє задавати напрями розробки та визначати необхідні інструменти.

б) формування переліку SEO-показників, що підлягають автоматизації.

Для побудови системи потрібно визначити, які метрики будуть збиратись і аналізуватись: статуси HTTP, швидкодія, дані індексації, метадані сторінок, семантична релевантність контенту, структура внутрішніх посилань, видимість у SERP тощо. Ці показники формують каркас майбутнього аудиту.

в) вибір набору технологій та інструментів. На даному етапі формується стек технологій, який включатиме:

1) мову програмування (javascript, Python [5] тощо);

2) методи реалізації AI-модулів (NLP, кластеризація, ML-алгоритми);
3) засоби збору даних (парсинг, API пошукових інструментів, сканери сайтів);
4) системи збереження та обробки інформації.

Вибір технологій напряму визначає ефективність системи.

г) створення архітектури майбутнього рішення.

Потрібно визначити структуру системи: модулі збору даних, модулі аналізу, модуль ШІ, система звітності, а також логіку взаємодії між ними. Чітка архітектура є основою масштабованості та стабільної роботи.

д) підготовка датасетів для ШІ-модулів.

Щоб алгоритми могли аналізувати контент, кластеризувати ключові слова або оцінювати релевантність сторінок, необхідно сформувати набори даних. Це можуть бути навчальні вибірки з відкритих джерел, власні семантичні бази або дані з SEO-інструментів. Якість датасету впливає на ефективність моделей.

е) розробка алгоритмів обробки та аналізу SEO-даних.

На цьому етапі формується логіка, яка визначатиме, як саме система буде знаходити помилки, аналізувати показники, формувати рекомендації та пріоритизувати їх відповідно до критичності.

є) планування інтеграції модулів штучного інтелекту.

Важливо визначити, які саме елементи аудиту будуть автоматизовані за допомогою ШІ:

1) семантичний аналіз контенту;
2) кластеризація запитів;
3) виявлення патернів технічних помилок;
4) прогнозування впливу змін на позиції сайту;
5) генерація рекомендацій.

Це дозволяє визначити точки впровадження інтелектуальних алгоритмів.

ж) проєктування системи формування звітів.

Оскільки SEO-аудит має приносити користувачам конкретні інсайти, необхідно заздалегідь продумати структуру звітності: формат презентації результатів, пріоритизація проблем, рекомендації щодо оптимізації.

з) планування тестування та валідації.

Система потребує перевірки коректності роботи алгоритмів, точності моделі ШІ, стійкості до високих навантажень та відповідності вимогам. Важливо передбачити етапи тестування: модульне, інтеграційне, функціональне та порівняльне з результатами ручного аудиту.

Загалом, перелічені дії є фундаментом переходу до практичної реалізації системи автоматизації SEO-аудиту. Вони забезпечують методичну послідовність, мінімізують ризики на етапах розробки та гарантують, що створене рішення відповідатиме поставленим у роботі цілям.

2.2 Вибір технологій, архітектурний підхід та стек інструментів

Розробка системи автоматизованого SEO-аудиту на основі штучного інтелекту потребує ретельно продуманого вибору технологій, що забезпечать стабільність, масштабованість, продуктивність та простоту подальшого розвитку рішення. Вибір технологій визначається цілями системи, особливостями обробки великих масивів даних, необхідністю інтеграцій із зовнішніми SEO-інструментами та здатністю виконувати інтелектуальний аналіз із використанням моделей ШІ.

Для реалізації системи пропонується використання модульної сервіс-орієнтованої архітектури (SOA) або мікросервісного підходу, який забезпечує гнучкість, можливість окремого масштабування компонентів, а також швидке оновлення алгоритмів аналізу без зупинки всього сервісу. Архітектура передбачає наявність таких основних модулів:

а) модуль збору даних (Crawler / Scraper)

Виконує сканування вебсайту, збір технічних показників, HTML-структури [6], метаданих, внутрішніх посилань, швидкості завантаження та інших SEO-сигналів.

б) модуль інтеграцій із зовнішніми API

Отримує додаткові дані з Ahrefs, Semrush, Serpstat, Google Search Console, Google PageSpeed Insights тощо.

в) аналітичний модуль (AI-Analyzer) застосовує моделі ШІ для:

1) класифікації SEO-проблем;
2) генерації рекомендацій;
3) кластеризації запитів;
4) виявлення аномалій;
5) автоматизації контентного аналізу;

г) модуль звітності (Report Engine) генерує звіти у форматах PDF/HTML/JSON, візуалізує дані, формує підсумкові висновки;

д) інтерфейс користувача (Frontend) забезпечує взаємодію користувача з системою, відображає метрики, графіки, статуси перевірок;

е) база даних (Data Storage) містить результати аудиту, проміжні обчислення, параметри сайтів та історичні дані.

Таке розбиття дозволяє ефективно оновлювати окремі частини системи та реалізувати незалежний розвиток модулів.

Для створення системи автоматизованого SEO-аудиту оптимальним є поєднання web-технологій, інструментів машинного навчання та засобів обробки великих даних.

а) Back-end

1) Python – основна мова для аналітики, завдяки потужній екосистемі (NumPy, Pandas, Scikit-learn, BeautifulSoup, Selenium);
2) FastAPI / Flask -– для розробки швидкого та легкого REST API;
3) Node.js (опціонально) – для високонавантажених або стрімких операцій (наприклад, паралельний краулінг);

б) скрапінг і краулінг:

1) Selenium – для обробки JavaScript-сайтів;
2) Requests / aiohttp – для легких HTTP-запитів;
3) Scrapy – для швидкого масштабованого веб-краулінгу;

в) інтеграції з API SEO-платформ:

1) Ahrefs API;
2) Semrush API;
3) Serpstat API;
4) Google Search Console API;
5) Google Analytics Data API;
6) PageSpeed Insights API.

Ці інтеграції дозволяють значно розширити набір зібраних сигналів: беклінки, органічний трафік, ключові слова, видимість, Core Web Vitals, CTR тощо.

а) моделі штучного інтелекту:

1) OpenAI [4] GPT-моделі / LLaMA / Mistral – генерація рекомендацій, пояснень, оптимізація контенту;
2) Scikit-learn, TensorFlow, PyTorch – власні моделі для кластеризації, предиктивної аналітики;
3) LangChain / LlamaIndex – для роботи з великими наборами даних і створення AI-агентів;

б) бази даних:

1) PostgreSQL – структуровані дані про сайти, метрики, звіти;
2) MongoDB – неструктуровані дані (HTML, JSON-звіти, сирі логі);
3) Elasticsearch – для повнотекстового пошуку;

в) Frontend

1) React / Vue – для побудови інтерактивних панелей;
2) Chart.js / Apache ECharts – для візуалізації метрик;
3) TailwindCSS – для UI/UX.

Для ефективної роботи над проєктом застосовується такий стек:

- GitHub / GitLab – система керування версіями;
- Docker – контейнеризація сервісів для легкого розгортання;
- CI/CD (GitHub Actions) – автоматизація тестування та деплою;
- Postman / Swagger – тестування та документування API;
- Jupyter Notebook – тестування та розробка алгоритмів ШІ;
- Grafana + Prometheus – моніторинг продуктивності системи.

Вибір технологій базується на таких критеріях:

- масштабованість: microservices + Docker дозволяють масштабувати окремі частини;
- гнучкість: Python та модульна архітектура дають можливість швидко змінювати алгоритми ШІ;
- продуктивність: використання асинхронних бібліотек і Scrapy забезпечує швидкий краулінг;
- інтеграційність: REST API спрощує роботу з SEO-сервісами;
- простота подальшої підтримки: популярні та стабільні технології гарантують легкість масштабування.

2.3 Опис процесу розробки

Процес розробки системи автоматизованого SEO-аудиту складається з низки послідовних етапів, кожен з яких забезпечує формування повноцінного функціонального продукту – від аналізу вимог до тестування та розгортання. Нижче подано детальний опис ключових етапів, методів, інструментів і принципів, що формують основу розробки.

Першим етапом є визначення функціональних і нефункціональних вимог до системи.

На основі попередніх цілей і сучасних практик SEO-аудиту формуються ключові задачі:

- виконання технічного SEO-аналізу (стан індексації, мета-теги, robots.txt, sitemap, canonical, HTTP-коди, внутрішні посилання);
- аналіз структури контенту та його відповідність SEO-гайдлайнам;
- збір швидкісних метрик (Core Web Vitals);
- інтеграція із зовнішніми SEO-сервісами;
- обробка великих обсягів HTML-даних;
- автоматичне виявлення SEO-порушень і генерація рекомендацій;
- побудова AI-модулю для інтелектуального аналізу.

Результатом цього етапу є повне технічне завдання (ТЗ) і опис архітектури.

Архітектура системи будується за принципами модульності та сервісної роздільності.

Система включає:

а) модуль збору даних (Crawler Engine):

1) реалізує:
   - краулінг сторінок сайту;
   - обробку динамічного DOM;
   - збір HTML, метаданих, структурних елементів;
   - перевірку внутрішніх посилань;
   - оцінку технічних параметрів (респонс-коди, canonical тощо);
2) використані технології:
   - Python (Scrapy, Requests, aiohttp) – для масштабованого і асинхронного краулінгу;
   - Selenium/Playwright – для JavaScript-сайтів;

б) модуль зовнішніх інтеграцій. Підключення до:

1) Ahrefs API (беклінки, доменні показники);
2) Semrush API (ключові слова, конкуренти);
3) Serpstat API (видимість, аудит помилок);
4) Google Search Console (пошукові запити, індексація);

5) PageSpeed Insights API (Core Web Vitals);

в) аналітичний модуль (AI SEO Analyzer):

1) цей модуль використовує ШІ та алгоритми машинного навчання для:

- класифікації SEO-помилок;
- розпізнавання патернів у внутрішніх посиланнях;
- аналізу контенту;
- визначення релевантності ключових слів;
- виявлення дубльованого контенту;
- генерації текстових рекомендацій за допомогою LLM;

2) використані технології:

- Python: scikit-learn, Pandas, NumPy;
- LLM-інтеграція: GPT, LLaMA, Mistral, LangChain;
- NLP-бібліотеки: spaCy, NLTK;

г) модуль зберігання даних. Архітектура зберігання передбачає:

1) структуровані SEO-метрики – PostgreSQL;

2) HTML та сирі логи краулінгу – MongoDB;

3) великі лог-файли, історія змін – Elasticsearch;

д) модуль генерації звітів відповідає за:

1) формування структурованих PDF/HTML-звітів;

2) візуалізацію графіків;

3) підсумкові AI-рекомендації;

е) Веб-інтерфейс розробляється на основі:

1) React / Vue;

2) TailwindCSS;

3) Chart.js;

4) REST API з бекенду.

Один із найскладніших етапів – створення краулера, який має відповідати вимогам до SEO-аудиту.

Основні функції краулера:

– сканування сайту згідно з правилами robots.txt – коректне і етичне парсування;
– глибина краулінгу – обмеження рівнів вкладеності.

Також до функцій краулера можна віднести -виявлення технічних SEO-проблем, таких як:

– 404, 500, 301-редиректи;
– некоректні canonical-теги;
– дубльовані заголовки;
– відсутність alt-текстів;
– відсутність schema.org розмітки;
– биті посилання.

Аналіз структури сторінки: Title, Description, H1-H6, OpenGraph, структуру контенту, індексацію.

Технічна реалізація:

– Scrapy як основний фреймворк краулінгу;
– асинхронні запити через aiohttp;
– парсинг HTML: BeautifulSoup, lxml;
– аналіз швидкості рендеру: інтеграція PageSpeed API.

AI-компонент працює на трьох рівнях:

а) рівень ML-аналітики виконує:

1) кластеризацію сторінок за типом контенту (K-means);
2) виявлення аномалій у структурі сайту;
3) класифікацію помилок за ступенем критичності;

б) NLP-аналіз контенту функції:

1) оцінка унікальності контенту;
2) перевірка семантичної відповідності ключам;
3) визначення тональності;
4) побудова TF-IDF профілю сторінок;
5) виявлення дубльованих або «тонких» сторінок (thin content);

в) LLM-рівень (генеративний ШІ) використовується для:

1) автоматичного формування рекомендацій у людському стилі;
2) пояснення знайдених проблем;
3) класифікації SEO-помилок по ISO-подібній структурі;
4) генерації оптимізованих Title, Description, H1;
5) створення підсумкових висновків у звіті.

Backend реалізується через FastAPI, що забезпечує:

– високий performance;
– зручну документацію Swagger;
– асинхронну обробку запитів;
– можливість масштабування.

API надає ендпоінти для:

– запуску аудиту сайту;
– перегляду результатів;
– отримання детальних метрик;
– завантаження PDF-звіту;
– аналітики за період.

Frontend побудований як SPA-додаток:

– React + TailwindCSS;
– інтерактивні графіки (ECharts/Chart.js);
– фільтрація та сортування SEO-проблем;
– інтерфейс перегляду рекомендацій;
– індикатори критичності (Low/Medium/High).

Проводяться такі типи тестування:

– юніт-тести Python/JS для ключових модулів;
– інтеграційні тести роботи API;
– навантажувальні тести краулера;
– перевірка коректності роботи AI-рекомендацій;
– E2E-тести (UI → API → AI-модуль → звіт).

Система розгортається на:

- Docker-контейнерах;
- оркестрації через Docker Compose/ Kubernetes (опціонально);
- CI/CD через GitHub Actions.

Узагальнений workflow системи:

- користувач вводить домен → API;
- система запускає краулінг;
- дані потрапляють у базу;
- AI-модуль проводить аналіз;
- генерується підсумковий звіт;
- користувач отримує резюме + рекомендації.

2.4 Висновки до розділу 2

Розробка інтелектуальної системи SEO-аудиту потребує комплексного підходу, що включає ретельне планування, визначення вимог, вибір технологічного стеку та побудову модульної архітектури. Система має складатися з краулера, аналітичних модулів, інтеграцій з SEO-сервісами, AI-компонентів та зручного інтерфейсу. Використання сучасних технологій, інструментів машинного навчання й сервісно-орієнтованої архітектури забезпечує масштабованість, точність аналізу та автоматизацію ключових процесів. Узгоджене поєднання цих елементів дозволяє створити ефективний, гнучкий і стійкий інструмент для повного SEO-аудиту сайтів.

3 РЕАЛІЗАЦІЯ ПРОТОТИПУ

3.1 Загальна структура проєкту

Для повної системи SEO-аудиту (НЕ прототипу) передбачену структуру, зображену нижче. Це потрібно для розуміння того, чим відрізняється прототип від повноцінного продукту.

```
seo-audit-ai/
├── README.md
├── LICENSE
├── .gitignore
├── .env
├── docs/
│   ├── architecture.md
│   ├── api.md
│   └── runbook.md
├── infra/
│   ├── docker-compose.yml
│   ├── k8s/            # (опціонально) manifests для Kubernetes
│   └── terraform/      # (опціонально) інфраструктура в хмарі
├── backend/
│   ├── app/
│   │   ├── main.py
│   │   ├── api/
│   │   │   ├── routes.py
│   │   │   └── schemas.py
│   │   ├── services/
│   │   │   ├── crawler.py
│   │   │   ├── serp_client.py
│   │   │   └── integrations.py
│   │   ├── ai/
│   │   │   ├── model_runner.py
│   │   │   ├── embeddings.py
│   │   │   └── prompt_templates.py
│   │   ├── db/
│   │   │   ├── models.py
│   │   │   └── repository.py
│   │   └── utils/
│   │       └── logging.py
│   ├── requirements.txt
│   └── Dockerfile
```

```
├── ai/
│   ├── notebooks/      # експерименти, підготовка датасетів
│   ├── datasets/       # сирі / оброблені датасети
│   ├── training/       # скрипти тренування моделей
│   ├── models/         # збережені/серіалізовані моделі
│   └── eval/            # скрипти валідації та метрики
├── frontend/
│   ├── src/
│   │   ├── components/
│   │   ├── pages/
│   │   └── api/
│   ├── public/
│   ├── package.json
│   └── Dockerfile
├── data/
│   ├── raw/
│   └── processed/
├── scripts/
│   ├── seed_db.py
│   ├── maintenance/
│   └── run_local.sh
├── tests/
│   ├── unit/
│   ├── integration/
│   └── e2e/
├── ci/
│   └── github-actions.yml
└── tools/
    ├── exporters/     # excel/csv/pdf генератори
    └── monitoring/
```

Короткий опис ключових елементів:

- README.md – швидка інструкція: як запустити локально (Docker), як запустити окремі сервіси, посилання на документацію;
- .env – перелік обов’язкових змінних оточення (API-ключі: OPENAI_KEY, SERPAPI_KEY, AHREFS_TOKEN, URLS для БД тощо);
- docs/ – архітектура, API-контракти, runbook з інструкціями для розгортання та відновлення;
- infra/docker-compose.yml – локальний стек (backend, frontend, postgres, mongo, elasticsearch, redis);

- backend/app/ – FastAPI-сервіс із роутами (запуск аудиту, статус), сервісами збору даних (crawler, serp), ai-модулем (виклик embedding/LLM), repository для БД;
- ai/ – експериментальний простір: ноутбуки, датасети, тренувальні скрипти. Сюди заливаєш датасети і звідси – артефакти моделей (pickle, torch, saved_model);
- frontend/src/ – SPA (React/Vue) панель з дашбордом, переглядом результатів аудиту, детальними звітами та кнопкою запуску нового аудиту;
- data/ – зберігання прикладів/результатів локально (не для production);
- scripts/ – утиліти: заповнення тестової БД, міграції, cron-скрипти;
- tests/ – юніт/інтеграція/E2E тести (pytest, Playwright/Cypress);
- ci/ – базові workflow для перевірки PR: linter, тест, build images;
- tools/exporters/ – модуль для формування PDF/Excel/CSV звітів.

3.2 Створення прототипу

Загальна схема проєкту містить більше функціоналу, ніж потрібно, тому для прототипу створено простішу версію, яка містить лише ключовий функціонал. Струкрура прототипу:

```
seo-audit-ai-prototype/
├── backend/
│   └── app/
│       ├── main.py
│       ├── api/
│       │   └── routes.py
│       ├── services/
│       │   └── crawler.py
│       └── ai/
│           └── model_runner.py
├── .env
└── requirements.txt
```

- main.py (Додаток А) – FastAPI сервер, точка входу, маршрути, health-check;
- routes.py (Додаток Б) – Обробка HTTP-запитів /api/audit, виклик сервісів та AI-модуля;
- crawler.py (Додаток В) – Збір SEO-параметрів з веб-сторінки (title, meta, H1, H2, текст);
- model_runner.py (Додаток Г) – Використання GPT для оцінки контенту та формування SEO-рекомендацій;
- .env (Додаток Д) – містить ключ API;
- requirement.txt (Додаток Д) – містить бібліотеки для встановлення.

Прототип відрізняється від запланованої системи тим, що виконує лише функції, які стосуються On-Page оптимізації, тобто працює з контентним наповненням вебсайту. Також спрощено функціонал компоненту ШІ аналізу – початкова ідея базується на навчанні нейромережі на даних існуючих вебсайтів з гарними показниками, які стосуються SEO, і потім порівнювати їх з даними, які збере краулер, завдяки цьому аналіз контенту стане точнішим, бо рекомендації, отримані на виході будуть не просто оптимізовані під рекомендації від Google, але й під ознаки, які не є очевидними, але які добре сприймаються пошуковими ботами Google, і тому шлях до підвищення позицій в пошуковій видачі стане простішим. Для прототипу же нам буде достатньо API-запиту через OpenAI GPT-4 з промптом:

```
Ти експерт з SEO. Оціни цю веб-сторінку за критеріями:
        – title
        – meta description
        – H1
        – довжина та якість контенту
    title: "{title}"
    meta_description: "{meta_description}"
    h1: {h1}
    text: "{text[:2000]}"
```

```
Вкажи:
1. score (0-4)
2. issues (список проблем)
Формат відповіді JSON:
{{"score": int, "issues": ["text"]}}
```

Такий варіант обрано тому, що моделі OpenAI натреновані і натреновуються на тисячах даних вебсторінок, включно з необхідними нам параметрами, які впливають на SEO. Це аналогічний підхід тому, який має бути використано в повноцінній системі автоматизованого аудиту, але є певний недолік – моделі OpenAI тренуються на всіх вебсторінках, не лише на гарно-оптимізованих, тому в рекомендаціях ми не побачимо особливих ознак з якими потрібно працювати. Відповідно на виході будуть лише загальні рекомендації, які тісно пов’язані з гайдлайнами Google без урахування особливостей вебсайтів в конкретній пошуковій видачі за конкретними пошуковими запитами. Звісно що при реалізації повної системи такий підхід виключено, бо мета створення системи не лише автоматизувати процеси SEO-аудиту, але і зробити аудит більш якісним у майбутньому. Ще варто зазначіти, що узагальнений ворфклоу системи відповідно до повноцінного продукту зберігається, але без пункту з використанням бази даних.

Потік даних та схема зв’язків (рис. 3.1), яку створено в Draw.io [9]:

- клієнт → main.py. Клієнт надсилає POST-запит на /api/audit із URL сторінки;
- main.py → routes.py. Main.py делегує запит маршруту в routes.py;
- routes.py → crawler.py. Викликається функція audit_url(url). Повертає словник: title, meta_description, h1, h2, text, status_code;
- routes.py → model_runner.py. Викликається analyze_content(...) з отриманими параметрами. GPT аналізує контент і повертає: {"score": int, "issues": ["текст проблем"]};
- routes.py → main.py → клієнт. Об’єднується результат crawler + AI і повертається JSON у відповіді.

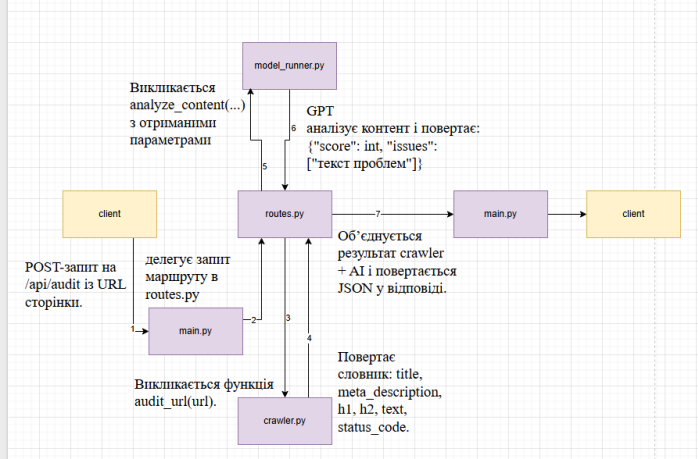


Рисунок 3.1 – Блок-схема зв'язків між модулями прототипу

3.3 Тестування прототипу

Додано ключ API, який було попередньо створено до файлу .env. Слід зауважити, що оскільки OpenAI не надає безкоштовний доступ до API-запитів, то відповідно для тестування було поповнено рахунок акаунту OpenAI на 5$.

Встановлено залежності командою

```
pip install -r requirements.txt
```

Запущено сервер FastAPI командою (рис. 3.2)

```
uvicorn backend.app.main:app --reload --port 8000
```

```
C:\Users\stanislav.levit\seo-audit-ai-prototype>uvicorn backend.app.main:app --reload --port 8000
INFO:     Will watch for changes in these directories: ['C:\\Users\\stanislav.levit\\seo-audit-ai-prototype']
INFO:     Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO:     Started reloader process [9744] using StatReload
INFO:     Started server process [23340]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
```

Рисунок 3.2 – Запущений сервер FastAPI

Перевірено "health" ендпоінт (рис. 3.3) командою

```
curl http://localhost:8000/health
```

Статус код 200 означає успішну відповідь на запит від сервера.

```
PS C:\Users\stanislav.levit> curl http://localhost:8000/health


StatusCode        : 200
StatusDescription : OK
Content           : {"status":"ok"}
RawContent        : HTTP/1.1 200 OK
                    Content-Length: 15
```

Рисунок 3.3 - Перевірка точки виходу “health”

Відправлено цільовий запит до OpenAI командою

```
curl -X POST "http://localhost:8000/api/audit" -H "Content-Type: application/json" -d "{\"url\":\"https://www.site.com/\"}" -o output.json
```

Замість https://www.site.com/ в команді потрібно вказати адресу сторінки, яку потрібно аналізувати. Результат запиту зберігається в файл output.json. Команду виконано, маємо такі дані в файлі output.json (рис. 3.4).

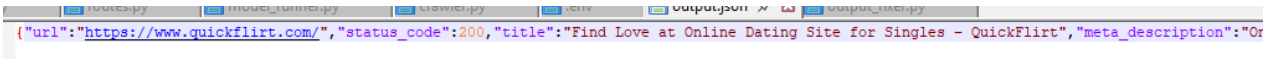


Рисунок 3.4 - Приклад даних з output.json

Дані, отримані в файлі output.json важко читати, тому зроблено скрипт output_fixer.py (Додаток Е) на мові Python. Оновлено схему зв’язків між модулями (рис. 3.5). Новий скрипт оброблює дані з output.json і на їх основі створює читабельний HTML-звіт (рис. 3.6).

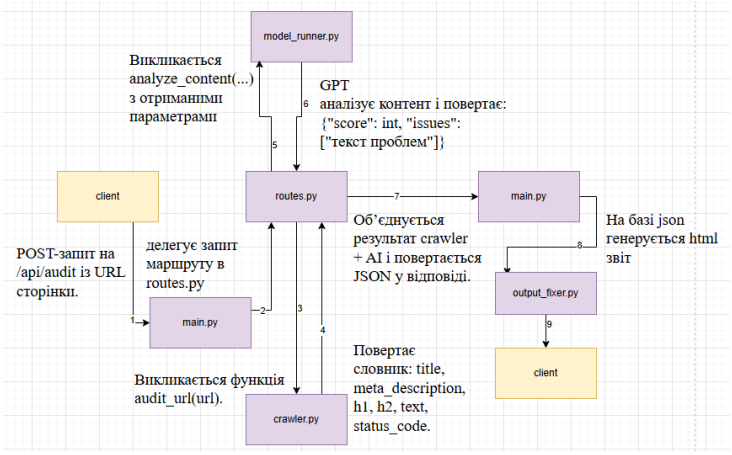


Рисунок 3.5 – Оновлена блок-схема зв’язків між модулями



Рисунок 3.6 - Готовий HTML-звіт

3.4 Аналіз відповідей

Завдяки API-запиту ми отримали аналіз декількох SEO-показників: Meta Title, Meta Description, H1, та контенту. Маємо такі відповіді:

- title міг би бути конкретнішим та містити більше релевантних ключових слів;
- description добре написаний, але міг би містити більше релевантних ключових слів;
- тег H1 є релевантним та добре написаним;
- контент досить довгий та повторюваний, він міг би бути більш лаконічним та захопливим;
- у контенті можна використовувати більше релевантних ключових слів та внутрішніх посилань;
- сторінка містить багато нерелевантної інформації та могла б бути краще структурована.

Висновки, які можна зробити з наданих рекомендацій:

Потрібно додати в title, ще як мінімум одне релевантне ключове слово. Але зробити це потрібно так, щоб зберегти зміст, та не вийти за межі

рекомендованої для title довжини(В англійській мові 40-60 символів максимально допустима доважина, яка вважається оптимальною).

В description потрібно зробити те саме що і в title (для description в англійській мові оптимальна довжина 140-160 символів).

Контент дійсно є повторюваним, але не довгим (до 400 слів), і так, він не дуже захопливий, але він є лаконічним та корисним, і я погоджуюсь з тим, що можна додати внутрішнє посилання в текст. Але не на внутрішню сторінку, а на зовнішню сторінку якого-небудь авторитетного ресурсу, на якій є релевантне дослідження або стаття за тематикою вебсайта, який ми проаналізували. Це може додати авторитетності тексту, який ми проаналізували, відповідно шансів стати вище в пошуковій видачі стане більше. Я також не погоджуюся з тим, що в текст потрібно додати більше ключів тому, що в тексті і так ~10% ключів, якщо включати в список не лише комерційні ключі, але і брендові, а це доволі не мало, зазвичай використовують значення ~7%.

Що стосується нерелевантної інформації - я погоджуюся, бо краулер міг захопити частини коду сторінки, які в цілому не несуть інформативний характер, але вважаю що цей пункт поки що можна ігнорувати.

3.5 Можливі покращення прототипу

Після аналізу отриманих рекомендацій можна зробити висновок, що прототип працює, але є функціонал, який можна доробити перед тим, як переходити до створення повноцінного продукту. Тож для наступного прототипу потрібно внести такі корективи:

- краулер – мінімізувати збір нерелевантної інформації, додати функцію рахування символів в мета-даних (Title, Description, H1), додати збір мета-тегу canonical, бо наразі він його не збирає, додати збір внутрішніх посилань, додати вибір із тексту ключових фраз

релевантних ніші, додати функцію для рахування кількості слів основного контенту, та вирахування відсотка ключих слів порівняно з загальною довжиною тексту;

– ШІ модуль – покращити промпт для більш конкретних відповідей з ремоендаціями щодо потрібних дій та кроків до їх реалізації, додати інформацію, яка покращить знання ніші перед аналізом.

Після реалізації наступного прототипу можливо буде перейти до відмови в користуванні API-запитами до OpenAI, підготувати датасети з даними сторінок, які добре оптимізовані та гарно ранжуються в пошуковій видачі, натренувати власну модель на цих датасетах та отримувати більш досконалий аналіз контенту. Далі потрібно буде додати до краулера можливості збору даних про зовнішні посилання як на цільовому ресурсі, так і на конкурентних ресурсах за певними пошуковими запитами і на порівнянні цих даних розробляти стратегію просування через посилання (Linkbuilding). Наступним кроком може бути створення універсального інтерфейсу, в якому ми зможемо ввести потрібну адресу цільового ресурсу (можливо адреси конкурентів), по кнопці запустити аналіз та отримати відповіді на головне питання – як потрапити в топ пошукової видачі Google за потрібними запитами.

3.6 Оцінка результату

Після аналізу тестової вебсторінки прототипом системи SEO-аудиту отримано та проаналізовано ряд рекомендацій щодо пошукової оптимізації цієї сторінки. Відповідно на вебсторінку внесено наступний ряд правок:

– додано одну тематичну ключову фразу в title;

– додано одну тематичну фразу в description;

– обрано статтю на зовнішньому авторитетному ресурсі для інтеграції посилання на нього в тексті тестової сторінки. Основними

критеріями вибори статті є відомість ресурсу на якому вона розміщена (Наприклад вебсайт Wikipedia), спільна тематика з тестовою сторінкою та наявність статистики;

– дописано три речення для можливості органічної інтеграції внутрішнього посилання на авторитетний ресурс. Новосторені речення представляють собою відкриття для читача факту пов’язаного зі статистикою в ніші до якої відноситься тестова сторінка, і про який читач вірогідно не знає. Таким чином текст тестової сторінки став більш інформативним, захопливим, і в ньому з’явилося середовище для інтеграції посилання на обраний авторитетний ресурс, перейшовши за яким читач зможе дізнатись про повне дослідження пов’язане зі статистикою згаданою в тексті тестової сторінки. Метою такої інтгерації є спонукання пошукового бота до формування зв’язку між сторінкою авторитетного ресурсу та тестовую сторінкою, яка на на цей ресурс посилається і додати авторитетності статті на тестовій сторінці. Відповідно загальна якість контенту сторінки з точки зору пошукового бота має стати вище і відповідно мають почати зростати позиції в пошуковій видачі за релевантними пошуковими запитами;
– інтегровано посилання на обраний зовнішній ресурс в текст тестової статті.

Наступним кроком розпочато моніторинг стану вебсторінки в Google Search Console (рис. 3.7).

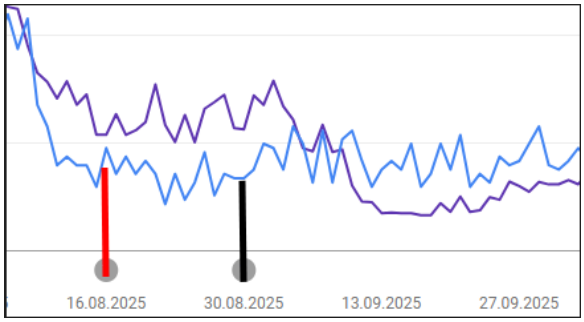


Рисунок 3.7 – Моніторинг стану тестової сторінки в Google Search Console

На скріншоті, який зображено вище є чотири кольорових лінії:

- синя – графік кліків (показує скільки людей перейшло на сторінку через пошук в Google, кожна точка це один день);
- фіолетова – графік показів (показує скільки разів сторінку було показано в пошуковій видачі за різними пошуковими запитами);
- червона – позначає дату внесення правок на вебсторінку (16.08.2025);
- чорна – позначає дату, коли сторінку було повторно проіндексовано пошуковим ботом Google.

Можна побачити, що загальна динаміка на момент внесення правок є негативною, тобто позиції сторінки в пошуковій видачі погіршуються і відповідно зменшується потік органічного трафіку. З 22.08.2025 по 02.09.2025 загальна динаміка стала позитивною. Динаміка кліків залишалась позитивною до 04.09.2025, потім стабілізувалась. Динаміка показів з 02.09.2025 до 18.09.2025 стала негативною. Ці дані свідчать про те, що під час внесення правок тестова сторінка почала відновлювати свої попередньо-втрачені позиції в пошуковій видачі, і правки вебсторінки на базі рекомендацій отриманих завдяки прототипу системи SEO-аудиту допомогли не лише відновлювати позиції впродовж недовгого періоду, але й допомогли зберегти та стабілізувати потік органічного трафіку на сторінку під час не анонсованого оновлення пошукових алгоритмів Google від початку вересня 2025. Звісно динаміка показів свідчить, що починаючи з вересня втрачено багато запитів, за якими сторінку могло бути показано сторінку в пошуку, і тому оновлення пошукових алгоритмів значно вплинуло на стан сторінки з точки зору загальної видимості в пошуковій видачі.

Сумарно можна зробити висновок, що результат є позитивним, оскільки під час оновлення пошукових алгоритмів не почалась негативна динаміка по клікам (негативна динаміка по клікам і показам спостерігалась на багатьох вебсайтах, з якими працює моя команда). Однак слід зауважити, що такий результат став можливим не лише завдяки рекомендаціям,

отриманим завдяки ШІ в прототипі системи SEO-аудиту, а і завдяки аналізу цих рекомендацій мною, як спеціалістом з досвідом в сфері пошукової оптимізації.

ВИСНОВКИ

На момент завершення дослідження проблема автоматизації SEO-аудиту вебсайтів є актуальною. Існує список інструментів, який допомагає спеціалісту отримати дані про SEO-показники вебсайтів і з яких спеціалістом буде сформовано план пошукової оптимізації. Так само, як і на початку дослідження, процес традиційного SEO-аудиту займає багато часу і якість такого аудиту залежить напряму від досвіду спеціаліста. В процесі дослідження проаналізовано інструменти та підходи для проведення SEO-аудиту; досліджено можливості використання штучного інтелеку для автоматизації SEO-процесів; розроблено архітектуру системи автоматизованого аудиту та реалізовано програмний прототип, який аналізує основні SEO-параметри вебсторінок. Тестування прототипу показало, що рекомендації надані штучним інтелектом є прийнятними, але вони підлягають перевірці з боку спеціаліста. Прототип дозволив скоротити час витрачений на один з ключових аспектів пошукової оптимізації – On-Page SEO. Отримані результати показали позитивну реакцію від пошукових ботів, що свідчить про те, що SEO-аналіз виявився ефективним при скорочені витраченого на нього часу, і тому доведено, що використання ШІ для автоматизації SEO-аудиту є доречним і також ефективним. Це означає, що початкової мети було досягнуто маючи лише прототип.

В перспективі повна версія системи SEO-аудиту зможе прибрати необхідність перевірки власних рекомендацій відносно просування в пошукових системах і майже повністю прибрати участь спеціаліста завдяки інтеграції елементів пов’язаних з машинним навчанням і вже існуючих інструментів, які нададуть дані, завдяки яким машинне навчання можливо буде повноцінно реалізувати. Але в цьому дослідженні такий варіант з усім передбаченим функціоналом не було реалізовано через завеликі потенційні грошові витрати.

ПЕРЕЛІК ПОСИЛАНЬ

1. Google Search Console. URL : https://search.google.com/search-console/about (дата звернення : 20.10.2025).

2. Higher rankings. Quality traffic. Measurable results. URL : https://moz.com/ (дата звернення : 19.05.2025).

3. Make your business discoverable – In search, AI, and beyond. URL : https://ahrefs.com/ (дата звернення : 15.05.2025).

4. OpenAI Platform. URL : https://platform.openai.com/docs/overview (дата звернення : 20.06.2025).

5. Python 3.14.1 documentation. URL : https://docs.python.org/3/ (дата звернення : 10.06.2025).

6. Resources for Developers, by Developers. URL : https://developer.mozilla.org/en-US/ (дата звернення : 21.06.2025).

7. Screaming Frog SEO Spider. URL : https://www.screamingfrog.co.uk/seo-spider/#spider-features (дата звернення : 17.05.2025).

8. Search Engine Optimization (SEO) Starter Guide. URL : https://developers.google.com/search/docs/fundamentals/seo-starter-guide (дата звернення : 17.05.2025).

9. Security-first diagramming for teams. URL : https://www.drawio.com/ (дата звернення : 21.06.2025).

10. Speed up search marketing goals achievement. URL : https://serpstat.com/uk/ (дата звернення : 18.05.2025).

11. Start learning about Google Analytics. URL : https://developers.google.com/analytics (дата звернення : 21.05.2025).

12. Win digital brand visibility, everywhere it matters. Semrush. URL : https://www.semrush.com/ (дата звернення : 16.05.2025).

ДОДАТОК А

Програмний код на мові Python у файлі main.py

```
from fastapi import FastAPI
from fastapi.middleware.cors import CORSMiddleware
from backend.app.api.routes import router as api_router

# Створюємо основний застосунок FastAPI з назвою
app = FastAPI(title="SEO Audit AI Prototype")

# Додаємо middleware для CORS, щоб дозволити запити з будь-яких
доменів
app.add_middleware(
    CORSMiddleware,
    allow_origins=["*"],   # Дозволити запити з усіх джерел
    allow_methods=["*"],     # Дозволити всі HTTP-методи (GET,
POST, PUT тощо)
    allow_headers=["*"]    # Дозволити всі заголовки
)

# Підключаємо основні маршрути API за префіксом /api
app.include_router(api_router, prefix="/api")

# Маршрут для перевірки стану сервера
@app.get("/health")
def health():
    return {"status": "ok"}  # Повертає JSON зі статусом
```

ДОДАТОК Б

Програмний код на мові Python у файлі routes.py

```
from fastapi import APIRouter
from pydantic import BaseModel, HttpUrl
from backend.app.services.crawler import audit_url
from backend.app.ai.model_runner import analyze_content

# Створюємо роутер для групування API-маршрутів
router = APIRouter()

# Модель запиту: приймає лише один параметр – валідний URL
class AuditRequest(BaseModel):
    url: HttpUrl

# Маршрут для аудиту сторінки
@router.post("/audit")
def audit(req: AuditRequest):
    # Виконуємо технічний аудит URL (статус-код, теги, контент тощо)
    result = audit_url(req.url)
    # Запускаємо аналіз контенту за допомогою ШІ-модуля
    ai_result = analyze_content(
        title=result.get("title", ""), # Передаємо title
        meta_description=result.get("meta_description", ""),   # Meta description
        h1=result.get("h1", []),       # Список H1
        text=result.get("text", "")    # Текст сторінки
    )
    # Об’єднуємо результати crawлера та ШІ-аналізу і повертаємо одним JSON
    return {**result, **ai_result}
```

ДОДАТОК В

Програмний код на мові Python у файлі crawler.py

```
import requests
from bs4 import BeautifulSoup
from typing import Dict
import re

# Функція для аудиту URL: повертає технічні та контентні дані сторінки
def audit_url(url: str) -> Dict:
    result = {"url": url}
    try:
        # Виконуємо HTTP-запит до сторінки
        r = requests.get(url, timeout=10, headers={"User-Agent": "seo-audit-bot/1.0"})
        result["status_code"] = r.status_code

        # Парсимо HTML-сторінку
        soup = BeautifulSoup(r.text, "html.parser")

        # Отримуємо title, якщо він є
        result["title"] = soup.title.string if soup.title else ""

        # Отримуємо meta description
        result["meta_description"] = (soup.find("meta", attrs={"name": "description"}) or {}).get("content", "")

        # Збираємо всі H1 та H2 заголовки
```

```
        result["h1"]   =   [h.get_text(strip=True)   for   h   in
soup.find_all("h1")]
        result["h2"]   =   [h.get_text(strip=True)   for   h   in
soup.find_all("h2")]

        # Видаляємо скрипти, стилі і noscript, щоб не заважали у
текстовій витяжці
        for tag in soup(["script", "style", "noscript"]):
            tag.decompose()

        # Працюємо лише з <body>
        body = soup.body
        if body:
            # Отримуємо видимий текст із body
            text = body.get_text(separator="\n", strip=True)

            # Прибираємо зайві порожні рядки
            text = re.sub(r"\n\s*\n", "\n", text)

            result["text"] = text
        else:
            # Якщо body немає – текст порожній
            result["text"] = ""

    except Exception as e:
        # Фіксуємо помилку, якщо щось пішло не так
        result["error"] = str(e)

    return result
```

ДОДАТОК Г

Програмний код на мові Python у файлі model_runner.py

```
from dotenv import load_dotenv
import os
import openai
import json
import re

# Завантажуємо змінні середовища з файлу .env
load_dotenv()

# Підтягування API-ключа OpenAI з environment variables
openai.api_key = os.getenv("OPENAI_KEY")

def analyze_content(title: str, meta_description: str, h1: list,
text: str) -> dict:
    """
    Аналізує сторінку за SEO-критеріями:
    - title
    - meta description
    - H1
    - довжина та якість тексту
    Повертає словник формату:
    {"score": int, "issues": [str]}
    """

    # Формуємо промпт для моделі (те, що вона повинна
проаналізувати)
    prompt = f"""
Ти експерт з SEO. Оціни цю веб-сторінку за критеріями:
```

```
- title
- meta description
- H1
- довжина та якість контенту

title: "{title}"
meta_description: "{meta_description}"
h1: {h1}
text: "{text[:3000]}"

Вкажи:
1. score (0-4)
2. issues (список проблем)

Формат відповіді JSON:
{{"score": int, "issues": ["text"]}}
"""

    try:
        # Виклик моделі OpenAI (GPT-4) з переданим промптом
        response = openai.ChatCompletion.create(
            model="gpt-4",
            messages=[{"role": "user", "content": prompt}],
            temperature=0     # Температура 0 → максимальна
точність та передбачуваність
        )

        # Отримуємо контент відповіді
        content = response.choices[0].message.content

        # Шукаємо JSON у відповіді через регулярний вираз
        match = re.search(r'\{.*\}', content, re.DOTALL)
        if match:
            # Якщо знайдено – парсимо JSON
            return json.loads(match.group())
        else:
```

```
            # Якщо JSON не знайдено
            return {"score": 0, "issues": ["AI did not return
valid JSON"]}

    except Exception as e:
        # Якщо виклик OpenAI закінчився помилкою
        return {"score": 0, "issues": [f"AI error: {str(e)}"]}
```

ДОДАТОК Д

Приклад оголошення змінної для збереження ключа API у файлі .env та перелік бібліотек для встановлення у файлі requirements.txt

Файл .env:

```
OPENAI_KEY=sk-proj-ваш_ключ_API
```

Файл requirements.txt:

```
fastapi
uvicorn
requests
beautifulsoup4
pydantic
openai
```

ДОДАТОК Е

Програмний код на мові Python у файлі output_fixer.py

```
import json

# Читаємо JSON-вивід з файлу output.json
with open("output.json", "r", encoding="utf-8") as f:
    data = json.load(f)

# Функція визначає колір для score (оцінки)
def score_color(score):
    if score <= 1:
        return "red"      # Поганий результат
    elif score <= 3:
        return "orange"   # Середній результат
    else:
        return "green"    # Хороший результат

# Формуємо HTML-контент з даних аудиту
html_content = f"""
<!DOCTYPE html>
<html lang="uk">
<head>
<meta charset="UTF-8">
<title>SEO Аналіз: {data['url']}</title>
<style>
body {{ font-family: Arial, sans-serif; margin: 20px; }}
h1 {{ color: #333; }}
h2 {{ color: #555; }}
.score {{ font-weight: bold; color:
{score_color(data['score'])}; }}
```

```
.issues {{ background: #f9f9f9; padding: 10px; border-radius: 5px; }}
.issues li {{ margin-bottom: 5px; }}
.section {{ margin-bottom: 20px; }}
</style>
</head>
<body>

<h1>SEO Аналіз сторінки</h1>

<div class="section">
<strong>URL:</strong> <a href="{data['url']}" target="_blank">{data['url']}</a><br>
<strong>Статус код:</strong> {data.get('status_code', 'N/A')}<br>
<strong>Score:</strong> <span class="score">{data['score']}</span>
</div>

<div class="section">
<h2>Title</h2>
<p>{data['title']}</p>
</div>

<div class="section">
<h2>Meta Description</h2>
<p>{data['meta_description']}</p>
</div>

<div class="section">
<h2>H1</h2>
<ul>
{"".join(f"<li>{h}</li>" for h in data.get('h1', []))}
</ul>
</div>
```

```
<div class="section">
<h2>H2</h2>
<ul>
{"".join(f"<li>{h}</li>" for h in data.get('h2', []))}
</ul>
</div>

<div class="section">
<h2>Проблеми (issues)</h2>
<ul class="issues">
{"".join(f"<li>{issue}</li>"  for  issue  in  data.get('issues',
[]))}
</ul>
</div>

</body>
</html>
"""

# Зберігаємо згенерований HTML у файл
with open("seo_report.html", "w", encoding="utf-8") as f:
    f.write(html_content)

print("Готово! Відкрий seo_report.html у браузері для перегляду
звіту.")
```