

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра інженерії програмного забезпечення

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри _____ Є. О. Давиденко

підпис

«__» _____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

**Дослідження SEO-аналітики вебсайтів за допомогою
методів штучного інтелекту**

Спеціальність «Інженерія програмного забезпечення»

121 – КРМ.1 – 608м.21810806

Студент

_____ В. С. Вальковський

підпис

«__» _____ 2024 р.

Керівник канд. техн. наук, доцент

_____ Г. В. Горбань

підпис

«__» _____ 2024 р.

Миколаїв – 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра інженерії програмного забезпечення

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Є.О. Давиденко

«_____» _____ 2024 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи магістра

Видано студенту групи 608 факультету комп'ютерних наук

(прізвище, ім'я, по батькові студента)

1. Тема кваліфікаційної роботи «Дослідження SEO-аналітики вебсайтів за допомогою методів штучного інтелекту»

Затверджена наказом по ЧНУ від «10» _____ листопад 2024 р. № _____

2. Строк представлення кваліфікаційної роботи «_____» _____ 2024 р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні

1. Вітчизняна та зарубіжна науково-технічна документація з питань застосування методів штучного інтелекту в системах аналізу веб-ресурсів.

2. Документація із суміжних технологій.

3. Статистичні дані про популярні сайти та їх основні характеристики (тематика, кількість відвідувань, позиції в пошукових системах тощо).

4. Перелік питань, що підлягають розробці

1. Теоретичні аспекти проведення SEO-аналітики веб-сайтів та можливості використання методів штучного інтелекту.

2. Функціональні та нефункціональні вимоги до системи SEO-аналітики на основі штучного інтелекту.

3. Розробка функціональної моделі системи: бізнес-процеси, інтерфейси, сценарії взаємодії.

4. Архітектура програмного забезпечення системи з урахуванням ШІ.

5. Алгоритми та методи застосування машинного навчання в системі.

5. Перелік графічних матеріалів

Діаграми UML, зображення сторінок веб-інтерфейсу системи, блок-схеми алгоритмів, компоненти системи та схематичне зображення механізму їх взаємодії.

6. Завдання до спеціальної частини

1. Кодування та тестування програмного забезпечення.
2. Експериментальна апробація програмного забезпечення.
3. Аналіз результатів використання програмного забезпечення.

Керівник роботи _____ канд. техн. наук, доцент Горбань Г.В.

(посада, прізвище, ім'я, по батькові)

(підпис)

Завдання прийнято до виконання

_____ Вальковський В'ячеслав Сергійович

(прізвище, ім'я, по батькові студента)

(підпис)

Дата видачі завдання «_____» _____ 2024 р.

КАЛЕНДАРНИЙ ПЛАН виконання кваліфікаційної роботи

Тема: Дослідження SEO-аналітики вебсайтів за допомогою методів штучного інтелекту _____

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Розробка та затвердження завдання на виконання КРМ	08.10.2023	10.11.2023	виконано
2.	Огляд літератури за темою роботи	20.11.2023	05.12.2023	виконано
3.	Складання календарного плану КРМ	06.12.2023	07.12.2023	виконано
4.	Аналіз предметної області	09.12.2023	14.12.2023	виконано
5.	Розробка проєктних рішень	05.01.2024	07.01.2024	виконано
6.	Моделювання та конструювання ПЗ	09.01.2024	14.01.2024	виконано
7.	Кодування, тестування та апробація розробленого ПЗ, аналіз результатів тестування, розробка керівництва користувача	15.01.2024	09.02.2024	виконано
8.	Відгук керівника КРМ	13.02.2024	13.02.2024	виконано
9.	Оформлення КРМ та презентації	20.01.2024	19.02.2024	виконано
10.	Попередній захист	08.02.2024	15.02.2024	виконано
11.	Рецензування			виконано
12.	Завершення оформлення КРМ та презентації	10.02.2024	19.02.2024	виконано
13.	Захист кваліфікаційної роботи	26.02.2024	26.06.2024	виконано

Розробив студент Вальковський В'ячеслав Сергійович _____
(прізвище, ім'я, по батькові) (підпис)
«__» _____ 2024 р.

Керівник роботи канд. техн. наук, доцент Горбань Г. В. _____
(посада, прізвище, ім'я, по батькові) (підпис)
«__» _____ 2024 р.

АНОТАЦІЯ

до кваліфікаційної роботи магістра
«Дослідження SEO-аналітики вебсайтів за допомогою методів штучного
інтелекту»

Студент 608м гр.: Вальковський В'ячеслав Сергійович

Керівник: канд. техн. наук, доцент Горбань Г. В.

Основними результатами роботи є теоретичне узагальнення процесів проведення SEO-аналітики веб-сайтів та можливостей застосування методів штучного інтелекту, запропонована архітектура програмного забезпечення системи з урахуванням сучасних алгоритмів штучного інтелекту, створене програмне забезпечення системи SEO-аналітики на основі штучного інтелекту, проведене тестування та оцінка якості розробленої системи, а також висновки щодо теоретичної та практичної значущості результатів.

Об'єкт дослідження: процеси SEO-аналітики вебсайтів.

Предмет дослідження: система SEO-аналітики вебсайтів на основі методів штучного інтелекту.

Мета дослідження: підвищення ефективності та вдосконалення процесу аналізу факторів пошукової оптимізації вебресурсів за рахунок розробки системи SEO-аналітики вебсайтів за допомогою методів штучного інтелекту.

Для досягнення визначеної мети необхідно вирішити такі **завдання**:

- дослідити теоретичні основи проведення SEO-аналітики вебсайтів та можливості застосування методів штучного інтелекту;
- розробити функціональну та інформаційну моделі системи SEO-аналітики;
- запропонувати архітектуру програмного забезпечення системи;
- створити ПЗ, провести його апробацію та оцінити якість;
- сформулювати висновки щодо теоретичних положень та практичної цінності розробки.

Сторінок – 74. Рисунків – 37. Таблиць – 1. Посилань – 42. Додатків – 2.

Ключові слова: SEO-аналітика, штучний інтелект, машинне навчання, веб-сайт, оптимізація, пошукова система, програмне забезпечення.

ABSTRACT

to the master's qualification work
"Research of SEO-analytics of websites using artificial intelligence methods"
Student 608m gr.: Valkovskyi Viacheslav Serhiiovych
Supervisor: Candidate of Technical Sciences, Associate Professor Horban G. V.

The main results of the work are a theoretical generalization of the processes of SEO-analytics of websites and the possibilities of applying artificial intelligence methods, the proposed architecture of the system software taking into account modern artificial intelligence algorithms, the created software of the SEO-analytics system based on artificial intelligence, testing and quality assessment of the developed system, as well as conclusions on the theoretical and practical significance of the results.

Object of research: SEO analytics processes of websites.

Subject of research: website SEO analytics system based on artificial intelligence methods.

Purpose of the study: Increasing the efficiency and improving the process of analyzing the factors of search engine optimization of web resources by developing a system of SEO analytics of websites using artificial intelligence methods.

To achieve this goal, it is necessary to solve the following **tasks**:

- to study the theoretical foundations of SEO website analytics and the possibilities of applying artificial intelligence methods;
- to develop functional and information models of the SEO analytics system;
- to propose the architecture of the system software;
- to create the software, test it and evaluate its quality;
- to formulate conclusions about theoretical provisions and practical value of the development.

Pages - 74. Figures - 37. Tables - 1. References - 42. Appendices - 2.

Keywords: SEO analytics, artificial intelligence, machine learning, website, optimization, search engine, software.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ	3
ВСТУП.....	4
1 ТЕОРЕТИЧНІ ОСНОВИ SEO-АНАЛІТИКИ ВЕБСАЙТІВ ТА ШТУЧНОГО ІНТЕЛЕКТУ	6
1.1 Поняття SEO-аналітики вебсайтів та її роль в підвищенні ефективності пошукової оптимізації	6
1.2 Основні поняття штучного інтелекту і його застосування в SEO-аналітиці .	10
1.3 Огляд сучасного стану проведення SEO-аналітики	15
1.4 Специфікації вимог до системи SEO-аналітики вебсайтів	20
Висновки до розділу 1.....	24
2 МОДЕЛЮВАННЯ ОБ'ЄКТУ ТА ПРЕДМЕТУ РОБОТИ. ФУНКЦІОНАЛЬНІ ТА ІНФОРМАЦІЙНІ МОДЕЛІ	25
2.1 Функціональна модель системи SEO-аналітики на основі методів ШІ	25
2.2 Інформаційна модель бази даних системи.....	29
Висновки до розділу 2.....	33
3 АРХІТЕКТУРА, МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	34
3.1. Вибір інструментів для розробки системи SEO-аналітики вебсайтів	34
3.2. Розробка архітектури програмного забезпечення.....	39
3.3. Діаграмне моделювання системи SEO-аналітики вебсайтів.....	42
3.4. Аналіз сучасних програм для проведення SEO-аналітики	46
Висновки до розділу 3.....	53
4 РОЗРОБКА ТА АПРОБАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ SEO-АНАЛІТИКИ	54
4.1. Кодування та тестування програмного забезпечення	54
4.2. Експериментальна апробація програмного забезпечення.....	60
4.3 Аналіз результатів використання програмного забезпечення.....	65
Висновки до розділу 4.....	70
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	71
ВИСНОВКИ	74
ДОДАТКИ.....	75

ПЕРЕЛІК СКОРОЧЕНЬ

CMS (Content Management System) - система управління контентом.

CSS (Cascading Style Sheets) - каскадні таблиці стилів.

CTR (Click-Through Rate) - відсоток клацань.

HTML (Hypertext Markup Language) - мова розмітки гіпертексту.

PPC (Pay-Per-Click) - платний пошуковий трафік.

ROI (Return on Investment) - рентабельність інвестицій.

SEM (Search Engine Marketing) - маркетинг у пошукових системах.

SEO (Search Engine Optimization) - оптимізація для пошукових систем.

SERM (Search Engine Reputation Management) - управління репутацією в пошукових системах.

SERP (Search Engine Results Page) - сторінка результатів пошуку.

SSL (Secure Sockets Layer) - протокол захищеного з'єднання.

UI (User Interface) - інтерфейс користувача.

URL (Uniform Resource Locator) - єдиний локатор ресурсу.

UX (User Experience) - користувацький досвід.

XML (Extensible Markup Language) - розширювана мова розмітки.

ВСТУП

Актуальність. Пошукова оптимізація є пріоритетним напрямком просування вебресурсів та залучення аудиторії, при цьому SEO-аналітика є ефективним інструментом для її здійснення. Застосування методів штучного інтелекту розширює можливості автоматизованого аналізу та оптимізації вебсайтів.

В Україні відсутня розроблена система SEO-аналітики на базі штучного інтелекту, що є предметом даного дослідження. Результати роботи здатні бути використані фахівцями з пошукової оптимізації та маркетингу.

Крім того, розробка матиме теоретичне значення для подальшого вивчення застосування методів штучного інтелекту в аналізі інформаційних ресурсів. Отже, обрана тема є актуальною та сприятиме подальшому розвитку досліджень і практики в цій галузі.

Об'єкт дослідження. процеси SEO-аналітики вебсайтів.

Предмет дослідження: система SEO-аналітики вебсайтів на основі методів штучного інтелекту.

Мета дослідження: підвищення ефективності та вдосконалення процесу аналізу факторів пошукової оптимізації вебресурсів за рахунок розробки системи SEO-аналітики вебсайтів за допомогою методів штучного інтелекту.

Для досягнення визначеної мети необхідно вирішити такі **завдання**:

- дослідити теоретичні основи проведення SEO-аналітики вебсайтів та можливості застосування методів штучного інтелекту;
- розробити функціональну та інформаційну моделі системи SEO-аналітики;
- запропонувати архітектуру програмного забезпечення системи;
- створити програмне забезпечення, провести його апробацію та оцінити якість;
- сформулювати висновки щодо теоретичних положень та практичної цінності розробки.

Обґрунтування необхідності нової розробки. Існує нагальна потреба у розробці вітчизняної системи SEO-аналітики вебсайтів на основі штучного інтелекту, оскільки на сьогодні в Україні такої системи немає.

Аналіз науково-технічної літератури свідчить про наявність недоліків у застосуванні перспективних підходів штучного інтелекту для завдань SEO-аналітики.

Провідні світові компанії та науковці активно розвивають дослідження в цій галузі, однак український ринок потребує власної розробки.

Враховуючи світові тенденції інтеграції штучного інтелекту в ІТ-системи, існує необхідність створення вітчизняної системи SEO-аналітики, що удосконалисть підходи застосування методів штучного інтелекту.

Основні напрямки досліджень:

- дослідження та узагальнення теоретичних основ проведення SEO-аналітики вебсайтів та застосування методів штучного інтелекту;
- розробка функціональної та інформаційної моделей системи SEO-аналітики на основі штучного інтелекту;
- запропонування архітектури програмного забезпечення системи з урахуванням сучасних алгоритмів машинного навчання;
- створення прототипу програмного забезпечення системи та його емпірична апробація на репрезентативних даних;
- оцінка якості та ефективності розробленої системи шляхом порівняння з аналогами.

Сферою застосування результатів дослідження буде оптимізація та просування вебсайтів, розробка ефективних онлайн-маркетингових стратегій, створення власного програмного забезпечення для SEO-аналітики, подальший розвиток теорії та методів застосування штучного інтелекту в аналізі інформаційних ресурсів, а також використання отриманих знань в освітньому процесі з метою вдосконалення підготовки фахівців у галузі програмування, маркетингу та інформаційних технологій.

1 ТЕОРЕТИЧНІ ОСНОВИ SEO-АНАЛІТИКИ ВЕБСАЙТІВ ТА ШТУЧНОГО ІНТЕЛЕКТУ

1.1 Поняття SEO-аналітики вебсайтів та її роль в підвищенні ефективності пошукової оптимізації

SEO-аналітика вебсайтів – це процес збору, аналізу та інтерпретації даних про вебсайт з метою вдосконалення його пошукової оптимізації. Ця аналітика дозволяє виявити слабкі місця в оптимізації сайту та визначити стратегії для підвищення його ефективності у пошукових системах [1].

SEO-аналітика вебсайту включає в себе ряд основних завдань, спрямованих на вимірювання, аналіз та вдосконалення різних аспектів оптимізації для покращення видимості та рейтингу сайту у пошукових системах. В цьому контексті важливою задачею є збір даних і надання цінної інформації для удосконалення SEO-стратегії вебсайту.

Першим завданням SEO-аналітики є вимірювання трафіку та відвідуваності сайту. Це є аналіз кількості відвідувачів, джерел їх переходу на сайт та способів взаємодії з контентом. Ці дані допомагають зрозуміти, як ефективно привертати аудиторію на сайт, які джерела трафіку є найбільш результативними і які аспекти контенту привертають найбільше уваги користувачів [2].

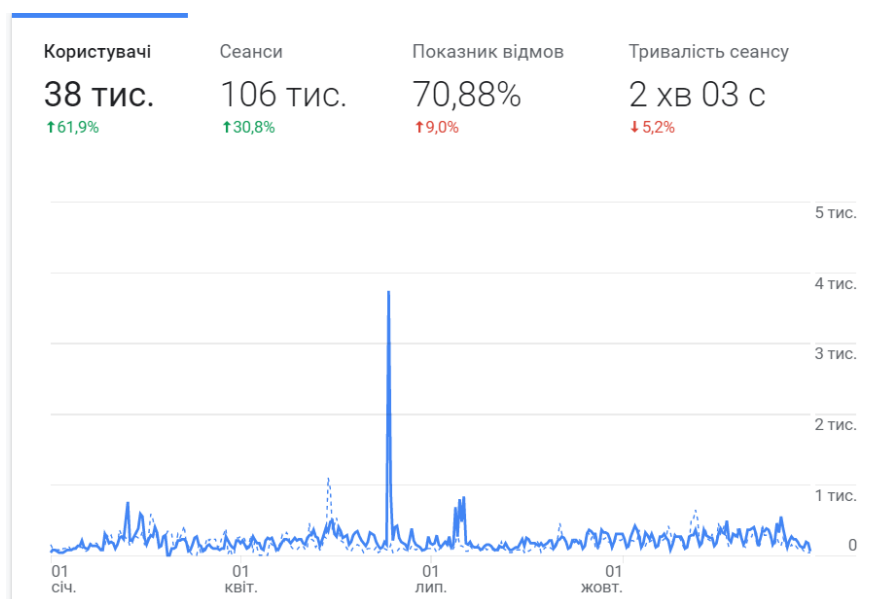


Рисунок 1.1 – Вимірювання трафіку та відвідуваності сайту

Друге завдання полягає в аналізі ключових слів і позицій в пошукових системах. SEO-аналітика дозволяє відстежувати розташування ключових слів в пошукових системах і виявляти можливості для покращення. Це допомагає визначити, які ключові слова є найбільш ефективними для привертання трафіку і покращення рейтингу сайту [3].

<input type="checkbox"/> Ключове слово	↓ Сер. к-сть запитів на місяць	Змінення за три місяці	Змінення порівняно з попереднім роком	Конкуренція
Указані вами ключові слова				
<input type="checkbox"/> спортивне харчування	1 тис. –10 тис.	0%	0%	Висока
Варіанти ключових слів				
<input type="checkbox"/> купити спортивне харчування	100–1 тис.	0%	0%	Висока
<input type="checkbox"/> спортивне харчування для набору маси	100–1 тис.	0%	+900%	Висока
<input type="checkbox"/> магазин спортивного харчування	100–1 тис.	0%	0%	Висока

Рисунок 1.2 – Аналіз ключових слів

Третє завдання SEO-аналітики – виявлення технічних проблем, які можуть впливати на видимість сайту у пошукових системах. Це пов’язано з виявленням проблем зі швидкістю завантаження сторінок, неправильною індексацією сторінок або недоступністю для пошукових роботів. Аналіз цих проблем дозволяє виправити їх і покращити показники SEO [4].

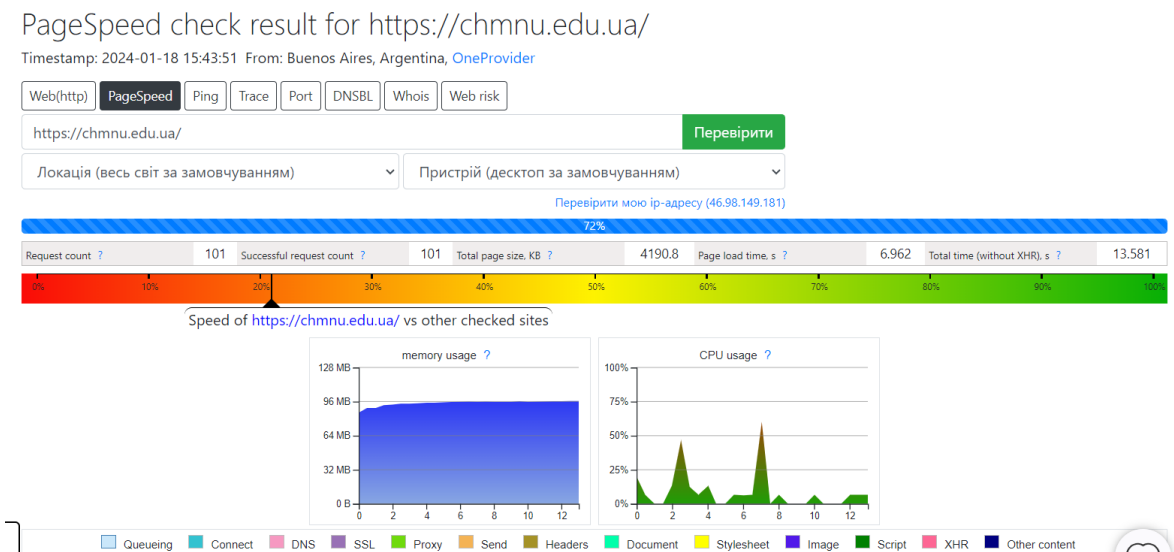


Рисунок 1.3 – Перевірка швидкості вебсайту

Четверте завдання – аналіз зовнішніх посилань. SEO-аналітика допомагає відстежувати, які сайти посилаються на ваш сайт і які посилання є найбільш цінними для обраної SEO-стратегії. Це дозволяє розуміти, як залучати більше якісних зовнішніх посилань [5].

Внутрішні посилання	Зовнішні посилання
Посилань 166, індексуються 152	Посилань 15 індексуються 15
Чорноморський національний університет імені Петра Могили https://chmnu.edu.ua/	Український журнал медицини, біології та спорту https://jmbms.com.ua/
Про університет https://chmnu.edu.ua/category/pro-universitet	Інституційний репозитарій https://dspace.chmnu.edu.ua/jspui/
Привітання ректора https://chmnu.edu.ua/privitannya-rektora/	Moodle 3.9 Чорноморського національного університету імені Петра Могили http://moodle3.chmnu.edu.ua/
Звіт ректора https://chmnu.edu.ua/zvit-rektora/	Кваліфікаційні роботи студентів Чорноморського національного університету імені Петра Могили http://krs.chmnu.edu.ua/
Ректорат https://chmnu.edu.ua/rektorat/	Інституційний репозитарій http://dspace.chmnu.edu.ua/
Вчена рада https://chmnu.edu.ua/vchena-rada-chornomorsko-natsionalnogo-universitetu-imeni-petra-mogili/	Старий сайт університету http://old.chmnu.edu.ua/
Структура https://chmnu.edu.ua/struktura-chnu/	Petro Mohyla TV https://www.youtube.com/channel/UCvuqftGH3g13JMniNZ9GImQ
Антикорупційна діяльність https://chmnu.edu.ua/zarobigannya-koruptsiyi/	MOODLE 3.9 http://moodle3.chmnu.edu.ua/

Рисунок 1.4 – Аналіз зовнішніх посилань

Останнє завдання полягає в вимірюванні конверсій. Конверсія – це досягнення бажаної дії відвідувачем вебсайту або рекламною кампанією, такої як покупка, заповнення форми або підписка. SEO-аналітика допомагає відстежувати, які джерела трафіку приносять найбільше конверсій і які сторінки мають найвищу конверсійну ефективність. Це дозволяє визначити, які канали привертання трафіку є найбільш прибутковими і як удосконалити сторінки для забезпечення високої конверсійної ефективності [6].

Отже, основні завдання SEO-аналітики вебсайту включають вимірювання трафіку та відвідуваності сайту, аналіз ключових слів та позицій у пошукових системах, виявлення технічних проблем, аналіз зовнішніх посилань та вимірювання конверсій. Ці завдання допомагають зрозуміти ефективність SEO-стратегії, виявити потенційні проблеми та можливості для покращення, а також визначити найефективніші канали привертання трафіку та забезпечити високу конверсійну ефективність.

Ще одним важливим показником для аналізу ефективності сайту в аналітиці є органічний трафік. Він демонструє кількість користувачів, які знайшли ресурс у пошукових системах без використання будь-якої платної реклами. Стеження за цим показником дає змогу оцінити, наскільки успішними є зусилля з оптимізації сайту під пошукові системи та наскільки привабливим є його контент для цільової аудиторії. Регулярне збільшення показника органічного трафіку свідчить про те, що вебресурс належним чином налаштований для індексації пошуковими системами і його вміст є актуальним для користувачів [7].

Пошукові системи, такі як Google, Bing або Yahoo, використовують свої алгоритми для визначення рейтингу вебсайтів у пошукових результатах. Ці алгоритми враховують різні фактори, включаючи релевантність, тобто міру відповідності вебсторінки до запиту користувача, авторитетність та якість контенту, щоб визначити, які сторінки будуть показані на перших позиціях.

PR (PageRank) – це алгоритм, розроблений Google, який використовується для оцінки авторитетності вебсторінок. Він враховує кількість та якість зовнішніх посилань, що спрямовують на сторінку, для визначення її важливості. Чим більше якісних посилань вказують на сторінку, тим вище її PR і, відповідно, шанси на високі позиції у пошукових результатах [8].

Bing використовує алгоритм Bing Ranking Algorithm, який має кілька компонентів для визначення рейтингу сторінок у пошукових результатах. Він оцінює релевантність, а також враховує фактори, такі як авторитетність джерела, якість контенту, користувацький досвід та інші фактори [9].

Yahoo використовує алгоритм, відомий як Yahoo Search Ranking Algorithm, для визначення рейтингу сторінок. Він також оцінює відповідність сторінки до запиту користувача, але деталі алгоритму не розкриваються публічно [10].

Варто зауважити, що точні деталі алгоритмів пошукових систем є комерційною таємницею, і компанії, такі як Google, Bing та Yahoo, не розкривають всіх своїх технічних деталей. Однак загальна ідея полягає у тому, щоб забезпечити користувачам якомога більш відповідні результати, враховуючи різні фактори, які вказують на якість та авторитетність вебсторінок.

1.2 Основні поняття штучного інтелекту і його застосування в SEO-аналітиці

Штучний інтелект (ШІ) є галуззю інформатики, що зосереджена на створенні машин або систем, які здатні виконувати завдання, що зазвичай вимагають людського інтелекту. Системи штучного інтелекту розробляються для навчання на власному досвіді, виявлення закономірностей та прийняття рішень на основі вхідних даних [11].

Перші задуми про можливість створити розумні машини з'явилися ще у 1950-х роках. Група ентузіастів на чолі з вченим Джоном Маккарті почала цікавитися ідеєю створити системи, здатні мислити на рівні людини. Саме тоді було вперше використано термін «штучний інтелект».

У 1956 році в Дартмутському коледжі відбулась історична конференція, де вчені сформулювали основні цілі та завдання для майбутніх досліджень в цій галузі. Протягом 1960-х років було досягнуто певних успіхів, зокрема створено розумну систему під назвою Логіст.

Однак в 1970-ті попередній оптимізм змінився розчаруванням. Очікування від штучного інтелекту виявилися перебільшеними. Наступні роки увійшли в історію як «зима штучного інтелекту».



Рисунок 1.5 – IBM 702: комп'ютер, який використовувало перше покоління дослідників ШІ

Новий етап розвитку розпочався в 1980-ті завдяки інтересу до нейронних мереж та еволюційних обчислень. В 1990-х науковці досягли значних успіхів у застосуванні глибинних нейронних мереж.

І ось нарешті у 2000-ні роки, коли обчислювальна потужність комп'ютерів різко зросла, дослідження в галузі штучного інтелекту переживають сьогодні справжній розквіт [12].

Штучний інтелект може бути класифікований на два основних типи: слабкий і загальний. Слабкий штучний інтелект відноситься до систем, які спеціалізуються на виконанні конкретних завдань і можуть демонструвати розумну поведінку у цих обмежених областях. Такі системи можуть виявляти інтелектуальні здібності, але не мають свідомості чи універсального розуму. Загальний штучний інтелект, натомість, описує системи, здатні до розуміння, навчання та виконання будь-яких завдань, які можуть виконувати люди [13].

Штучний інтелект може аналізувати великі обсяги даних, розпізнавати патерни (шаблони), розв'язувати складні проблеми та навчатися відновлювати свою продуктивність на основі отриманого досвіду.

Основні характеристики штучного інтелекту [14]:

- здатність до самонавчання – здатність системи покращувати свої параметри і досягати цілей без зовнішнього управління;
- адаптивність – здатність системи підлаштовуватися під змінювані умови за допомогою механізмів навчання та самонавчання;
- маневреність – можливість застосовувати отримані раніше знання в нових ситуаціях для досягнення поставленої мети;
- швидкодія – здатність швидко обробляти великі масиви даних і приймати рішення в реальному часі;
- розуміння людської мови – здатність сприймати і розуміти звичайну мову людини, а не лише формальні команди;
- автономність – можливість функціонувати без прямого керування людиною, на основі раніше отриманих знань і досвіду.

Машинне навчання (Machine Learning) – галузь штучного інтелекту, що дає змогу комп'ютерним програмам вчитися на великих наборах даних і покращувати свої результати без явно прописаного коду. Машинне навчання дозволяє системам

автоматично виявляти закономірності та зв'язки в даних для прийняття рішень і виконання завдань, як це роблять люди [15].

Машинне навчання широко застосовується в SEO-аналітиці для:

- побудови класифікаційних моделей для автоматичного розпізнавання тематики вебсторінок.
- прогнозування показників ефективності (наприклад, органічного трафіку) на основі аналізу контенту.
- виявлення ключових слів і тем, що використовуються на конкурентних сайтах.
- групування сторінок сайту для подальшої оптимізації.

Це дозволяє автоматизувати багато рутинних завдань SEO-аналітики і підвищити точність отриманих даних.

Обробка природної мови (Natural Language Processing, NLP) - галузь штучного інтелекту, що займається взаємодією комп'ютерів та людської мови. NLP дозволяє комп'ютерам аналізувати, розуміти й генерувати людську мову [16].

Обробка природної мови дозволяє:

- проводити поглиблений аналіз текстового контенту сайтів;
- визначати тематику, ключові слова й фрази текстів;
- порівнювати семантику різних вебсторінок;
- групувати контент за тематикою;
- аналізувати відгуки і коментарі користувачів тощо.

Це допомагає в ефективній SEO-оптимізації контенту та сайтів.

Кластеризація – це метод об'єднання об'єктів даних у групи (кластери), так, щоб об'єкти всередині однієї групи були максимально схожі між собою, а з іншими групами – максимально відрізнялися [17].

Класифікація – це процес розподілу об'єктів на класи або категорії відповідно до певних ознак. Завдання полягає не просто в групуванні об'єктів, а й в автоматичному призначенні їм класів.

Використання кластеризації та класифікації в SEO-аналітиці дозволяє:

- автоматично групувати сторінки сайту за тематикою;
- класифікувати нові тексти за раніше побудованими моделями;
- визначати тематичні кластери конкурентних ресурсів;
- аналізувати тематичну структуру сайтів та їх оптимізацію.

Це спрощує аналіз великих обсягів контенту та прийняття управлінських рішень.

Генетичний алгоритм – це метод оптимізації, заснований на принципах еволюції та природного добору [18].

Генетичний алгоритм починає роботу з початкової популяції можливих рішень задачі (хромосом). Кожне рішення (особина) оцінюється за допомогою функції придатності.

Генетичні алгоритми можуть бути ефективно використані для оптимізації SEO-параметрів вебсайтів:

- Генерація оптимальних ключових слів. ГА допомагають вибрати найкращі комбінації ключових слів, що максимізують результативність сайту.
- Оптимізація структури URL та внутрішньої зв'язності сайту. ГА беруть до уваги багато факторів для побудови оптимальної структури взаємопосилань.
- Розподіл контенту за тематичними розділами сайту. Застосовуючи ГА, можна отримати оптимальну організацію контенту.
- Генерація текстів для посібників/статей. ГА дозволяють створювати унікальні тексти з врахуванням SEO-параметрів.

Таким чином, генетичні алгоритми допомагають автоматизувати процес оптимізації сайтів, беручи до уваги велику кількість параметрів одночасно.

Переваги використання штучного інтелекту в SEO-аналітиці полягають в автоматизації та підвищенні ефективності процесів аналізу та оптимізації вебсайтів, що дозволяє спрощує роботу фахівців та обробляти великі масиви даних за короткий час. Крім того, алгоритми штучного інтелекту дають змогу оптимізувати процес прийняття управлінських рішень шляхом надання чітких рекомендацій на основі об'єктивного аналізу. Ще однією перевагою є покращення

точності прогнозування результатів SEO-оптимізацій завдяки навчанню моделей машинного навчання на великих масивах даних [19].

Для ефективної роботи методів штучного інтелекту необхідна велика кількість якісних даних. Проте при зборі даних для SEO-аналітики можуть виникати проблеми:

- незначна кількість даних для окремих вебресурсів та їх сторінок;
- неактуальні або неточні дані через постійні зміни на сайтах;
- відсутність певних параметрів на деяких ресурсах (наприклад мета-тегів);
- помилки при зборі даних та їх форматуванні.

Через це моделі штучного інтелекту можуть давати неточні прогнози або мати низьку якість при роботі з обмеженою вибіркою даних.

Отже, штучний інтелект як наукова галузь зосереджена на створенні інтелектуальних систем, здатних виконувати завдання, які зазвичай потребують людського розуму. За останні десятиліття в галузі ШІ відбувся справжній прорив завдяки зростанню обчислювальної потужності комп'ютерів і появі ефективних методів машинного навчання.

Сьогодні системи штучного інтелекту активно застосовуються в різних сферах: комп'ютерний зір, розпізнавання мови, медицина, освіта та багато інших. В SEO-аналітиці методи ШІ, такі як машинне навчання, обробка природної мови та класифікація, дозволяють автоматизувати складні завдання аналізу та оптимізації вебсайтів.

Проте для досягнення наступного рівня розвитку штучного інтелекту необхідно усунути обмеження доступності якісних даних і підвищити ефективність моделей при роботі в умовах обмеженої вибірки. Подальший прогрес у цих напрямках дозволить штучному інтелекту ще більше полегшити працю фахівців у різних галузях та позитивно вплинути на розвиток суспільства.

1.3 Огляд сучасного стану проведення SEO-аналітики

Все більше компаній використовують SEO-аналітику для покращення своєї видимості в Інтернеті та залучення більше цільового трафіку. Завдяки постійному розвитку технологій, аналітичні інструменти стають потужнішими і надають детальну інформацію для прийняття обґрунтованих рішень.

Огляд сучасного стану проведення SEO-аналітики включає такі поняття:

Використання аналітичних інструментів. Сучасна SEO-аналітика використовує різноманітні аналітичні інструменти, такі як Google Analytics, Google Search Console, SEMrush, Ahrefs та інші, для збору даних про вебсайт і його взаємодію з пошуковими системами. Ці інструменти надають важливу інформацію про трафік, ключові слова, показники конверсії, швидкість завантаження сторінок та інші метрики.

Аналіз конкурентів. Дослідження конкурентів є важливою складовою SEO-аналітики. Вона допомагає виявити сильні та слабкі сторони конкурентів, їх стратегії пошукової оптимізації, популярні ключові слова та тактики, які можна застосувати для покращення власного вебсайту.

Моніторинг та відстеження. SEO-аналітика передбачає постійний моніторинг та відстеження показників ефективності вебсайту. Це включає аналіз трафіку, позицій в пошукових системах, індексації сторінок, посилань, поведінки користувачів та інших метрик. Такий моніторинг дозволяє вчасно виявляти проблеми та вживати заходів для поліпшення рейтингу вебсайту.

Мобільна оптимізація. З огляду на зростання використання мобільних пристроїв, SEO-аналітика зосереджена на мобільній оптимізації. Це включає аналіз швидкості завантаження сторінок на мобільних пристроях, адаптивний дизайн, оптимізацію мобільних пошуків. Застосування є важливим етапом у відповідності сучасних трендів SEO-аналітики:

Важливість вмісту. Якісний вміст є ключовим фактором для SEO-аналітики. Він повинен бути унікальним, оригінальним і спеціально орієнтованим на

аудиторію. Аналітика допомагає визначити ефективні ключові слова та теми, що привертають трафік, та аналізує взаємодію користувачів з вмістом.

Технічна оптимізація. SEO-аналітика зосереджується на технічній оптимізації вебсайту. Вона включає аналіз структури сайту, швидкості завантаження сторінок, мобільної сумісності, розподілу внутрішніх посилань та інших факторів, які впливають на взаємодію з пошуковими системами.

Соціальні медіа. SEO-аналітика також охоплює аналіз соціальних медіа, оскільки вони впливають на видимість вебсайту в пошукових системах. Вона допомагає відстежувати соціальні сигнали, взаємодію з аудиторією та ефективність кампаній у соціальних медіа.

Відстеження ROI. Сучасна SEO-аналітика також зосереджена на вимірюванні поверненості інвестицій (ROI). Вона дозволяє відстежувати конверсії, продажі, прибуток і інші метрики, щоб визначити ефективність SEO-кампаній і визначити найбільш прибуткові стратегії.

Сьогодні вивченням методів та практики проведення SEO-аналітики активно займаються численні науковці та фахівці з цієї галузі. Їх ґрунтовні дослідження дозволяють визначити актуальні тренди розвитку, виділити важливі аспекти цієї складної процедури оптимізації вебсайтів. Завдяки науковим статтям і публікаціям можна проаналізувати поточний стан проведення SEO-аналітики, окреслити основні підходи та методології, що застосовуються фахівцями, а також виявити проблемні питання, над розв'язанням яких працюють дослідники. Огляд їхніх робіт надасть уявлення про сучасний рівень розвитку цієї галузі.

Спочатку розглянемо основні положення статті «Тенденції розвитку SEO-оптимізації та подолання існуючих ризиків» [20]:

Автори розпочинають з аналізу сучасних трендів у сфері пошукової оптимізації. Особлива увага приділяється ролі мобільного трафіку. Зазначається, що більшість користувачів сьогодні шукають інформацію саме через мобільні пристрої. Тому SEO-спеціалісти повинні оптимізувати сайти під мобільні телефони і планшети.

Інший тренд – розвиток семантичного пошуку. Алгоритми аналізують не лише текст, а й його зміст. Відповідно, контент має бути не лише унікальним, а й мати ціннісну інформацію для користувачів.

Експерти звертають увагу на поширення соціальних мереж, які стали важливим каналом просування. Сайти необхідно оптимізувати під розміщення та просування контенту у Facebook, Твіттер тощо.

Також аналізуються ризики SEO, на які необхідно звертати увагу: «чорний» лінкбїлдинг, накрутка показників тощо.

Інша стаття «SEO-оптимізація сайту підприємства як інструмент інтернет-маркетингу» [21] присвячена дослідженню застосування SEO-оптимізації сайту як ефективного інструменту інтернет-маркетингу на підприємстві.

Автор розглядає SEO-оптимізацію в контексті сучасних умов ведення бізнесу в мережі, коли конкуренція на ринку Інтернет вкрай висока, а вибір правильної стратегії просування продукту є критично важливим.

У статті наголошується на тому, що саме оптимізація сайту є одним з ключових інструментів отримання цільової аудиторії, адже перші позиції в пошукових системах забезпечують найбільшу кількість відвідувачів.

Автор розглядає основні принципи побудови ефективної SEO-стратегії: від аналізу ринку пошукових систем і різновидів оптимізації до етапів просування та чинників ранжування.

Хоча деякі положення роботи вже не є актуальними, вона залишається корисною базою для розуміння суті SEO-оптимізації та її ролі у сучасному цифровому маркетингу підприємств.

У дослідженні [22] автор Литвиненко А.С. розкриває сутність понять SEO-оптимізації та її важливість для просування сайту в умовах щільної конкуренції в мережі. Також сформульовано мету дослідження – вивчити внутрішні методи оптимізації для отримання високих позицій у пошуку.

Далі зазначається, що SEO постійно розвивається, змінюються алгоритми ранжування. Для 2019 року рекомендується зосередитися на внутрішній оптимізації сайту.

Стаття поділяє SEO на три основних напрямки: зовнішню та внутрішню оптимізацію, а також аналітику.

Детально проаналізовано складові внутрішньої оптимізації: формування семантичного ядра ключових слів на основі аналізу цільових запитів та конкурентів; оптимізація контенту та його структури, підбір ключових слів та їх вживання; налаштування метатегів title та description з урахуванням запитів; робота з файлом robots.txt та картою сайту.

Підкреслюється, що правильна внутрішня оптимізація є основоположним чинником успішного просування сайту в пошукових системах.

На основі аналізу статті можна зробити такі висновки щодо сучасного стану SEO:

- постійний розвиток SEO і зміни алгоритмів ранжування пошукових систем вимагають вивчення нових трендів та методів оптимізації;
- акцент робиться на внутрішні методи оптимізації та якісний, цінісний контент, а не маніпуляції та зовнішні фактори;
- формування семантичного ядра на основі аналізу цільових запитів та конкурентного середовища є одним із ключових етапів;
- важливими є створення логічної структури сайту, оптимізація метатегів, контенту з урахуванням його читабельності для користувачів;
- досягнення високих позицій потребує комплексного підходу, що включає оптимізацію, моніторинг та аналітику результатів;
- SEO має розглядатися як постійний процес, а не одноразовий захід, оскільки цифрове середовище швидко змінюється.

Отже, стаття дає уявлення про актуальні тенденції у сфері SEO та необхідність постійного вдосконалення підходів.

У статті «SEO-оптимізація контенту» [22] розглядається питання оптимізації контенту та її значення для бізнесу в цілому.

Автори статті зазначають, що оптимізація контенту є важливою складовою SEO-просування сайтів та сторінок у пошукових системах. Оптимізований контент

допомагає сайту отримувати більше відвідувачів з пошукових систем шляхом покращення позицій в пошукових запитах користувачів.

Розглядаються такі інструменти просування в Інтернеті як SEO та SMM. Зазначається, що SEO допомагає підвищити позиції сайту в органічному пошуку, а SMM - просування в соціальних мережах.

Дослідження аналізує методи оптимізації контенту: оптимізація заголовків, підзаголовків, тексту, метатегів, посилань та ін. Окремо розглядається оптимізація контенту для мобільних пристроїв.

Автори поділяють оптимізацію контенту на такі дві категорії:

– **Внутрішня оптимізація.** Під цим розуміється комплекс заходів, спрямованих на оптимізацію роботи всередині вебсайту. Сюди входить оптимізація текстового контенту, мета-даних, внутрішніх посилань тощо.

– **Зовнішня оптимізація.** Це комплекс заходів, спрямованих на отримання цільових зовнішніх посилань і підвищення цитованості сайту в Інтернеті.

Отже, сучасний стан SEO-оптимізації відображається у зростаючій важливості якісного та орієнтованого на користувача вмісту, технічної оптимізації вебсайту, дослідженні ключових слів та конкурентів, аналізі даних за допомогою різноманітних аналітичних інструментів, мобільній оптимізації, локальному SEO, соціальних медіа та відстеженні ROI. SEO-оптимізація є невід’ємною частиною ефективної онлайн-стратегії, і компанії зосереджуються на використанні аналітики та передових підходів для підвищення видимості своїх вебсайтів у пошукових системах та залучення цільового трафіку.

Крім вищезазначених аспектів, важливо також слідкувати за науковими дослідженнями в галузі SEO-оптимізації, щоб бути завжди попереду своїх конкурентів. Швидкий розвиток технологій та алгоритмів пошукових систем вимагає постійного оновлення знань і стратегій. Активна участь у конференціях, вебінарах, читання наукових статей та співпраця з експертами у цій сфері дозволяють отримувати нові ідеї, які можна впровадити для покращення позицій в пошукових системах вебресурсу.

1.4 Специфікації вимог до системи SEO-аналітики вебсайтів

ПРИЗНАЧЕННЯ ТА МЕЖІ ПРОЄКТУ

Призначення SEO-аналітики вебсайтів за допомогою методів ШІ

Призначення системи – проведення аналізу та оцінки ефективності SEO оптимізації вебсайтів за допомогою методів ШІ.

Погодження, що ухвалені в програмній документації

Мова програмування – Python.

Використання бібліотек Beautiful Soup для збору даних, NLTK для обробки текстів, Scikit-Learn для побудови машинних моделей.

Реалізація вебінтерфейсу за допомогою Eel.

Межі проєкту ПЗ

Програмне забезпечення розробляється для виконання поставлених у 1.1 цілей, без додаткових функціональних можливостей.

Не передбачається інтеграція з іншими інформаційними системами, крім використання зовнішніх бібліотек, зазначених у 1.2.

ЗАГАЛЬНИЙ ОПИС

Сфера застосування

Система призначена для сфери оптимізації вебсайтів та просування в пошукових системах (SEO-аналітика).

Характеристики користувачів

Користувачами є фахівці з SEO, маркетологи, вебаналітики, які мають:

- базові навички роботи на ПК;
- знання основних понять SEO (пошукові запити, ранжування в пошуку);
- досвід аналізу вебресурсів і пошукових даних.

Загальна структура і склад системи

Система складається з:

- модуля збору даних (Beautiful Soup);
- модуля обробки даних (NLTK);
- модуля машинного навчання (Scikit-Learn);

– Вебінтерфейсу (Eel, HTML/CSS/JS).

ФУНКЦІЇ СИСТЕМИ

Функція збору даних

Опис функції: збір HTML-коду та отримання текстового змісту сторінок обраних вебресурсів.

Вхідна і вихідна інформація: вхід – URL сторінок, вихід – зібраний HTML та тексти сторінок.

Функціональні вимоги: система повинна збирати дані з вказаних доменів; формат вихідних даних – текстовий файл.

Функція попередньої обробки

Опис функції: очищення, токенізація, лематизація текстів, отримання ключових слів.

Вхідна і вихідна інформація: вхід – тексти, вихід – оброблені тексти та ключові слова.

Функціональні вимоги: повинна видаляти зайві символи, робити тексти малими літерами, отримувати ключові слова.

Функція побудови машинних моделей

Опис функції: побудова класифікаторів і регресійних моделей на базі оброблених даних.

Вхідна і вихідна інформація: вхід – тексти, ключові слова, вихід – побудовані моделі.

Функціональні вимоги: повинна містити принаймні два класифікатори та одну регресійну модель.

Функція аналізу даних

Опис функції: аналіз зібраних і оброблених даних за допомогою побудованих моделей.

Вхідна і вихідна інформація: вхід – дані, моделі, вихід – результати аналізу.

Функціональні вимоги: повинна давати узагальнений опис даних, виявляти ключові теми, оцінювати ефективність SEO.

Функція надання результатів

Опис функції: візуалізація результатів аналізу через вебінтерфейс.

Вхідна і вихідна інформація: вхід – результати аналізу, вихід – користувацький інтерфейс.

Функціональні вимоги: забезпечувати зручне відображення результатів.

ВИМОГИ ДО ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

Джерела і зміст вхідної інформації (даних)

Вебсторінки обраних сайтів (HTML, текст).

Нормативно-довідкова інформація (класифікатори, довідники тощо)

Словник ключових слів у форматі .txt.

Класифікатори тематики контенту (моделі .pkl).

Вимоги до способів організації, збереження та ведення інформації

Дані та моделі зберігаються локально на сервері.

Немає обмежень на кількість запитів та користувачів.

ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ

Настільний ПК або ноутбук.

4-х ядерний процесор, 8 Гб RAM, 100 Гб SSD-накопичувач.

Інтернет-канал шириною до 1 Гбіт/с.

ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Архітектура програмної системи

Трирівнева архітектура: доступ, бізнес-логіка, зберігання даних.

Використання модульного підходу.

Системне ПЗ

Операційна система Windows.

Мережне ПЗ

Бібліотека Eel для вебсервера.

ПЗ ведення БД

База даних SQLite3.

Мова та технологія

Мова Python 3.9.

Бібліотека Eel для розробки вебзастосунку.

HTML, CSS, JS, Bootstrap.

ВИМОГИ ДО ЗОВНІШНІХ ІНТЕРФЕЙСІВ

Інтерфейс користувача

Вебінтерфейс.

Апаратний інтерфейс

Мережевий інтерфейс - Wi-Fi, Ethernet.

Інтерфейс введення-виведення – клавіатура, монітор.

Програмний інтерфейс

HTTP-розмежування (REST API).

JSON дані.

Комунікаційний протокол

HTTP/HTTPS.

Сервер - TCP-порт 80 або 443.

Клієнт - HTTP-запити.

ВЛАСТИВОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Доступність

Підтримка усіх основних веббраузерів.

Супроводжуваність

Періодичні оновлення, виправлення помилок та вдосконалення.

Переносимість

Перенос програми з одного комп'ютера на інший за допомогою бібліотек та папки з проектом.

Продуктивність

Швидке виконання основних функцій, в межах до 200 мс на сторінку.

Надійність

Дані легко відновити зі SQLite бази даних.

Безпека

Програма здійснює доступ в Інтернет, проте передавані дані не є конфіденційними і можуть бути відкрито отримані з вебсайту через браузер.

ІНШІ ВИМОГИ

Вимоги до макетування

Зручні елементи управління.

Сучасний дизайн.

Вимоги масштабованості

Підтримка роботи з великими даними.

Подальше нарощування функціоналу.

Висновки до розділу 1

В розділі детально розглянуто теоретичні основи SEO-аналітики вебсайтів та застосування штучного інтелекту в цій галузі.

Проаналізовано основні завдання SEO-аналітики, до яких належать вимірювання та аналіз таких показників як кількість відвідувачів сайту, джерела трафіку, способи взаємодії користувачів з контентом. Це дозволяє зрозуміти ефективні канали привертання аудиторії та аспекти контенту, що привертають увагу. Також розглядаються завдання з аналізу руху ключових слів в пошукових системах, перевірки технічних аспектів на предмет впливу на ранжування, дослідження зовнішніх посилань та вимірювання конверсій.

Розглянуто основні поняття штучного інтелекту. Детально описано застосування машинного навчання, обробки природної мови, кластеризації, класифікації, генетичних алгоритмів в SEO-аналітиці для автоматизації процесів.

Також у розділі оглянуто сучасний стан проведення SEO-аналітики вебсайтів.

Розроблена специфікація вимог до системи SEO-аналітики вебсайтів. У ній чітко визначено призначення та межі майбутнього проєкту, описано сферу застосування системи, характеристики користувачів, загальний функціонал та структуру.

Таким чином, комплексно розглянуто підходи до аналізу та оптимізації вебсайтів та застосування сучасних методів штучного інтелекту.

2 МОДЕЛЮВАННЯ ОБ'ЄКТУ ТА ПРЕДМЕТУ РОБОТИ. ФУНКЦІОНАЛЬНІ ТА ІНФОРМАЦІЙНІ МОДЕЛІ

2.1 Функціональна модель системи SEO-аналітики на основі методів ШІ

Функціональна модель системи SEO-аналітики на основі методів ШІ потрібна для таких цілей:

- модель дозволяє на етапі проєктування чітко уявити, які задачі система буде вирішувати і яким чином;
- модель допомагає ефективно поділити функціонал між компонентами системи та оптимізувати їх взаємодію;
- на основі моделі розробники складають плани створення окремих модулів та їх інтеграції;
- модель є базою для написання сценаріїв тестування функціональності окремих частин системи і їх взаємодії;
- функціональна модель є складовою частиною технічної документації на систему;
- модель допомагає плавно доповнювати систему новими можливостями, уточнюючи її.

Отже модель необхідна для ефективного планування, розробки та тестування складної системи SEO-аналітики.

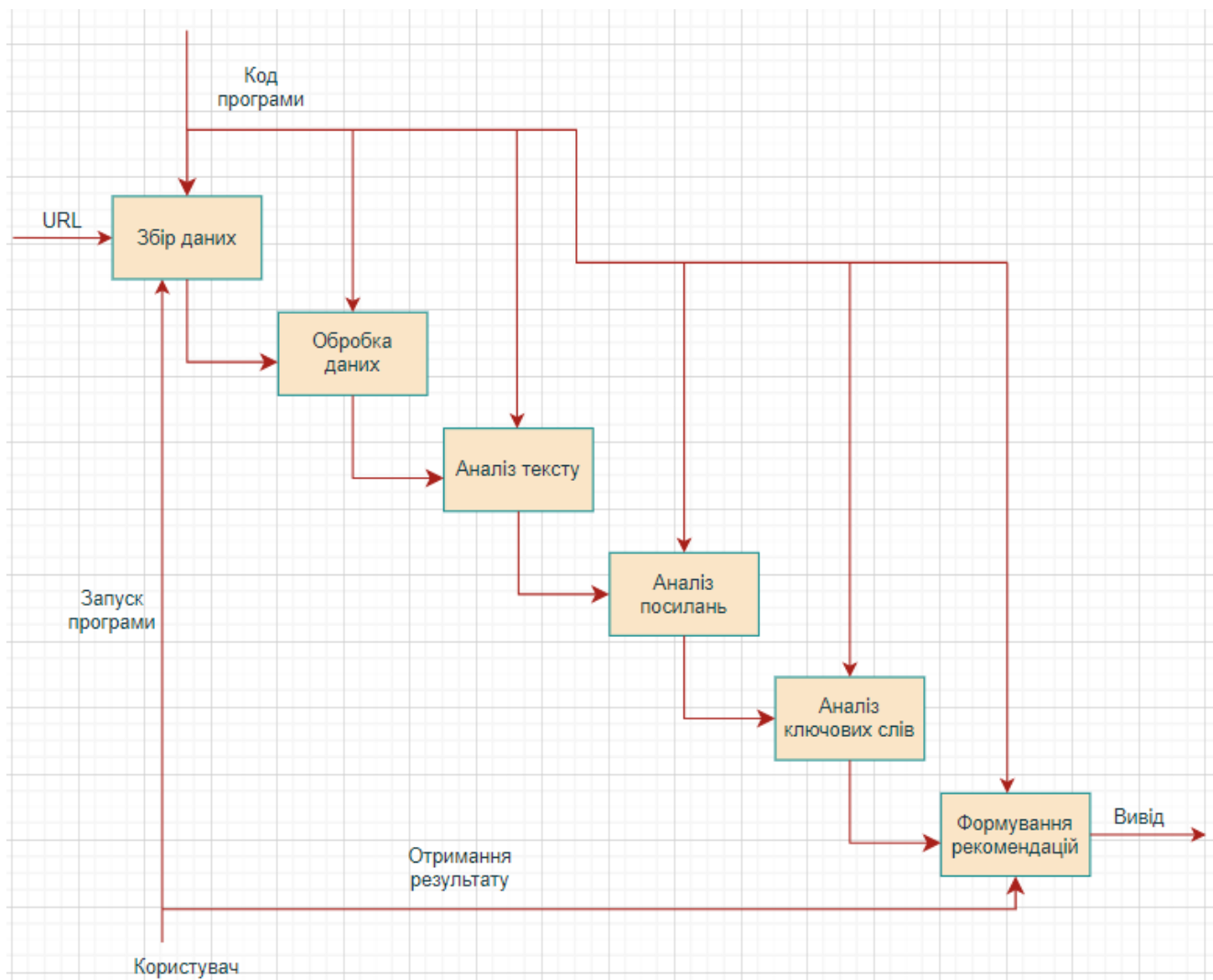


Рисунок 2.1 – Функціональна модель системи SEO-аналітики

Відповідно до рис. 2.1 маємо нижченаведені дані.

Вхідні дані:

- URL сайту, який потрібно проаналізувати;
- модуль збору даних:
 - здійснює збір даних з визначеного сайту;
 - збирає весь текстовий та структурний контент (сторінки, пости, категорії тощо);
 - збирає дані про внутрішні та зовнішні посилання;
 - збирає дані про ключові слова;
 - формує базу даних із зібраної інформації.

- модуль очищення даних:
 - видаляє шумову інформацію (рекламу, скрипти тощо);
 - стандартизує формати даних;
 - передає очищені дані для аналізу;
- модулі аналізу:
 - текстового контенту;
 - внутрішніх посилань;
 - зовнішніх посилань;
 - ключових слів;
- модуль формування рекомендацій:
 - на основі даних аналізу генерує поради щодо удосконалення SEO;
- вивід результатів:
 - рекомендації, що доступні користувачу програми;
- користувач:
 - запускає програмний код та отримує результат аналізу.

Після створення функціональної моделі системи SEO-аналітики, де були чітко специфіковані її компоненти та взаємозв'язки, було отримано уявлення про логіку роботи окремих модулів. Це дозволило сформулювати загальний алгоритм функціонування системи та подати його у вигляді блок-схеми.

Система SEO-аналітики розпочинає свою роботу з отримання URL сайту, який необхідно проаналізувати. Для цього використовується бібліотека Requests.

Наступним кроком є збір необхідних даних з цього сайту. Для парсингу HTML коду та екстракції текстового змісту, метаданих та інших даних використовується бібліотека BeautifulSoup.

Отримана сукупність даних проходить процес очищення та підготовки за допомогою регулярних виразів. Невірні символи, теги та інші «сміття» видаляються.

Далі проводиться аналіз текстового контенту за допомогою методів NLP та ML. Виділяються основні теми, емоційне забарвлення, класифікація текстів тощо.

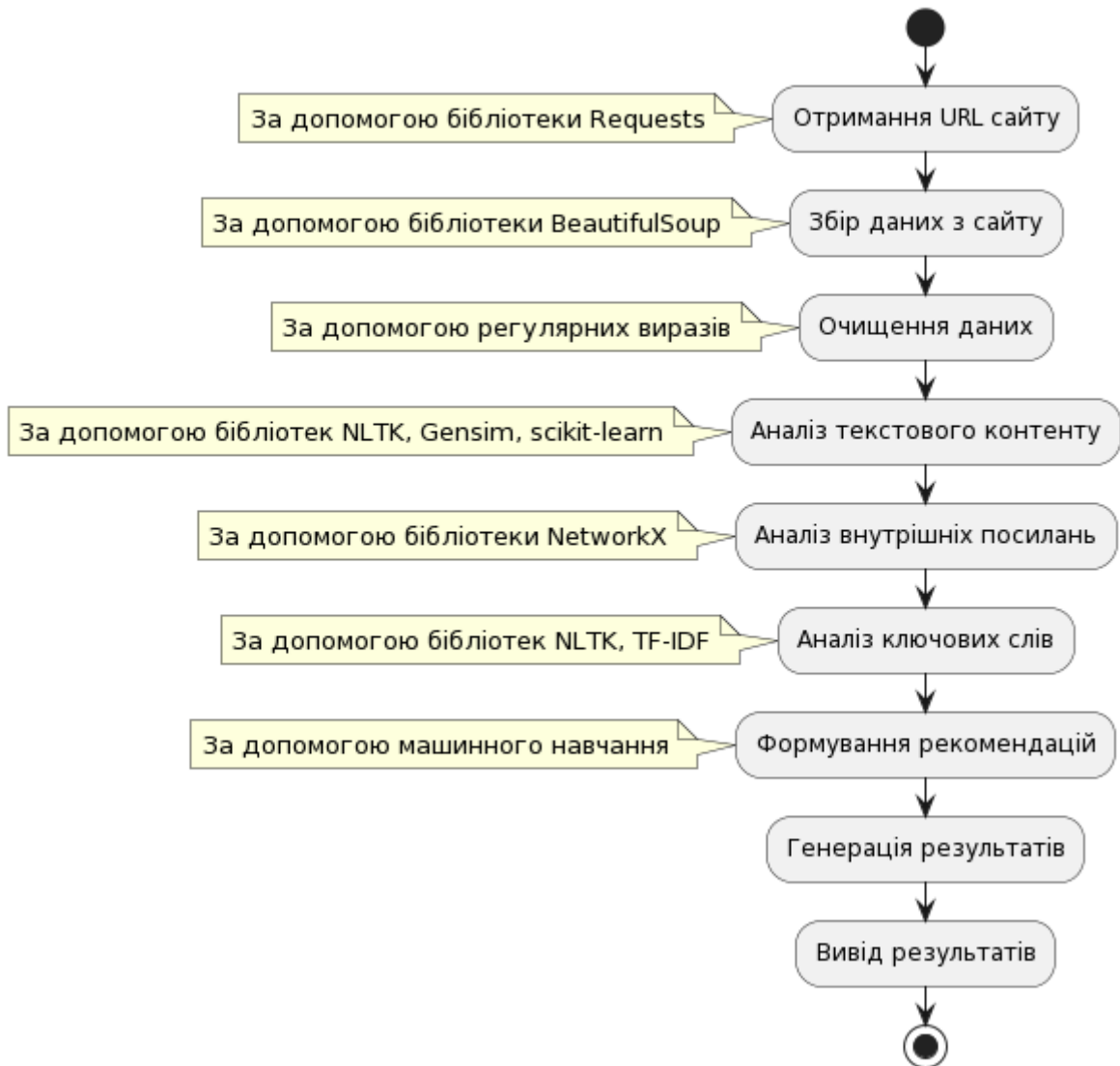
Загальний алгоритм роботи системи SEO-аналітики

Рисунок 2.2 – Загальний алгоритм функціонування систем

Аналізуються також внутрішні посилання на сайті за допомогою бібліотек графів, таких як NetworkX.

На основі отриманих даних формуються рекомендації щодо вдосконалення SEO-оптимізації сайту та його контенту.

Підсумкові результати генеруються та виводяться користувачеві, що завершує роботу системи.

Таким чином, отримано функціональну модель системи SEO-аналітики та детально описано алгоритм її роботи у вигляді блок-схеми. Це дозволить на наступних етапах розробки:

- коректно розподілити функціонал між модулями та компонентами;
- розробити логіку та код окремих модулів відповідно до їх призначення;
- провести інтеграцію модулів та отримати робочий прототип системи;
- перевірити, чи відповідає дійсна поведінка системи запланованому алгоритму;

- уточнити та вдосконалити окремі модулі та логіку роботи системи в цілому.

Таким чином, отримані модель і алгоритм допоможуть ефективно організувати подальшу розробку системи SEO-аналітики.

2.2 Інформаційна модель бази даних системи

Інформаційна модель бази даних – це формальний опис структури даних, які будуть використовуватися системою. Її мета – зрозуміти, яку саме інформацію потрібно зберігати, та як буде організований зв'язок між цими даними [24].

Спочатку в інформаційній моделі описують сутності – основні об'єкти, із якими буде працювати система. Наприклад, у системі аналітики це можуть бути вебсторінки, ключові слова та інше.

Для кожної сутності визначається набір атрибутів – властивостей, значення яких будуть зберігатись. Для сторінки це URL, назва, дата оновлення.

Окреслюються також зв'язки між сутностями – що з чим пов'язане. Наприклад, сторінка містить посилання на інші сторінки.

Така модель дозволяє зрозуміти логіку подання даних і є основою для створення схеми бази даних, яка буде ці дані фактично зберігати. Це корисно як для подальшої розробки системи, так і для спілкування всіх, хто буде з нею працювати.

Для розроблюваної системи SEO-аналізу ефективності сайтів необхідно визначити базові сутності, їхні атрибути та зв'язки між сутностями. Таким чином, буде створена інформаційна модель бази даних, яка відобразить логіку збереження й обробки необхідної інформації.

Для системи визначаємо основні сутності та атрибути. Також визначаємо зв'язки між сутностями, оскільки вони необхідні для подальшого аналізу та оптимізації сайтів.

Сутності та атрибути з типом даних:

- вебсторінка:
 - ID - integer, primary key, autoincrement;
 - URL – varchar(255);
 - Зміст – text;
 - Метадані – text;
 - Дата_аналізу – datetime;
- ключові слова:
 - ID - integer, primary key, autoincrement
 - Слово - varchar(100)
 - Кількість_вживань - integer
 - ID_сторінки - integer, foreign key до таблиці Вебсторінка
- посилання:
 - ID - integer, primary key, autoincrement;
 - ID_вихідної_сторінки - integer, foreign key до Вебсторінка;
 - ID_цільової_сторінки - integer, foreign key до Вебсторінка;
- машинна модель:
 - ID - integer, primary key, autoincrement;
 - Тип - varchar(50);
 - Параметри – text;
- результат моделі:
 - ID - integer, primary key, autoincrement
 - ID_моделі - integer, foreign key до Машинна модель
 - ID_об'єкта - integer, foreign key до Вебсторінка або Ключові слова
 - Результат – text.

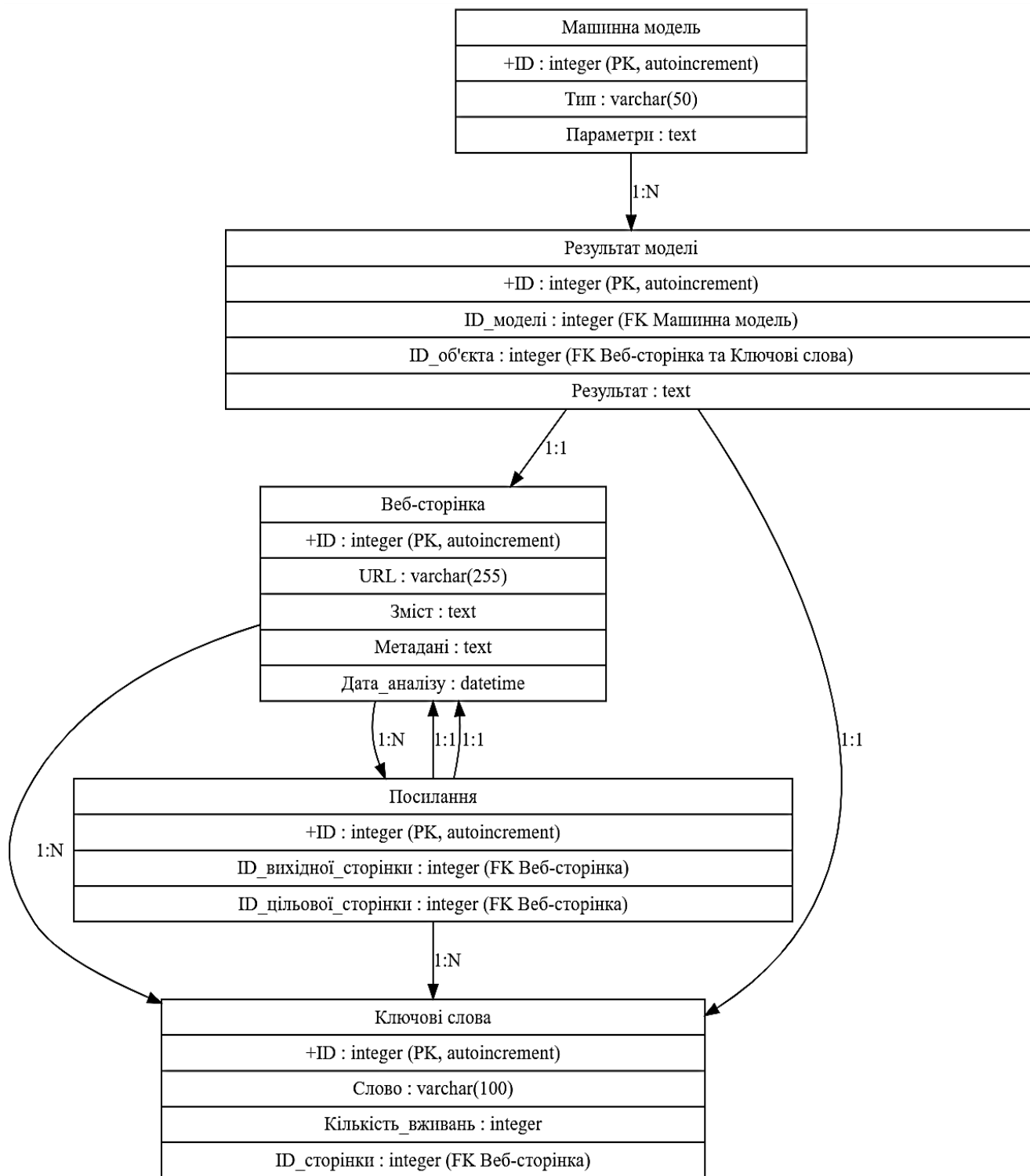


Рисунок 2.3 – Інформаційна модель бази даних системи

Зв'язки та їх типи:

- Вебсторінка – Ключові слова (один до багатьох);
- Вебсторінка – Посилання (один до багатьох);
- Посилання – Вихідна сторінка (один до одного);

- Посилання – Цільова сторінка (один до одного);
- Модель – Результат (один до багатьох);
- Результат – Вебсторінка (один до одного);
- Результат – Ключове слово (один до одного).

Кожна сутність буде розміщена в окремій таблиці.

Також у деяких атрибутів вказані додаткові обмеження:

- атрибут ID кожної сутності відзначений як «integer» та має обмеження "PK" (primary key) для позначення первинного ключа.

- атрибут ID_сторінки у сутності «Ключові слова» та атрибути ID_вихідної_сторінки та ID_цільової_сторінки у сутності «Посилання» відзначені як «integer» та мають обмеження «FK» (foreign key) до відповідних таблиць.

Також необхідно врахувати правила цілісності даних:

- Ідентифікатори (поля ID) в усіх таблицях повинні бути унікальними та автозбільшуватися. Це забезпечить унікальність кожного запису.

- Всі посилальні поля (ID_сторінки, ID_вихідної_сторінки, ID_цільової_сторінки, ID_моделі, ID_об'єкта) повинні вказувати на існуючі значення первинних ключів в інших таблицях. Це гарантує цілісність зв'язків між даними.

- Значення полів URL, Слово повинні бути унікальними в межах окремих таблиць Вебсторінка і Ключові слова відповідно.

- Поля Дата/Час повинні містити коректні дату і час.

Також можуть бути встановлені додаткові обмеження для окремих полів, наприклад мінімальна/максимальна довжина тексту.

Така інформаційна модель дозволить ефективно зберігати дані, необхідні для аналізу сайтів, а також отримувати їх для подальшої обробки в системі.

Таким чином, ретельно спроектована інформаційна модель є важливою основою не лише для бази даних, але й для успішної розробки системи в цілому на всіх етапах її життєвого циклу.

Висновки до розділу 2

В розділі 2 розроблена функціональна модель системи SEO-аналітики на основі методів штучного інтелекту дозволила чітко уявити логіку її роботи на етапі проектування та подати у вигляді графічної схеми. Визначення окремих компонентів моделі дасть змогу ефективно розподілити функціонал поміж ними та оптимізувати взаємодію. Наявність моделі допоможе скласти плани розробки окремих модулів системи та їх інтеграції, а також для написання сценаріїв тестування. Функціональна модель є складовою технічної документації на розробку системи і дасть можливість поетапно удосконалювати її, доповнюючи новими можливостями. Таким чином, розроблена функціональна модель є необхідною основою для наступних етапів ефективної розробки системи SEO-аналітики.

Інформаційна модель бази даних, розроблена для системи SEO-аналітики, дала змогу чітко уявити структуру та логіку зберігання даних, необхідних для функціонування системи. Визначення основних сутностей, їх атрибутів і зв'язків між сутностями дозволить ефективно зберігати та отримувати дані для подальшої обробки. Це є важливою основою не тільки для розробки бази даних, а й для успішної розробки всієї системи на всіх етапах її життєвого циклу, а також для спілкування всіх, хто буде над нею працювати. Ретельно спроектована інформаційна модель допоможе налагодити ефективну роботу з даними та забезпечить цілісність інформації, що зберігається.

3 АРХІТЕКТУРА, МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1. Вибір інструментів для розробки системи SEO-аналітики вебсайтів

Правильний вибір інструментів для розробки програмного продукту є одним з ключових етапів, який впливає на ефективність та якість розробки.

Доцільно обрати інструменти, які забезпечать високу продуктивність розробки, швидку реалізацію функціоналу, масштабованість та підтримку в подальшому. Також важливо орієнтуватись на популярність технологій, наявність документації та підтримки.

Тому на початковому етапі планування продукту слід приділити належну увагу аналізу та вибору інструментального забезпечення – це дасть змогу максимально ефективно розробляти проєкт та досягти поставлених цілей.

Обрано мову програмування Python [25] для розробки системи SEO-аналітики вебсайтів, оскільки Python є інтерпретованою мовою зі зручною синтаксичною структурою, що спрощує процес програмування та зменшує ймовірність помилок.

В Python існує велика кількість бібліотек для роботи з текстами, скрапінгу, аналізу даних, машинного навчання та інших задач, необхідних для проєкту. Це забезпечить прискорення розробки.

Python є універсальною мовою, придатною для розробки як вебсервісів, так і десктопних застосунків. Це дасть можливість створити як саму систему, так і її інтерфейс.

Існує велика кількість онлайн-ресурсів, документації та посилань на можливі рішення, що забезпечить надійну підтримку під час розробки.

Python є популярною мовою, тому знайдеться багато фахівців, які зможуть підтримувати і розвивати проєкт у майбутньому.

Отже, саме Python надасть можливість ефективно розробити систему SEO-аналізу вебресурсів.

Для реалізації проєкту обираємо інтегроване середовище розробки PyCharm (Community-версія), оскільки воно є найкращим рішенням порівняно з простими текстовими редакторами на зразок стандартних блокнотів [26].

На відміну від звичайних текстових редакторів, PyCharm містить усі необхідні інструменти для ефективної розробки на Python, такі як:

- повноцінний інтерпретатор мови;
- розширена підтримка синтаксису та автодоповнення;
- вбудовані інструменти налагодження;
- інтеграція з найпопулярнішими бібліотеками та фреймворками;
- управління версіями та резервуванням коду.

Це дозволить значно підвищити якість продукту в порівнянні з простими текстовими редакторами. Крім того, Community-версія PyCharm є безкоштовною.

Тому саме це середовище найкраще підходить для реалізації поставлених завдань.



Рисунок 3.1 – Інструменти для backend-частини

Для зберігання даних в системі SEO-аналітики обрано базу даних SQLite. Однією з головних переваг SQLite є те, що вона є однофайловою базою даних – всі дані зберігаються в одному файлі. Це робить її дуже простою та невимогливою до ресурсів [27].

Також SQLite характеризується простотою конфігурування та доступу – для роботи з нею не потрібні складні налаштування серверів чи клієнтів. Всі операції можна виконувати безпосередньо у коді програми.

Оскільки створюваний програмний продукт є безкоштовним та відкритим, то немає необхідності у додатковому шифруванні даних. SQLite повністю задовольнить вимоги проєкту по роботі з даними.

Таким чином, SQLite ідеально підходить як просте і невимогливе рішення для бази даних системи SEO-аналітики.

Для бекенд-частини системи SEO-аналітики необхідно використовувати бібліотеку BeautifulSoup для збору даних з вебсторінок. BeautifulSoup дозволить ефективно та швидко зчитувати потрібні дані з вебресурсів [28].

Також використовуватиметься бібліотека NLTK (Natural Language Toolkit) для обробки та аналізу текстової інформації. NLTK – це комплекс інструментів для роботи з природною мовою, який має вбудовані моделі для багатьох мов, у тому числі і української. У рамках проєкту NLTK дозволить здійснювати токенізацію тексту для виділення окремих слів, лематизацію для зведення слів до кореневої форми, підрахунок частоти слів на сайті. Це допоможе проаналізувати метатеги та ключові слова на ресурсі, що є важливо для SEO-аналізу [29].

Таким чином, ці бібліотеки забезпечать отримання даних з сайтів та їх подальшу обробку для отримання результатів аналізу.

Для фронтенд-частини системи SEO-аналітики використовуватимуться мови HTML, CSS та JavaScript [30].

HTML (Hypertext Markup Language) є мовою розмітки і визначатиме структуру сторінок та семантику елементів на них. HTML визначає змістовну частину документа, поділ на розділи та елементи.

CSS (Cascading Style Sheets) є мовою стилів і буде відповідати за візуальне оформлення та розміщення компонентів на сторінці. CSS дозволяє задавати шрифти, кольори, фони, розміри елементів та їх розташування.

JavaScript є мовою скриптів і додасть інтерактивність та динамічні ефекти на сайт. З його допомогою можна додавати різні ефекти, перевірки форм, анімації, обробку подій користувача та роботу з AJAX.

Ці три основні мови дозволять створити повноцінний інтерфейс системи з усіма необхідними функціональними можливостями.

Також для frontend-частини системи SEO-аналітики буде використовуватися фреймворк Bootstrap.

Bootstrap – це популярний фреймворк для створення адаптивних вебсайтів та вебзастосунків. Він поєднує в собі HTML, CSS та JavaScript.

Bootstrap надає вбудовані CSS-стилі для типових елементів інтерфейсу – форм, кнопок, меню, галерей, навігації та інших компонент. Це дозволяє створювати якісний та респонсивний дизайн швидко і легко [31].

Bootstrap автоматично оптимізує макет для різних розмірів екранів – від смартфонів до великих моніторів, також фреймворк містить JavaScript-плагіни, що додають інтерактивність на сайт.

Тому Bootstrap спростить та прискорить процес розробки фронтенд-частини системи SEO-аналітики.



Рисунок 3.2 – Інструменти для frontend-частини

Тобто, для відображення розробленого фронтенду системи SEO-аналітики буде створено вебінтерфейс.

Створений вебінтерфейс буде передаватися користувачеві через використання бібліотеки Eel [32].

Eel – це бібліотека для Python, яка дозволяє виконувати Python-скрипти безпосередньо в браузері. Таким чином, інтерфейс зможе динамічно генеруватися на основі даних та роботи бекенду системи.

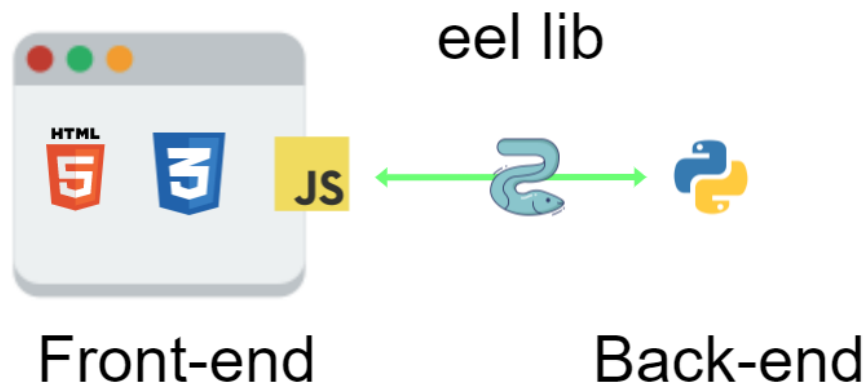


Рисунок 3.3 – Інструмент для взаємодії frontend та backend

Бібліотека Eel дозволить ефективно пов'язати фронтенд та бекенд-частини розроблюваної системи SEO-аналітики.

Наявність API Eel дозволяє викликати Python-функції безпосередньо з JavaScript-коду вебсторінки. Таким чином фронтенд зможе взаємодіяти з функціоналом, що реалізований на Python у бекенді.

При цьому фронтенд та бекенд залишатимуться окремими незалежними частинами системи, а Eel забезпечить їх тісну інтеграцію.

Підбір та обґрунтування використання різних програмних інструментів дозволить успішно створити якісний програмний продукт - систему SEO-аналітики вебсайтів.

Обрані мова Python та інструменти є надійними, перевіреними та широко поширеними. Це зменшує ризики в забезпеченні підтримки проєкту.

Такий підхід реалізуватиме усі ключові функціональні та якісні характеристики майбутньої системи. Завдяки цьому вдалося підібрати оптимальні рішення для розробки якісного продукту.

3.2. Розробка архітектури програмного забезпечення

Розробка архітектури для майбутньої системи є дуже важливою складовою процесу створення якісного програмного забезпечення.

По-перше, правильно спроектована та описана архітектура дозволяє краще спланувати розробку - розподілити функціональність та відповідальності між модулями, визначити їх взаємодії. Це полегшує керівництво проектом.

По-друге, архітектура формалізує структуру майбутньої системи, чітко описуючи її компоненти та зв'язки між ними. Це забезпечує краще розуміння та підтримку складних елементів програми.

Також архітектура дозволяє розробникам працювати, полегшуючи скоординованість їхніх зусиль. Добре спроектована структура забезпечує майбутню масштабованість та гнучкість програми.

Тому обов'язкове створення архітектури на початкових етапах дозволяє максимізувати якість та ефективність розробки програмного забезпечення.

MVC (Model-View-Controller) – це один із популярних шаблонів архітектури проектування програмного забезпечення [33].

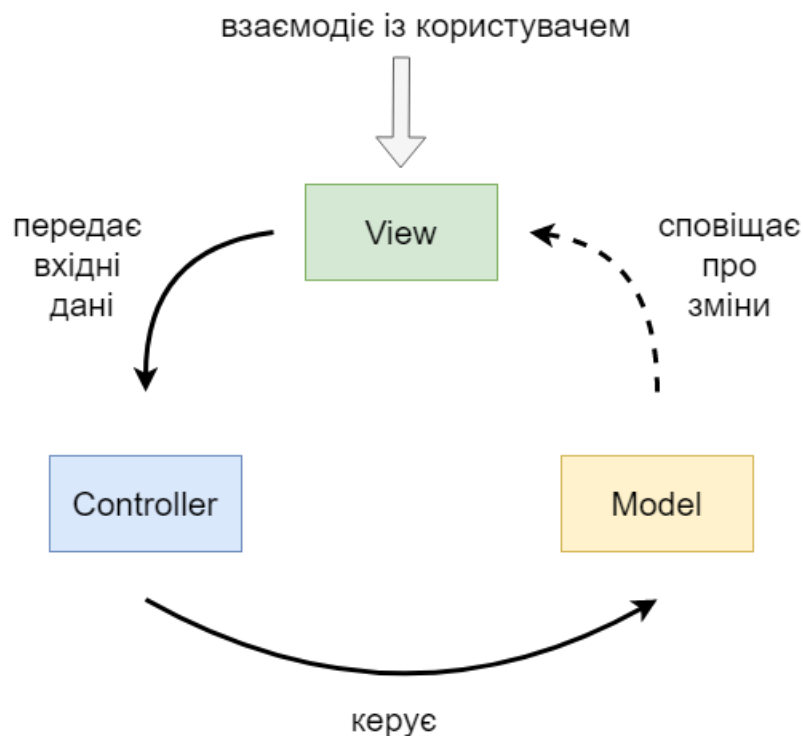


Рисунок 3.4 – Архітектура MVC

Model – це модель, що відображає дані та бізнес-логіку програми. Модель не знає про перегляд та контролер.

View – це представлення інформації, зазвичай це HTML, що генерується на підставі даних моделі. Перегляд не знає про модель та контролер.

Controller – це керувальний елемент, який приймає вхідні запити від користувача, обробляє їх, взаємодіє з моделлю для отримання/збереження даних і видає відповіді перегляду для відтворення.

Разом ці три елементи утворюють MVC. Контролер отримує запит від користувача, запитує дані до моделі та передає їх перегляду для відображення. Це дозволяє реалізувати розподілення відповідальності та інкапсуляцію.

Архітектура системи SEO-аналітики має такі особливості:

- тришарова архітектура MVC:
 - презентаційний шар (web) - фронтенд на HTML/CSS/JS;
 - бізнес-логіка (seo_script.py) - основний сценарій аналізу;
 - доступ до даних (database.db) - SQLite база для зберігання даних;
- використовується модульна структура з поділом на пакети за принципом відповідальності;
- фронтенд розроблено за допомогою фреймворка Bootstrap для кращої сумісності;
- підтримка складних бібліотек (NLTK, BeautifulSoup) реалізована в окремих модулях;
- застосовано основні принципи OOP, MVC і DRY для забезпечення гнучкості та зручності розробки;
- файлова структура відповідає стандартам Python і полегшує підтримку проекту.

Таким чином, запропонована архітектура дозволить ефективно розробити функціональну та масштабовану систему SEO-аналітики.

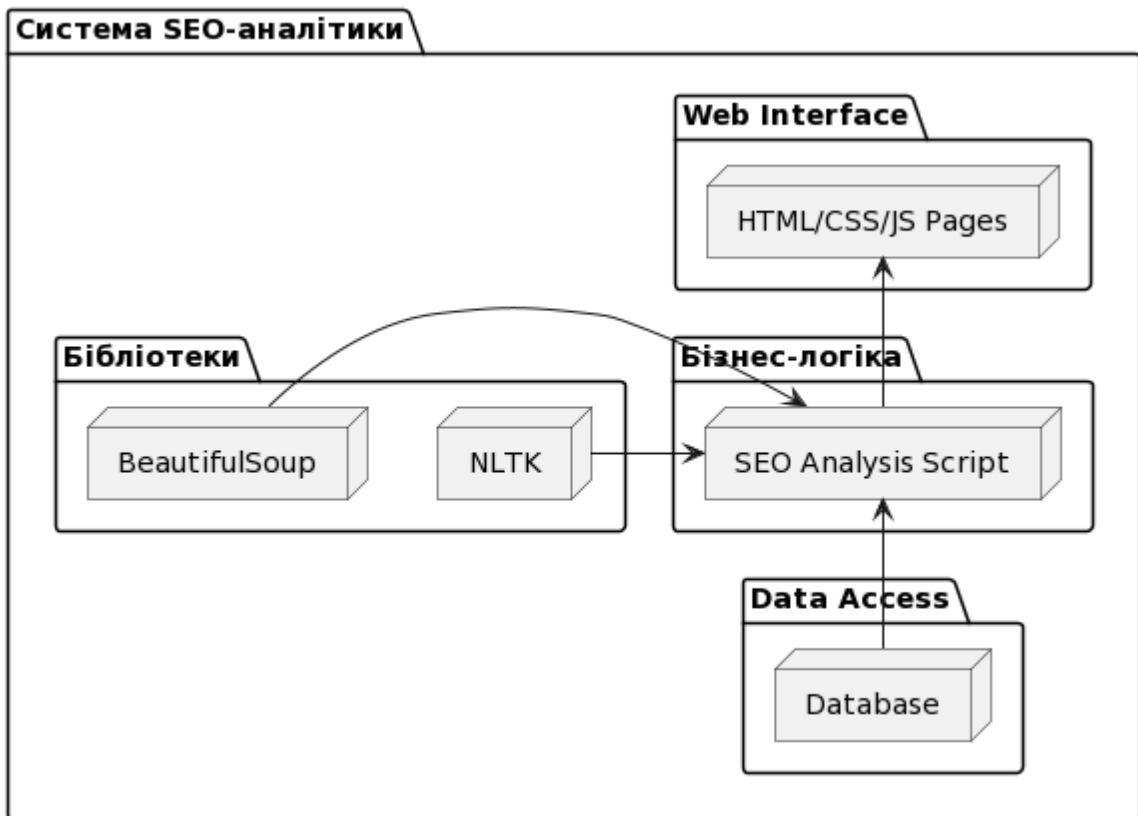


Рисунок 3.5 – Архітектура системи SEO-аналітики вебсайтів

У цій архітектурі:

- «Web Interface» включає вебсторінки, які використовуються для взаємодії з користувачем;
- «SEO Analysis Script» представляє бізнес-логіку системи. Він виконує аналіз вебсторінок, використовуючи дані, отримані з "Database", а також використовує бібліотеки NLTK та BeautifulSoup для обробки тексту та парсингу HTML відповідно;
- «Database» використовується для зберігання даних, необхідних для аналізу SEO;
- «HTML/CSS/JS Pages» представляє вебсторінки, які відображають результати SEO-аналізу і взаємодіють з «SEO Analysis Script».

Ця архітектура дозволяє розділити функціональність системи на окремі компоненти, що полегшує розробку, тестування та підтримку. Компоненти взаємодіють один з одним через визначені інтерфейси, такі як виклики функцій або доступ до бази даних.

3.3. Діаграмне моделювання системи SEO-аналітики вебсайтів

Діаграмне моделювання – це процес створення формалізованого опису системи у вигляді діаграм, які дозволяють зобразити її структуру та властивості.

Для діаграмного моделювання системи SEO-аналітики буде використана мова моделювання UML [34].

UML являє собою уніфікований стандарт, який дозволяє описувати та зображувати різні аспекти програмної системи. З його допомогою можна задокументувати та візуалізувати будову проєкту.

Це досягається за рахунок набору графічних позначень – діаграм, котрі моделюють різні сторони системи. Зокрема, можна показати структуру шляхом класових та компонентних діаграм, а також процеси за допомогою діаграм поведінки.

Такий підхід дозволить описати як складові проєкту SEO-аналітики, так і взаємодію та інформаційні зв'язки між ними на різних рівнях деталізації. При цьому усі елементи будуть зображені в єдиному форматі, що зручно для спільної роботи.

Тому UML як універсальна мова моделювання допоможе ефективно змоделювати запропоновану систему для подальшої реалізації.

Діаграма класів – це схематичне зображення структури системи та взаємозв'язків між її компонентами в об'єктно-орієнтованому форматі [35].

Вона відображає класи (абстрактні типи даних) у системі та їх властивості і методи, відношення між класами, такі як спадкування, агрегація, реалізація інтерфейсів, а також залежності між класами, які показують в який спосіб один клас використовує інший.

Створюючи діаграму класів для нашої системи, виділимо основні класи, що відповідають за функціональні частини системи, покажемо відношення між класами (які дані/методи передаються, у якому напрямку) і встановимо залежності класів від інших системних компонентів, наприклад бібліотек. Це допоможе краще зрозуміти логіку та архітектуру реалізації системи в цілому.

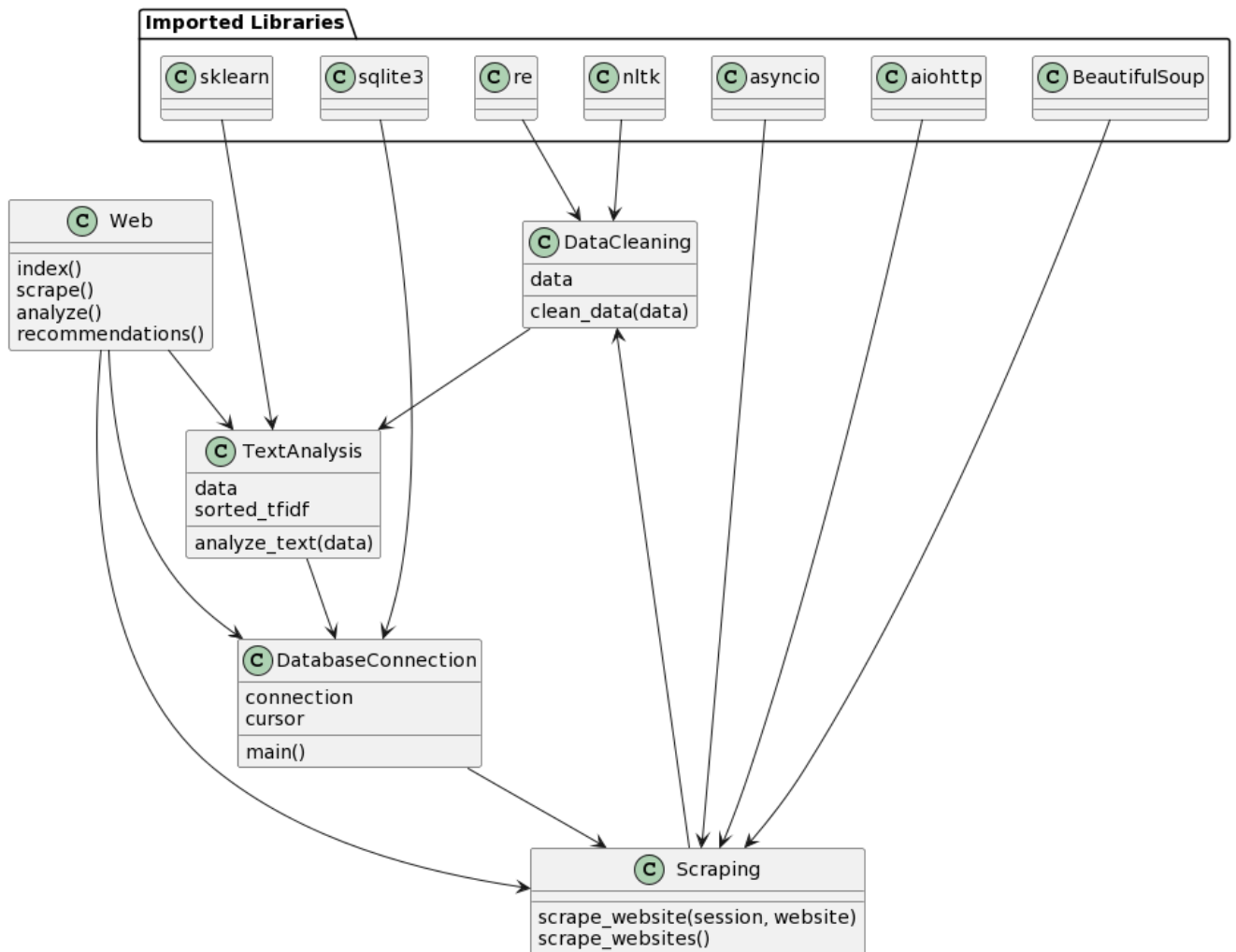


Рисунок 3.6 – Діаграма класів системи SEO-аналітики вебсайтів

Діаграма послідовності - це вид моделювання бізнес-процесів, яка візуалізує функціональні вимоги до системи. Вона відображає акторів (користувачів системи) та їх роль, прецеденти (функціональні вимоги), які описують кроки взаємодії між акторами і системою, а також кордон системи, що визначає межу між акторами та системою [36].

У діаграмі послідовності для системи SEO-аналізу вебсайтів зображено актора користувача, який взаємодіє з системою через вебінтерфейс. Показані такі функціональні вимоги: завантаження даних сайту, очищення та аналіз тексту, показ результатів аналізу, збереження результатів в базу даних. Також відображено отримання рекомендацій по аналізованим текстовим даним.

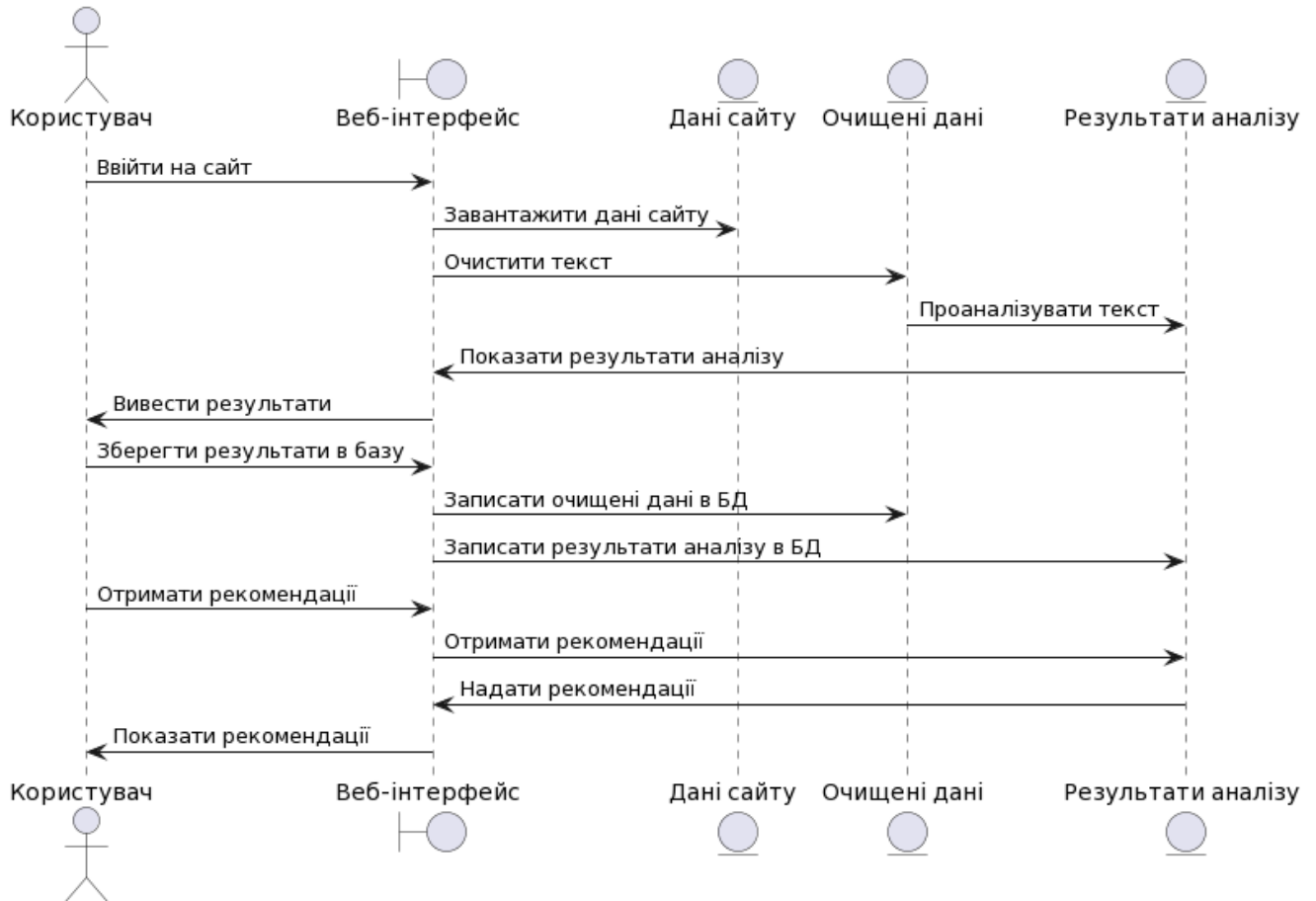


Рисунок 3.7 – Діаграма послідовності системи SEO-аналітики вебсайтів

Діаграма візуально демонструє процеси взаємодії користувача з системою для отримання необхідної інформації в рамках зазначених функціональних можливостей.

Діаграма діяльності - це вид діаграми, яка описує послідовність кроків і логіку виконання процесу [37].

Вона візуально відображає послідовність дій, що виконуються певними об'єктами або системою для досягнення поставленої мети. Корисною особливістю діаграми діяльності є можливість визначити альтернативні шляхи виконання процесу та умови їх вибору.

Діаграма діяльності дозволяє:

- вивчити та проаналізувати роботу системи чи бізнес-процесу;
- виявити недоліки та потенціал для оптимізації;
- розробити алгоритми та псевдокод;

– документувати процес заради подальшого аудиту та вдосконалення.



Рисунок 3.8 – Діаграма діяльності системи SEO-аналітики вебсайтів

У діаграмі діяльності системи SEO-аналітики вебсайтів зображено послідовний процес аналізу тексту: від завантаження даних і очищення, до визначення ключових слів, кластеризації, отримання й виведення результатів. Передбачено можливість обробки помилки завантаження.

У результаті моделювання системи SEO-аналітики вебсайтів було створено такі діаграми:

- діаграму класів, яка описує структуру та взаємозв'язки основних класів і об'єктів системи. Це допоможе розробникам правильно спроектувати архітектуру програмного коду;

- діаграму послідовності, що відображає функціональні можливості системи та взаємодію з нею користувачів та зовнішніх систем. Це уточнить функціональні вимоги до системи;

- діаграму діяльності, яка наочно описує послідовність кроків аналізу даних в системі - від імпорту даних до отримання результатів. Це допоможе розробити алгоритми та оптимізувати процес.

У результаті створені діаграми допоможуть краще розібратись у логіці функціонування системи, спроектувати її на етапі розробки та вдосконалити в майбутньому. Це полегшить подальшу роботу над проєктом.

3.4. Аналіз сучасних програм для проведення SEO-аналітики

Проведемо аналіз сучасного ринку програмного забезпечення для проведення SEO-аналітики та виявлення перспектив розробки власного програмного рішення.

SEO-аналітика є важливим інструментом для оптимізації вебсайтів та підвищення їх позицій в органічних пошуках. Сформувався стійкий попит на якісні та функціональні сервіси, здатні забезпечити глибокий аналіз сайтів та давати рекомендації експертів.

У рамках дослідження проаналізуємо чотири найбільш популярних на сьогодні інструменти SEO-аналітики: Ahrefs, Google Analytics, Moz та SEMrush. Буде розглянуто їхні функціональні можливості, переваги та недоліки для виявлення перспективних напрямків розвитку власного рішення.

Ahrefs є одним з найпопулярніших інструментів для SEO-аналітики, використовуваних в професійному середовищі. Він володіє широким набором функцій та можливостей, які допомагають вивчати та аналізувати різні аспекти оптимізації пошукової системи [38].

Одна з основних функцій Ahrefs полягає у визначенні ключових слів. Цей інструмент надає користувачам змогу здійснювати дослідження та аналіз потенційно вигідних ключових слів для покращення позиціонування їх вебсторінок у пошукових системах.

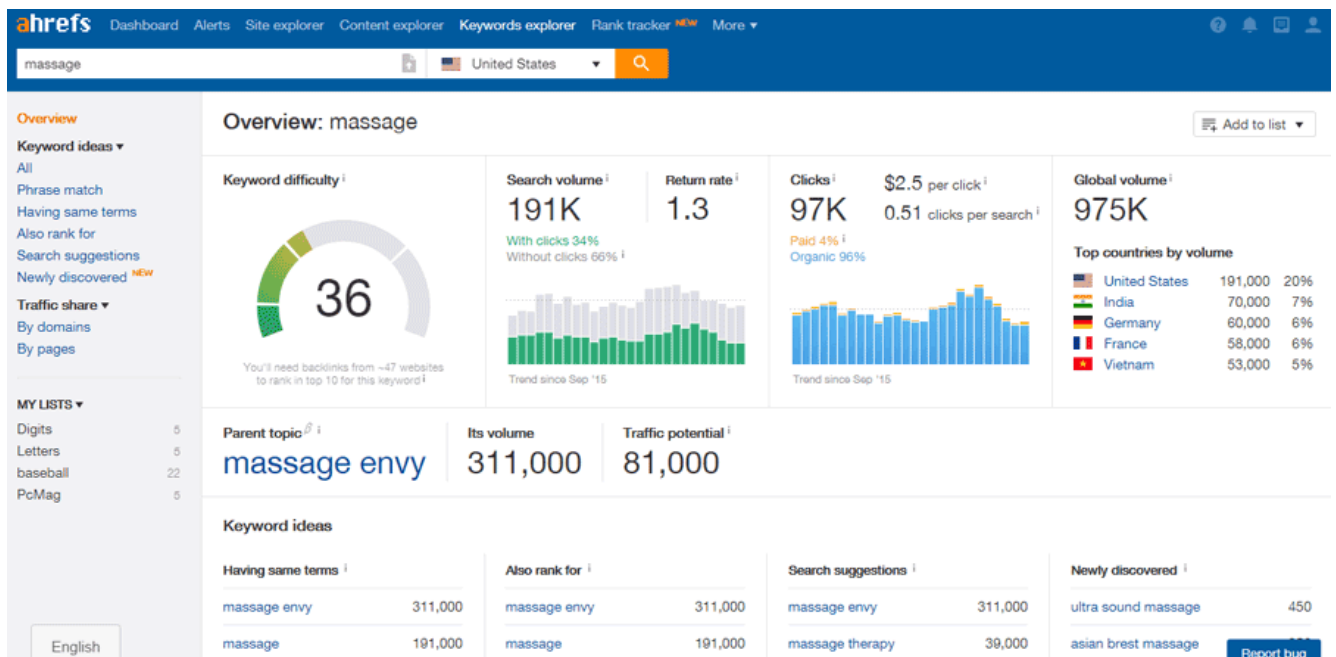


Рисунок 3.9 – Інтерфейс Ahrefs

Крім вивчення ключових слів, Ahrefs дозволяє аналізувати конкурентів. Користувачам надається можливість вивчати та порівнювати стратегії конкурентів, а також виявляти їхні слабкі місця та переваги. Ця функція допомагає підвищити конкурентоспроможність вебсайту й покращити його позицію у пошукових системах.

Перевірка зовнішніх посилань є ще однією важливою функцією Ahrefs. Інструмент аналізує посилання, які спрямовані на сайт, та надає важливі дані щодо якості та кількості цих посилань.

Рейтинги є ще однією вагомою характеристикою, яку забезпечує Ahrefs. Інструмент дозволяє відстежувати та аналізувати рейтинг вебсайту в результатах

пошуку. Також, Ahrefs надає розширені звіти та графіки для візуалізації даних, що допомагають легше розуміти та аналізувати результати вашої SEO-стратегії.

Одним із недоліків Ahrefs є те, що це повністю комерційний продукт – для отримання доступу до повного функціоналу та зручних звітів користувачам необхідно придбати платний пакет. Також Ahrefs не надає підтримки української мови в інтерфейсі та звітах, що ускладнює його використання для українських фахівців з SEO-маркетингу.

Google Analytics є безкоштовним інструментом, розробленим компанією Google, який забезпечує можливість відстежування та аналізу відвідуваності вебсайту. Цей інструмент надає користувачам дані про статистику вебтрафіку, показники конверсії та поведінку користувачів, що робить його корисним інструментом для збору та аналізу даних про вебсайт [39].

Інструмент збирає дані про кількість відвідувачів, перегляди сторінок, тривалість сесій та інші метрики, що дають змогу зрозуміти, як користувачі взаємодіють з вебсайтом. Це допомагає визначити популярність різних сторінок та розділів вебсайту, а також виявити потенційні проблеми, такі як висока швидкість відхилень або низька тривалість сесій.

Крім загальної статистики вебтрафіку, Google Analytics надає користувачам розширені функції, спрямовані на SEO-аналітику. Наприклад, інструмент дозволяє відстежувати ключові слова, за допомогою яких користувачі знаходять вебсайт у результатах пошуку. Це дозволяє визначити ефективність ключових слів та визначити, які терміни є найбільш вдалими для повернення трафіку.

Крім того, Google Analytics надає можливість аналізувати розташування вебсайту в результатах пошуку. Це важливо для визначення, наскільки успішно вебсайт просувається в пошукових системах та які позиції займає у порівнянні з конкурентами. За допомогою цих даних, користувачі можуть виявити потенційні можливості для покращення SEO-стратегії та позиціонування свого вебсайту.

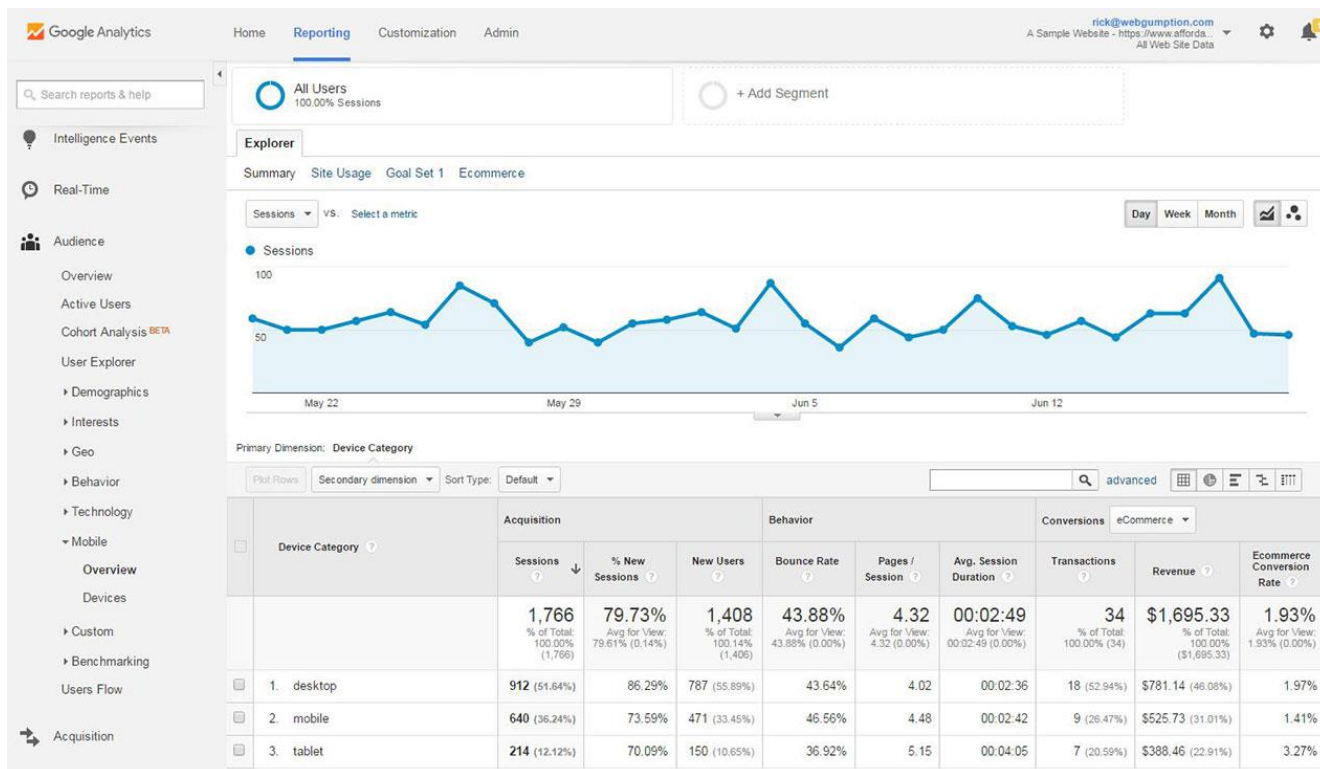


Рисунок 3.10 – Інтерфейс Google Analytics

Основні недоліки Google Analytics:

– відсутній відкритий доступ до коду та API. Це ускладнює модифікацію інструменту та його налаштування під конкретні потреби користувача. Немає можливості розширити функціонал самостійно;

– дані збираються та обробляються на серверах Google, що створює залежність від цієї компанії та обмежує конфіденційність даних.

Moz є комплексним набором інструментів, спеціально розроблених для здійснення SEO-аналітики. Цей набір інструментів надає широкий спектр можливостей для дослідження ключових слів, аналізу конкурентів, вимірювання авторитету домену, аналізу посилань та інших важливих аспектів, пов'язаних з оптимізацією пошукової системи [40].

Один з основних функціональних блоків Moz - це визначення ключових слів. За допомогою набору інструментів Moz користувач може провести детальний аналіз ключових слів, що дозволяє визначити їхню популярність, зміни в часі, а також конкурентну ситуацію. Це допомагає вибрати оптимальні ключові слова для вебсторінок та побудувати ефективну стратегію оптимізації контенту.

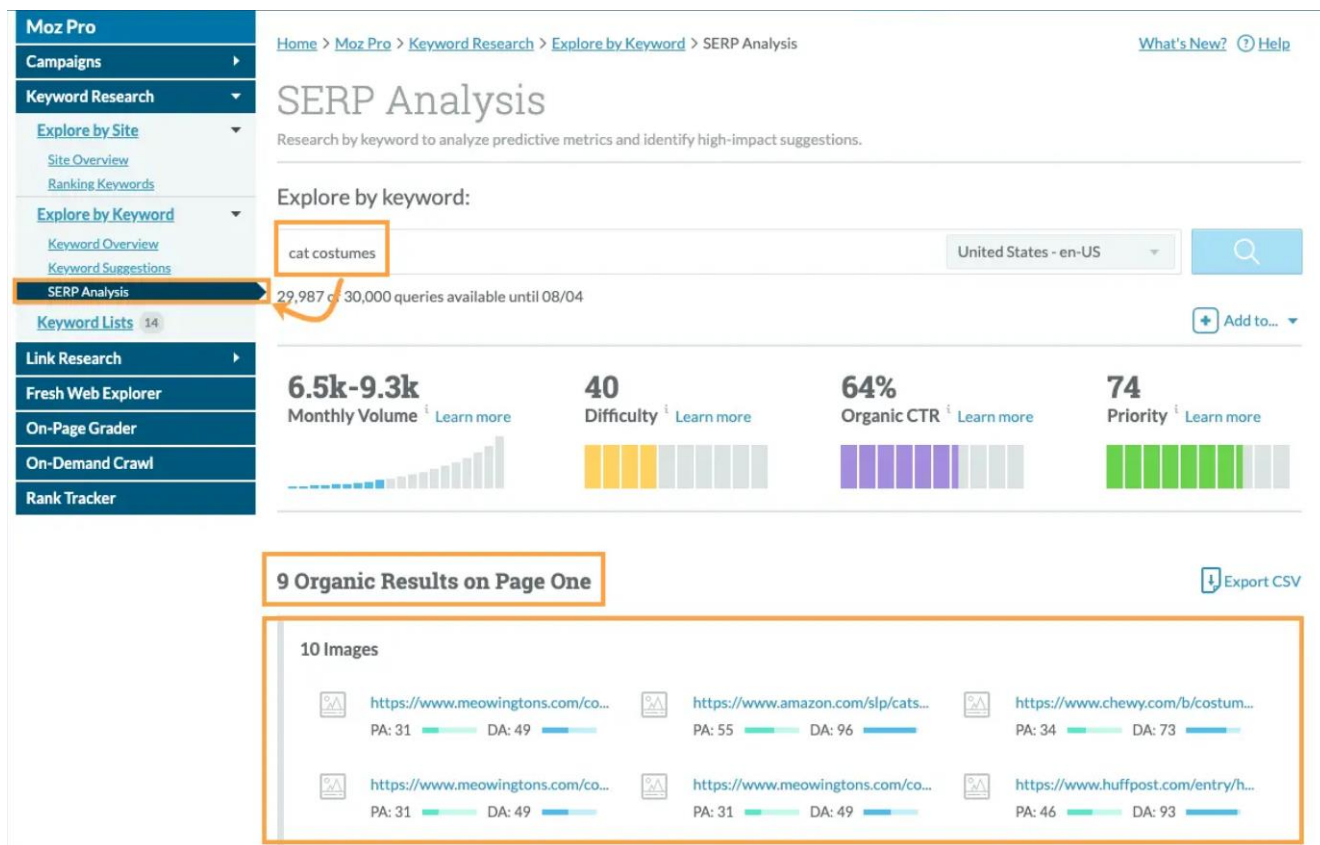


Рисунок 3.11 – Інтерфейс Moz

Крім того, Moz надає можливість аналізу конкурентів. Це дозволяє отримати цінні знання для розробки власної ефективної стратегії SEO.

Також Moz надає можливість вимірювати авторитет домену. Це оцінка важливих метрик, таких як доменна авторитетність та сторінкова авторитетність. Ці метрики дозволяють зрозуміти, яка репутація вебсайту в очах пошукових систем, що в свою чергу впливає на його позиціонування в пошукових результатах.

Ще Moz надає інструменти для аналізу посилань. Це допомагає визначити потенційні можливості для отримання якісних посилань і покращення позицій в пошукових системах.

Основні недоліки сервісу Moz:

- базовий функціонал обмежений, основні можливості доступні лише в платній підписці;
- відсутня локалізація під українську мову, весь інтерфейс та документація англійською;

- немає можливості отримати доступ до вихідного коду та API інструменту. це унеможлиблює модифікацію чи налаштування під конкретні потреби;
- немає опції розширити функціонал та налаштування самостійно через обмеження доступу до коду.

SEMrush є популярним інструментом для SEO-аналітики, який надає широкий спектр функцій. Однією з головних функцій є вивчення ключових слів.

Крім того, SEMrush надає можливості для аналізу конкурентів, де користувачі можуть вивчати сильні та слабкі сторони конкурентів, оцінювати їхні оптимізаційні стратегії та знаходити можливості для покращення власної SEO-стратегії. Оцінка трафіку є ще однією важливою функцією SEMrush. Вона дозволяє користувачам оцінювати кількість відвідувачів, які приходять на вебсайт, та аналізувати їх поведінку [41].

SEMrush також надає інструменти для аналізу посилань, де користувачі можуть вивчати посилання, які ведуть на їх вебсайт, оцінювати їх якість та репутацію, а також відстежувати зміни в динаміці.

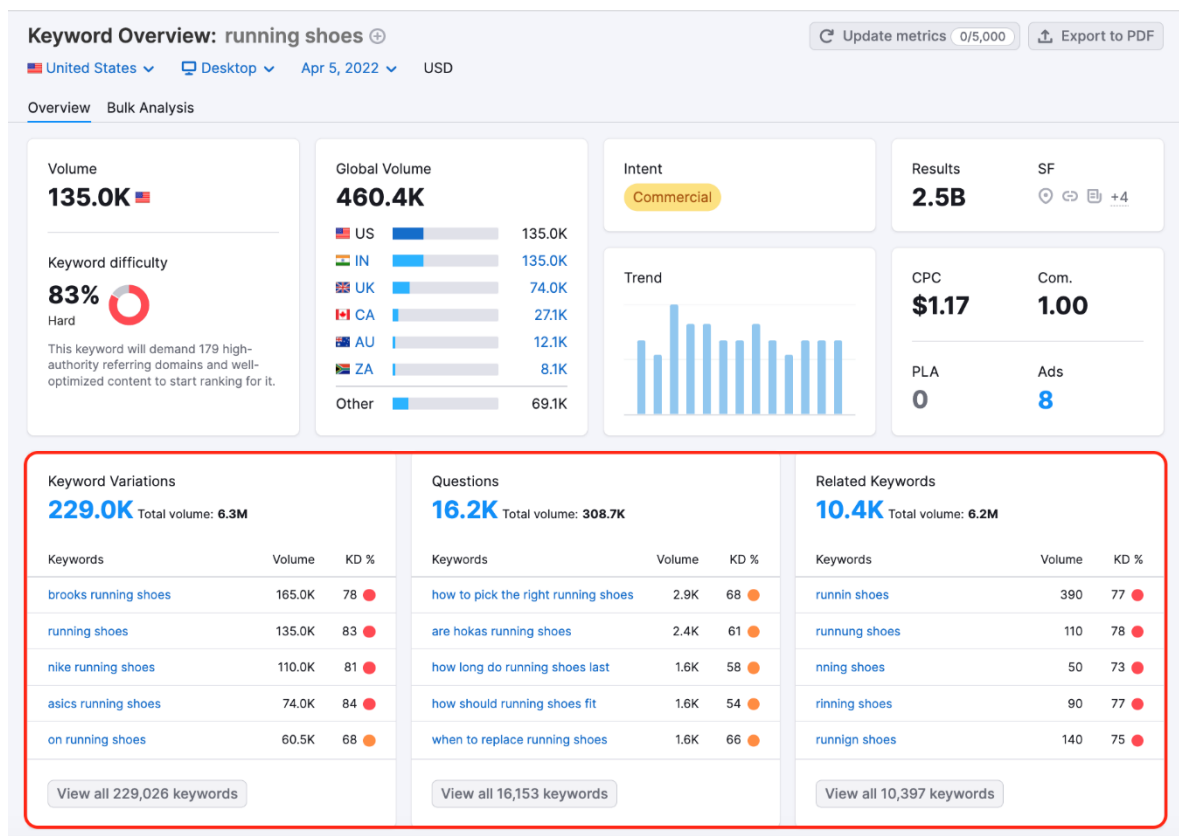


Рисунок 3.12 – Інтерфейс SEMrush

Крім основних функцій, SEMrush пропонує інструменти для соціального медіа-маркетингу та контентного маркетингу. Ці інструменти дозволяють користувачам розробляти комплексні SEO-стратегії, які включають в себе не тільки оптимізацію для пошукових систем, але й використання соціальних медіа та цікавого контенту для досягнення високих результатів.

Основні недоліки SEMrush:

- Повністю платний сервіс, для використання більшості функцій необхідно оформлювати підписку, що є додатковим фінансовим навантаженням.
- Інтерфейс та документація лише англійською, відсутня локалізація українською, що ускладнює роботу українським користувачам.
- Немає відкритого доступу до API та коду, тому неможливо самостійно розширювати функціонал чи додавати власні налаштування.
- Періодично виникають технічні проблеми зі швидкістю та сервісом, що знижує якість роботи.

У рамках дослідження проаналізовано такі популярні інструменти як Ahrefs, Google Analytics, Moz та SEMrush. Ці сервіси є лідерами ринку аналітики та дозволяють отримати цінні дані про сайт.

Визначено, що якщо інструмент є функціональним та якісним, він зазвичай є платним (Ahrefs, SEMrush, Moz). Тоді як безкоштовні версії (Google Analytics) мають недоступність вихідного коду.

Саме тому прийнято рішення розробити власний інструмент SEO-аналітики, який буде:

- повністю безкоштовним для користувача;
- не розміщуватись на хмарних сервісах, а працювати локально на ПК користувача;
- мати відкритий вихідний код для можливості модифікацій та розширень.

Така концепція дозволить створити функціональний та якісний інструмент SEO-аналітики, який буде повністю безкоштовним для користувачів.

Висновки до розділу 3

У розділі 3 комплексно розглянуто питання архітектури, моделювання та проєктування програмного забезпечення для системи SEO-аналітики вебсайтів.

Було детально проаналізовано варіанти програмних інструментів для розробки системи та вибрано найбільш оптимальні - мову Python та IDE PyCharm. Це дозволить створити модульну та налаштовувану систему з відкритим вихідним кодом.

Запропоновано детальний опис архітектури системи за трирівневою моделлю MVC (модель, перегляд, контролер). Визначено функції та взаємодії основних складових цієї моделі - бази даних, сервісних компонентів, представлення інтерфейсу. Це забезпечить високу якість архітектури та її масштабованість.

За допомогою веборієнтованих діаграм UML створено детальні моделі класів та послідовностей для найважливіших об'єктів системи. Це дасть змогу ефективно проєктувати логіку програмного коду системи.

Також проаналізовано функціональні можливості аналогічних продуктів, що дозволить визначити перспективні напрямки розвитку власної розробки та уникнути помилок конкурентів.

Отже, у розділі виконано ґрунтовну підготовчу роботу з архітектурного проєктування та моделювання майбутньої системи SEO-аналітики.

4 РОЗРОБКА ТА АПРОБАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ SEO-АНАЛІТИКИ

4.1. Кодування та тестування програмного забезпечення

Код системи SEO-аналітики розпочнемо зі створення основного скрипта `seo_script.py`, в якому буде прописана логіка бекенда.

В `seo_script.py` будуть реалізовані основні функціональні модулі - отримання та аналіз даних, їх зберігання. Цей код буде тестуватися шляхом виводу результатів в консоль.

Спочатку для розробки системи SEO-аналітики ми створюємо віртуальне середовище в Python за допомогою `virtualenv`, це необхідно для ізоляції залежностей проєкту та уникнення конфліктів при встановленні пакетів.

Після активації віртуального середовища, ми встановлюємо необхідні бібліотеки та інструменти за допомогою `pip` та імпортуємо їх на початку в `seo_script.py`.

```
import re
import nltk
import asyncio
import aiohttp
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
from sklearn.feature_extraction.text import TfidfVectorizer
from bs4 import BeautifulSoup
import sqlite3
```

Наступним етапом розробки системи SEO-аналітики буде підготовка лінгвістичних даних.

Для англійської мови ми завантажимо словник стоп-слів з бібліотеки NLTK:

```
nltk.download('stopwords')
```

Для української мови файл із стоп-словами завантажимо локально з репозиторію GitHub:

```
ukrainian_stopwords = open('stopwords_ua.txt', encoding='utf-8').read().split('\n')
```

Також для морфологічного аналізу та лематизації українського тексту завантажимо з NLTK:

```
nltk.download('punkt')
nltk.download('wordnet')
```

Наявність стоп-слів та інструментів обробки мови дозволить ефективно чистити та аналізувати контент сайтів українською.

Наступним кроком буде розробка асинхронної функції для збору даних з сайту:

```
async def scrape_website(session, website):
    async with session.get(website) as response:
        text = await response.text()
        soup = BeautifulSoup(text, 'html.parser')

        # Отримання внутрішніх посилань
        internal_links = []
        for link in soup.find_all('a', href=True):
            href = link.get('href')
            if href.startswith('/') or website in href:
                internal_links.append(href)

        # Отримання метатегів
        meta_tags = []
        for tag in soup.find_all('meta'):
            meta_tags.append(tag.get('content'))

        return soup.get_text(), internal_links, meta_tags
```

Наступним кроком буде написання асинхронної функції `scrape_websites()` (весь скрипт `seo_script.py` наведено в додатку А), яка:

- отримає список сайтів для збору в перемінній `websites`;
- ініціалізує порожні списки для зберігання даних, посилань та метатегів - `data`, `internal_links`, `meta_tags`;
- створить сесію `aiohttp.ClientSession`;
- запустить функцію `scrape_website` для кожного сайту в списку `websites` та збере результати в перемінну `results` за допомогою `asyncio.gather`;
- додасть результати кожної ітерації `scrape_website` до відповідних списків;
- поверне всі зібрані дані разом зі списком оброблених сайтів.

Тобто функція `scrape_websites` об'єднує роботу з декількома сайтами, запускаючи збір кожного в окремому потоці, а потім збираючи отримані дані для подальшого аналізу.

Далі створюємо функцію `clean_data()`:

```
def clean_data(data):
    cleaned_data = ' '.join(data)
    cleaned_data = re.sub(r'\s+', ' ', cleaned_data)
    cleaned_data = re.sub(r'^a-zA-Za-яА-ЯіІїїєЄ\.'"\.\, \- \? \! \s]',
'', cleaned_data)
    return cleaned_data
```

У цій функції отримані текстові дані (параметр `data`) очищуються від зайвого сміття:

- `join()` об'єднує всі елементи списку `data` в один рядок, розділяючи їх пробілом (' ');
- `sub()` за допомогою регулярного виразу заміняє повторювані пробіли одним;
- `sub()` за допомогою іншого регулярного виразу видаляє всі символи, крім літер української та латинської абетки, пробілів, символів пунктуації тощо.

Таким чином ми очищаємо текст від зайвих пробілів, символів та форматування, лишаючи лише корисні слова.

Повертається очищений рядок `cleaned_data`, готовий для подальшого аналізу.

Функція `analyze_text` займається аналізом та обробкою текстових даних за допомогою методів обробки тексту:

Токенізація - розбиває текст на окремі слова.

```
# Токенізація
tokens = word_tokenize(data)
```

Лематизація - виводить кореневу форму слів.

```
# Лематизація
lemmatizer = WordNetLemmatizer()
tokens = [lemmatizer.lemmatize(token) for token in tokens]
```

Фільтрація стоп-слів - видаляє найчастотніші і беззмістовні слова.

```
# Видалення стоп-слів
stop_words = set(stopwords.words('english'))
```

```
tokens = [token for token in tokens if token.lower() not in stop_words]
```

TF-IDF - будує вектор найважливіших слів з урахуванням їх частоти і рідкості.

```
# Побудова TF-IDF векторів
tfidf_vectorizer = TfidfVectorizer()
tfidf_matrix = tfidf_vectorizer.fit_transform([data])

# Отримання словника та TF-IDF значень
feature_names = tfidf_vectorizer.get_feature_names_out()
tfidf_values = tfidf_matrix.toarray()[0]

# Створення словників зі значеннями TF-IDF
tfidf_dict = {feature_names[i]: tfidf_values[i] for i in range(len(feature_names))}
```

Повертає словник слів зі значеннями TF-IDF у порядку спадання.

```
# Відображення слів за значенням TF-IDF у порядку спадання
sorted_tfidf = sorted(tfidf_dict.items(), key=lambda x: x[1], reverse=True)
```

Таким чином код аналізує та виводить ключові слова документу.

В завершенні функція `main()` (Додаток А) викликає функцію `scrape_websites` для збору даних і отримує з неї дані, внутрішні посилання, метатеги та список сайтів. Очищає отримані дані за допомогою функції `clean_data` та аналізує текст за допомогою функції `analyze_text`. Створює базу даних SQLite і таблиці для зберігання даних сайтів, неочищених даних, очищених даних, внутрішніх посилань, метатегів та результатів аналізу і рекомендацій. Заповнює таблиці отриманими даними і генерує рекомендації на основі результатів аналізу. На завершення закриває з'єднання з базою даних, централізуючи всі дані і результати в одній БД. Викликає цю функцію через `asyncio.run(main())` для асинхронного виконання.

При написанні цього коду кожна функція тестувалася окремо шляхом виклику її з модуля і аналізу отриманих результатів, які виводилися в консоль.

Це дозволяло перевірити коректність логіки окремих етапів обробки даних та налагодити взаємодію між функціями перш ніж об'єднати їх в основний скрипт.

Такий підхід дозволяє написати надійний та налагоджений код завдяки поетапній перевірці функціональності.

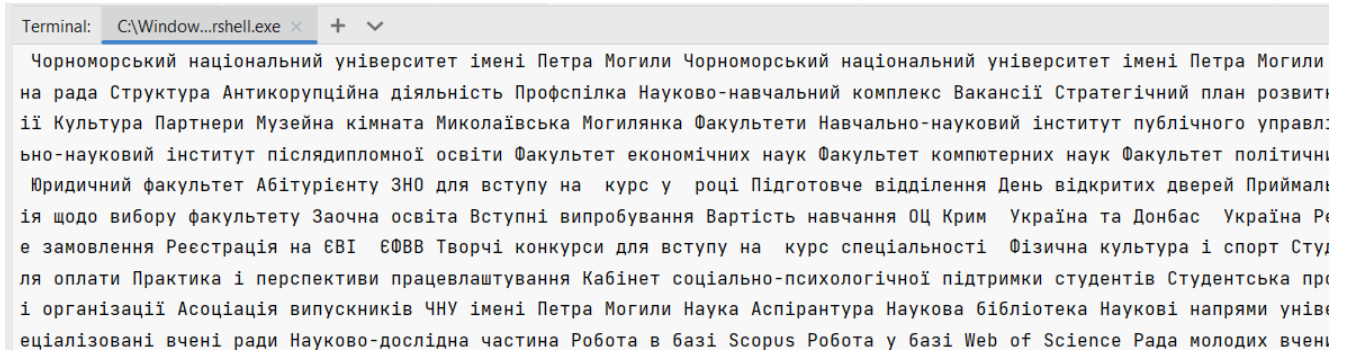


Рисунок 4.1 – Тестування функції `clean_data()`

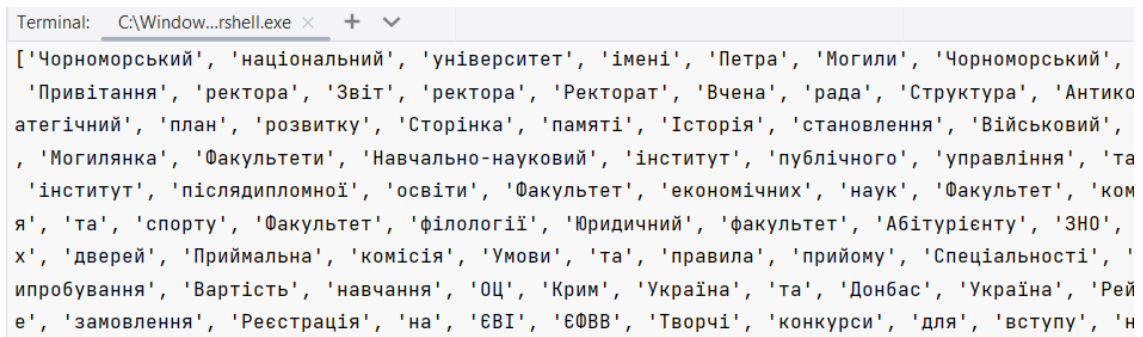


Рисунок 4.2 – Тестування виводу ключових слів

Для перевірки коректності запису даних в створену базу даних SQLite, після завершення виконання основного скрипту, буде використано зовнішню програму - браузер баз даних DB Browser для SQLite [42].

DB Browser для SQLite дозволяє відкривати створені бази типу SQLite і переглядати в ній створені таблиці та записи.

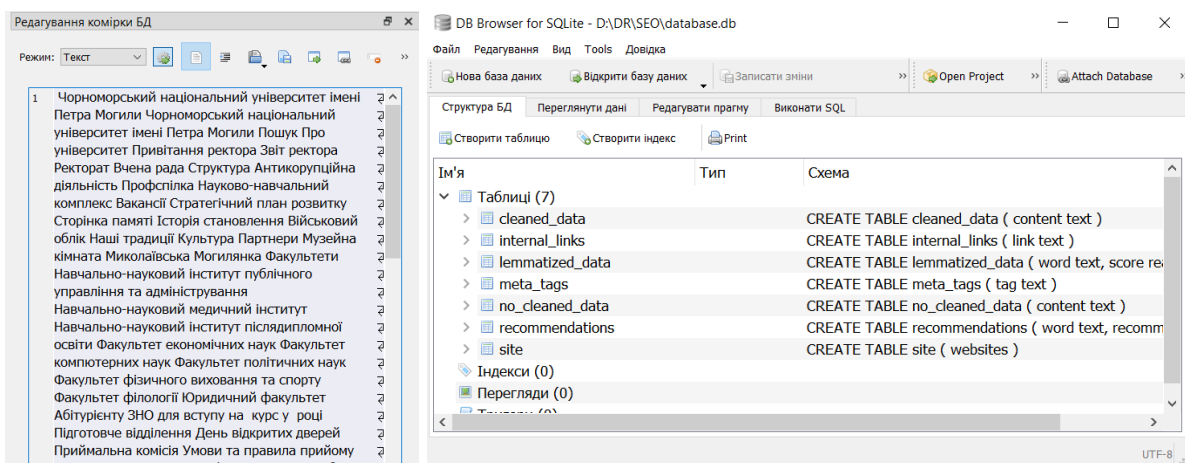


Рисунок 4.3 – Тестування запису даних в базу даних

Після розробки бекенду та створення бази даних із збереженими результатами, наступним кроком буде розробка фронтенду для візуалізації та взаємодії з даними.

Для цього в кореневій папці проєкту створюється папка «web», в якій буде розміщено файлову структуру для фронтенду:

- в папці "web" створюються підпапки "css", "js", "images" для розміщення відповідних типів файлів.
- в корені папки "web" розміщується файл index.html - головна сторінка сайту.
- в "css" поміщаються файли зі стилями для формування дизайну.
- в "js" - javascript файл(и) для логіки та взаємодії з даними.
- в "images" - графічні файли для сайту.

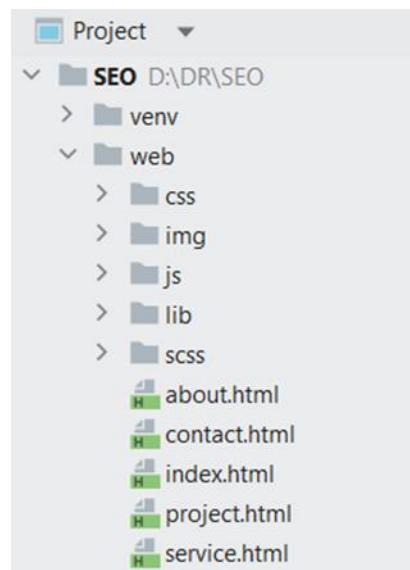


Рисунок 4.4 – Структура папки web

Таким чином, створюється структура для фронтенд частини застосунку, де реалізуємо інтерфейс для перегляду результатів роботи бекенду.

У файлі main.py (див. Додаток Б) реалізовано з'єднання фронтенду з бекендом за допомогою бібліотеки Eel.

Скрипт містить:

- імпорт бібліотек Eel та SQLite для роботи з БД;

- функції з позначкою `@eel.expose`, які виконують запити до БД та повертають значення, доступні з JS;
- метод `eel.init()`, який ініціалізує Eel та вказує доступ до папки з фронтендом;
- підключення до БД та виконання SQL запитів в кожній функції;
- повернення даних;
- `eel.start()` для запуску фронтенду.

Таким чином Python отримує доступ до бази даних, а JS - до функцій Python, що дозволяє побудувати взаємодію.

4.2. Експериментальна апробація програмного забезпечення

Для перевірки коректної роботи розробленого програмного забезпечення було проведено його експериментальну апробацію.

Тестування інтерфейсу фронтенду на коректність відображення даних.

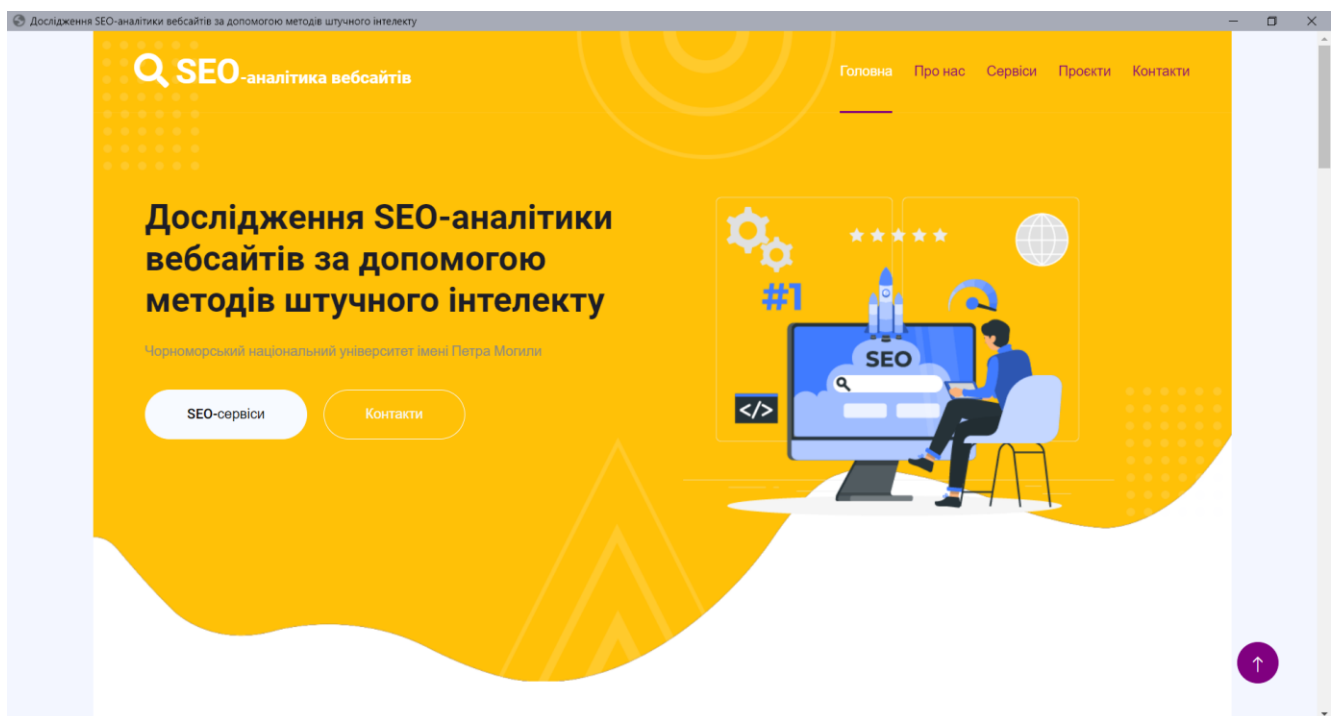


Рисунок 4.5 – Апробація головної сторінки

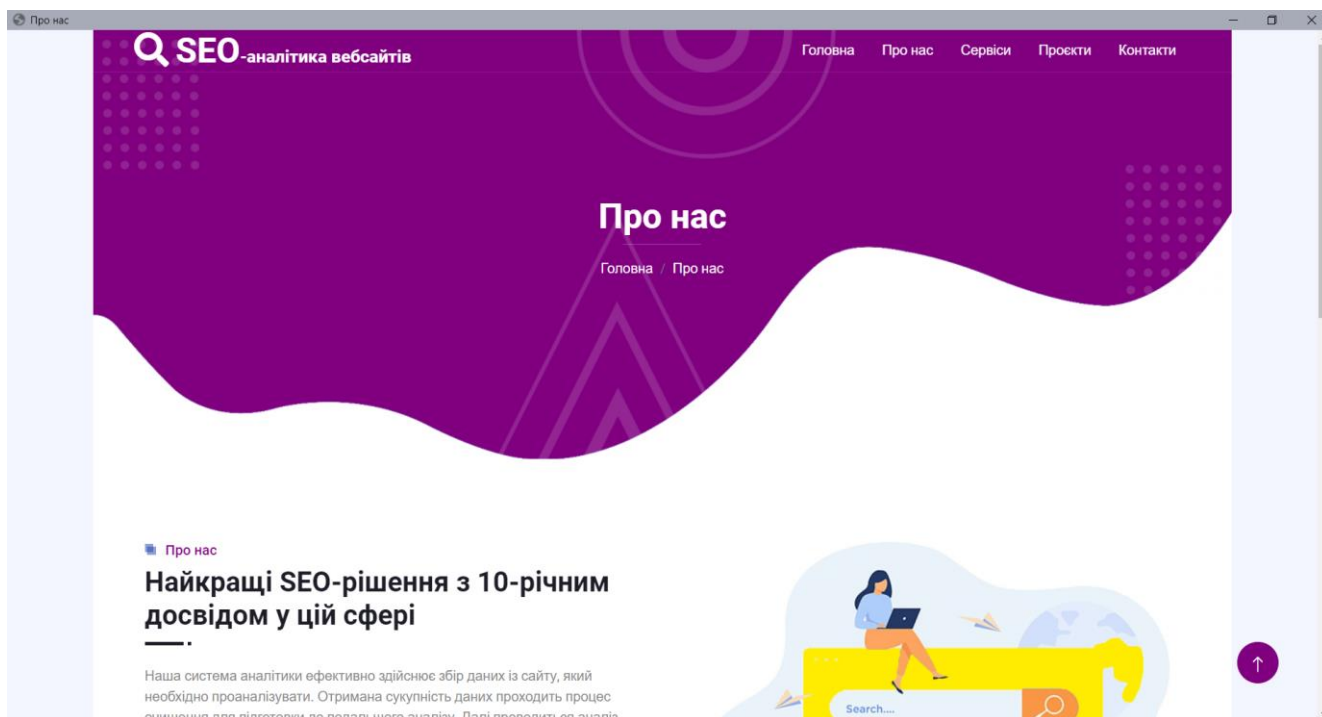


Рисунок 4.6 – Апробація сторінки «Про нас»

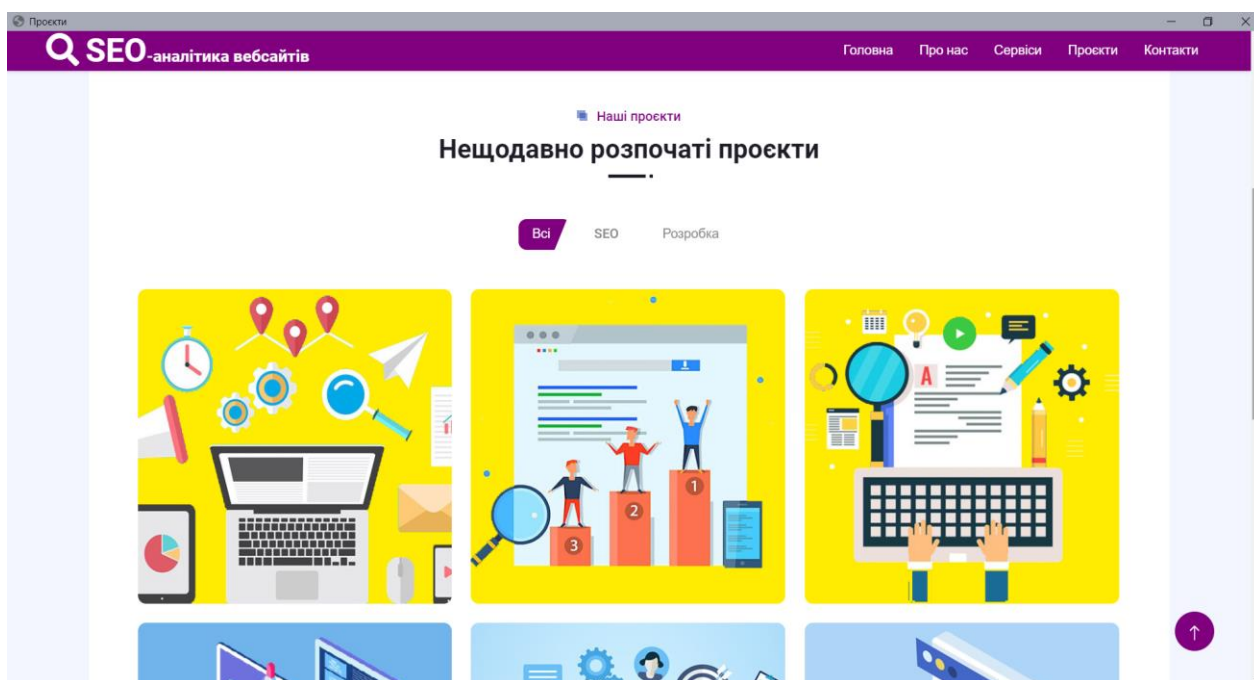


Рисунок 4.7 – Апробація сторінки «Проекти»

При апробації сторінки проекти додатково була перевірена функціональність JavaScript, яка дозволяє фільтрувати відображення проектів в залежності від натиснутого елемента:

- натискання кнопки «Всі» виводить мініатюри усіх проектів на сторінці;

- кнопка "SEO" - тільки три проєкти з категорії SEO;
- кнопка "Розробка" - три проєкти з категорії розробка.

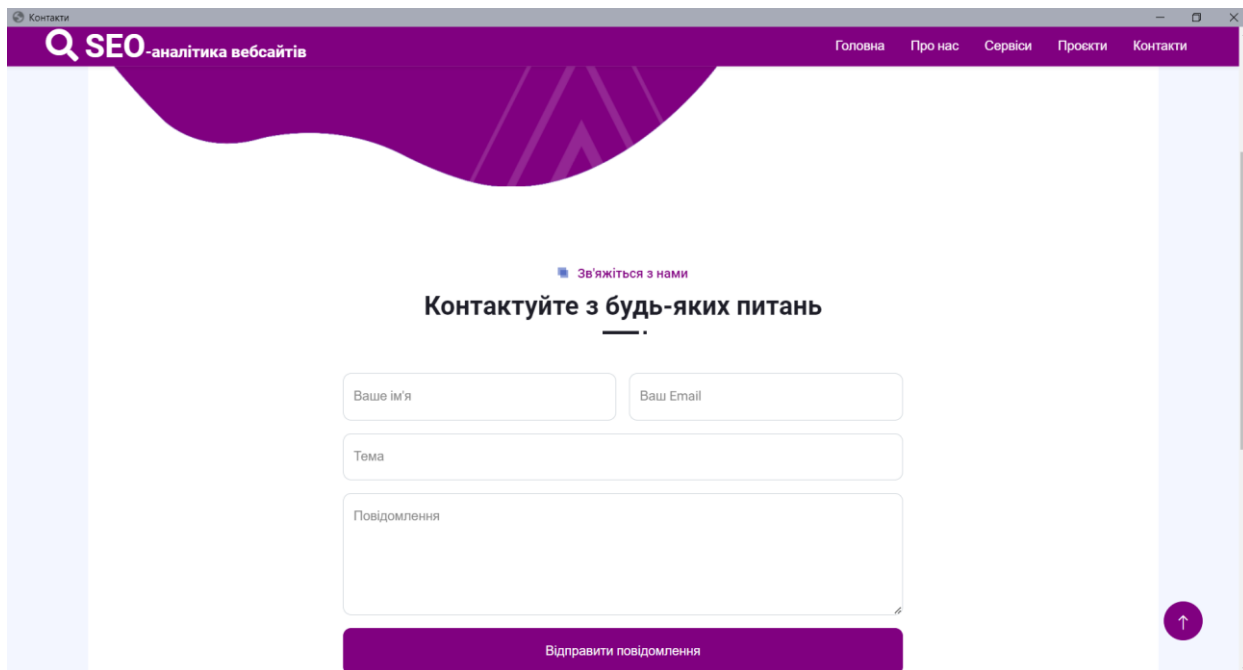


Рисунок 4.8 – Апробація сторінки “Контакти”

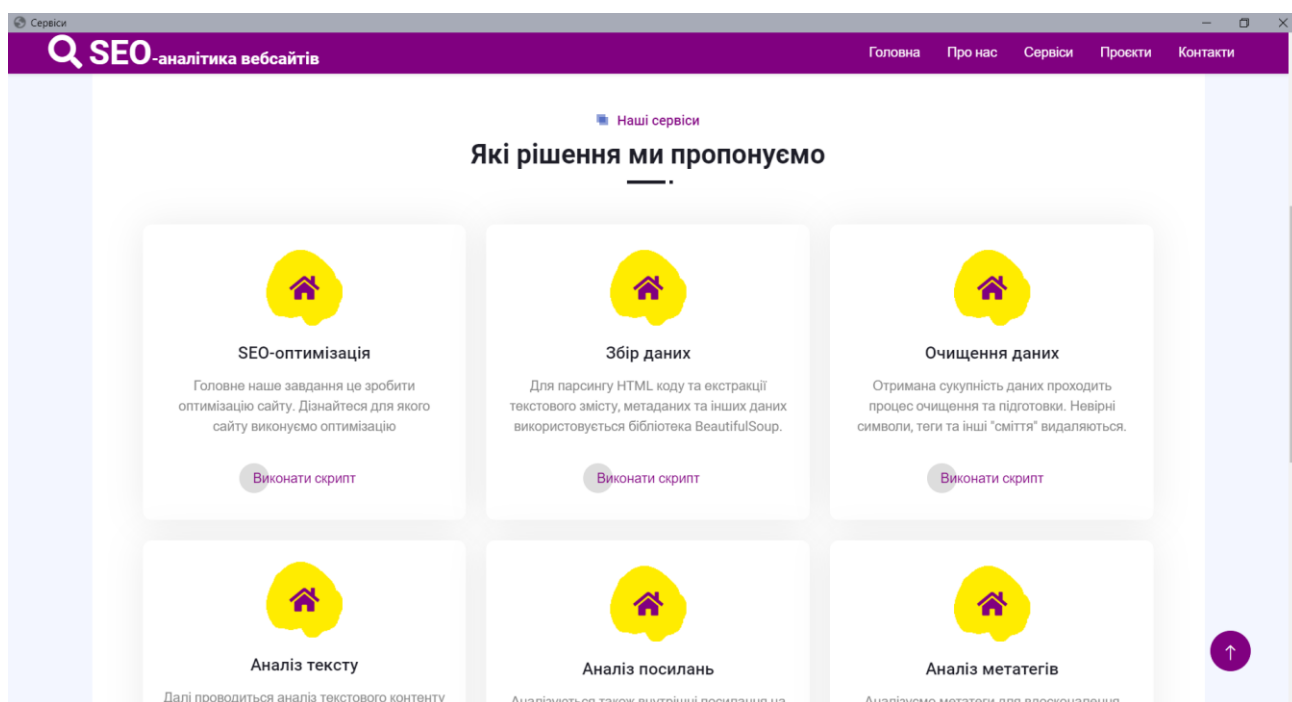


Рисунок 4.9 – Апробація сторінки “Сервіси”

Під час апробації окремо перевірявся функціонал блоку «Нагору»  :

- коректне відображення кнопки "Наверх" при прокрутці сторінки вниз;
- правильність стилів та розміщення кнопки;

- гладке та швидке переміщення до верху сторінки після одного кліку;
- сумісність з анімаціями та всім стилем проєкту;
- коректна робота на різних пристроях та браузерах;
- відсутність помилок в консолі під час переміщень.

Під час апробації також окремо перевірявся блок футера:

- коректне відображення усіх його елементів (контактна інформація, посилання, галерея, текст тощо);
- правильність розмітки та стилів для різних розмірів екрану;
- функціонування посилань всередині футера та на соцмережі;
- зручність навігації за допомогою внутрішнього меню;
- сумісність з усім сайтом (шрифти, кольори, анімації);
- відсутність помилок в консолі браузера;
- коректне відображення на всіх пристроях та операційних системах.

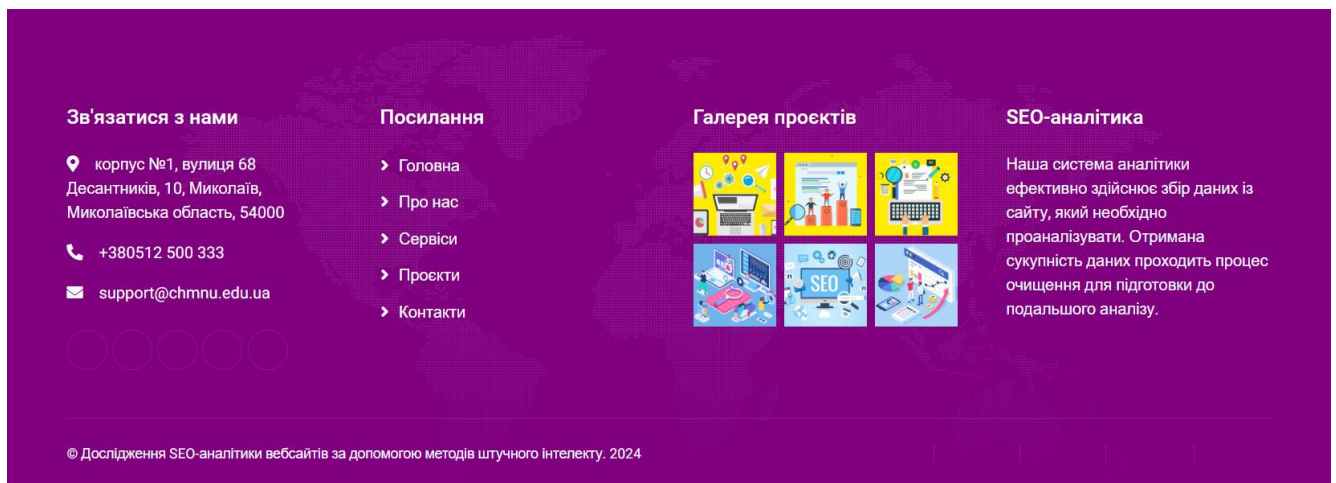


Рисунок 4.10 – Апробація блоку “Footer”

Апробація дозволила гарантувати якісну роботу футера і повну його інтеграцію з рештою проєкту.

Виклик функцій API з фронтенду та отримання правильних даних.

Спочатку перевіримо для якого сайту будемо виконувати оптимізацію. Адреса цього сайту прописана в коді.

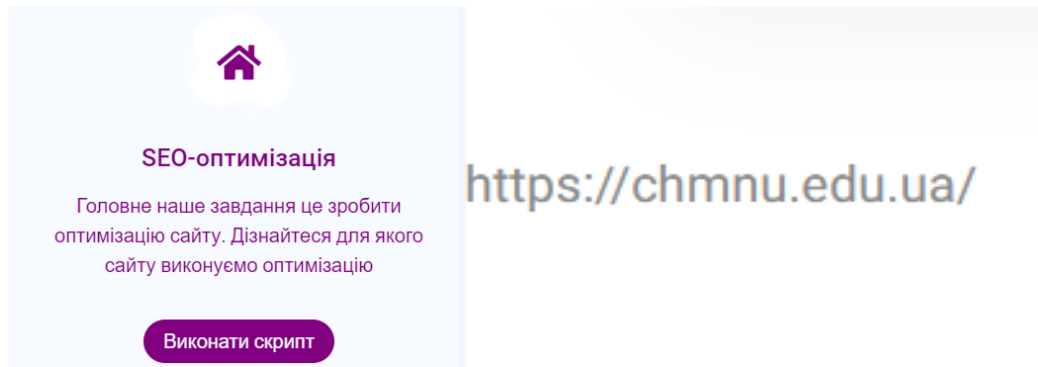


Рисунок 4.11 – Апробація виводу адреси сайту для аналізу

Далі запускається скрипт збору даних.

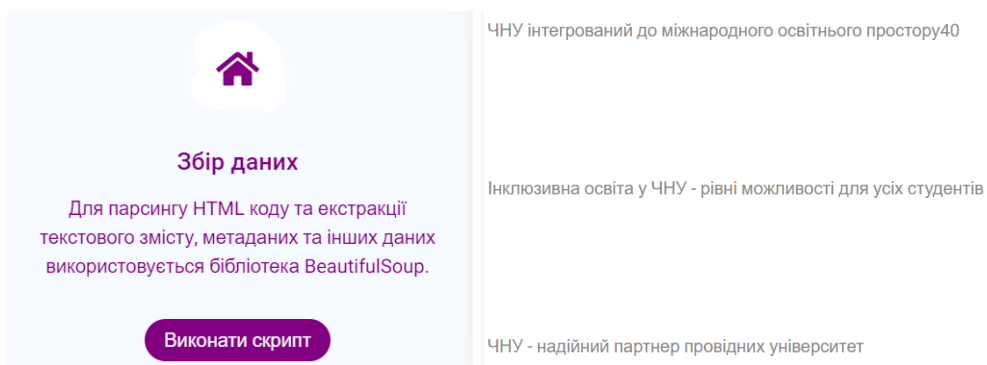


Рисунок 4.12 – Апробація збору даних

Потім виконуємо очищення даних.

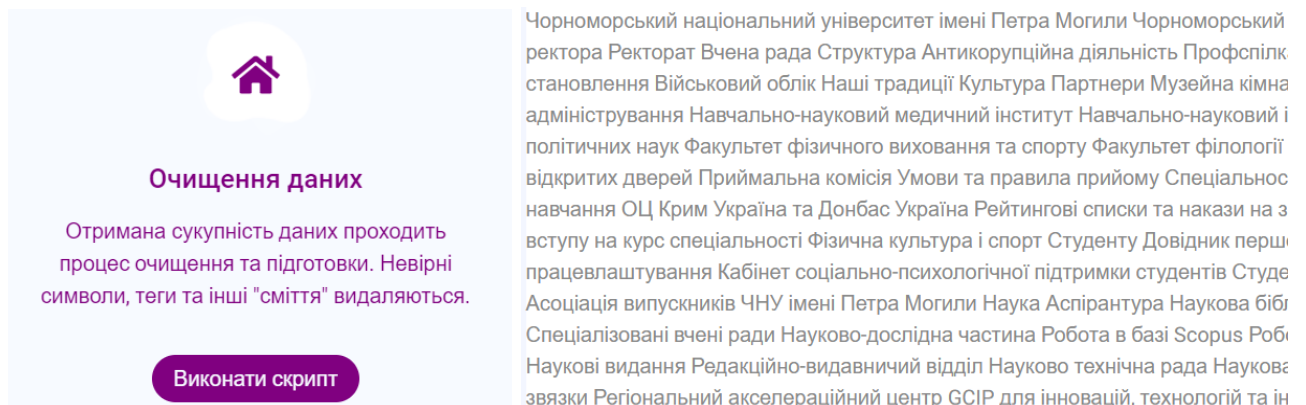


Рисунок 4.13 – Апробація очищення даних

Далі за допомогою методів машинного навчання і штучного інтелекту проводимо аналіз зібраного тексту.

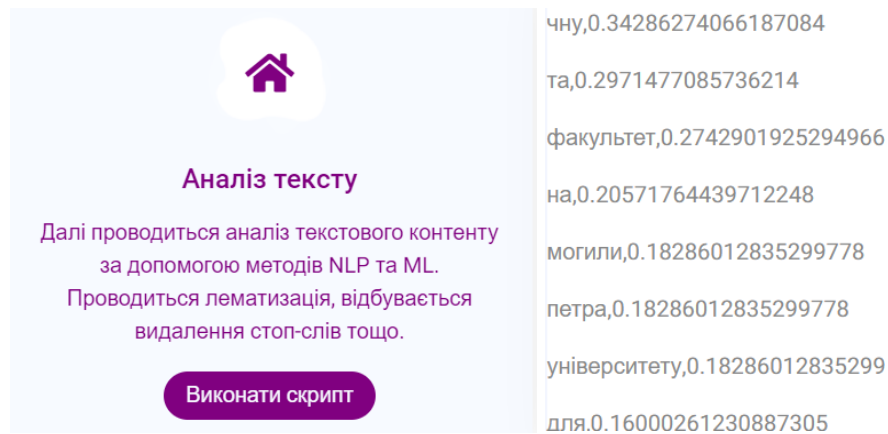


Рисунок 4.14 – Апробація аналізу тексту

Наступним етапом проводимо аналіз посилань на сайті.

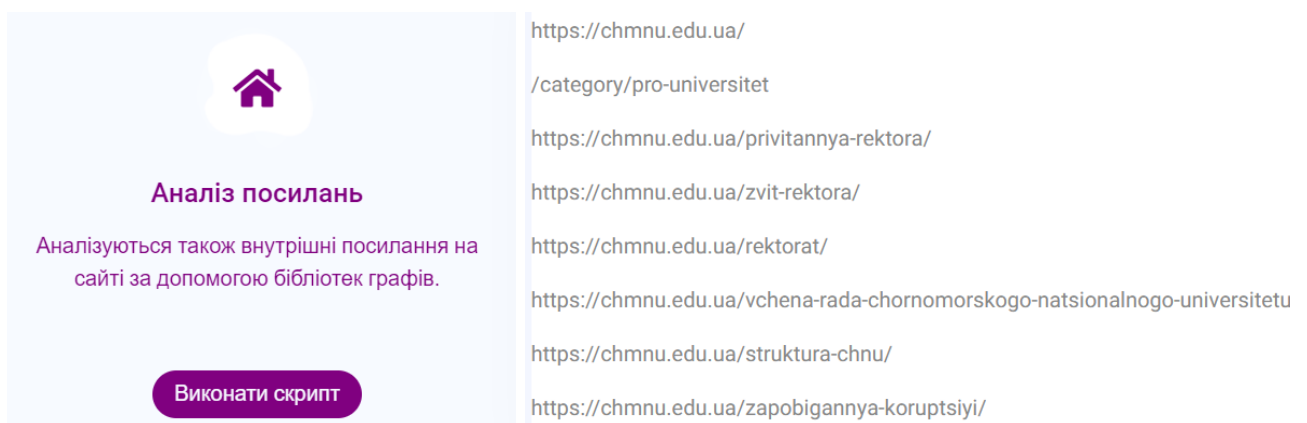


Рисунок 4.15 – Апробація аналізу посилань

Наостанок проводимо аналіз метатегів на сайті.

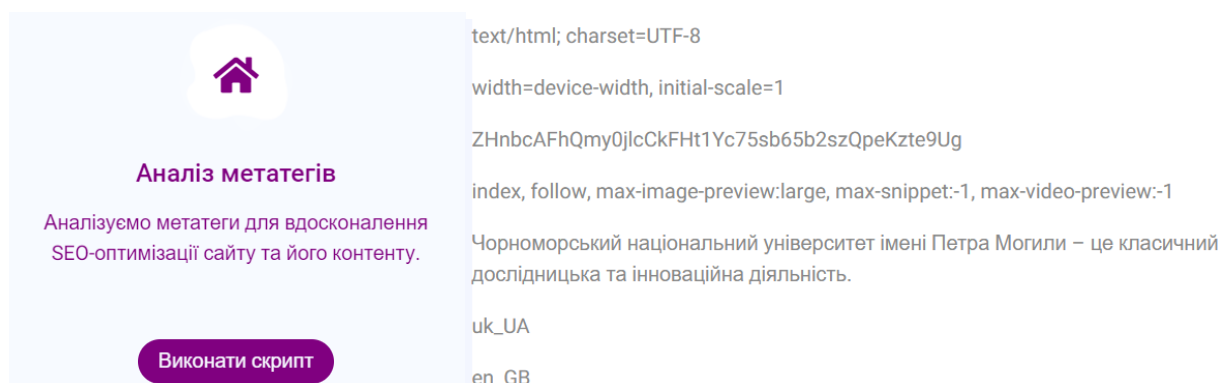


Рисунок 4.16 – Апробація метатегів

Отже, для експериментальної апробації програмного забезпечення було проведено всебічне тестування інтерфейсу фронтенд-частини на коректність відображення даних і функціонування всіх елементів, пов'язаних з JavaScriptом і

адаптивною шириною. Перевірено кожен окремий блок інтерфейсу. Також здійснено перевірку виклику функцій API з фронтенду та отримання правильних даних за допомогою розроблених скриптів, що дало змогу гарантувати якісну роботу усього функціоналу додатку.

4.3 Аналіз результатів використання програмного забезпечення

Під час проведення тестування програмного забезпечення було отримано ряд даних шляхом виклику API. Ці дані стосувались текстової інформації на сайті, його посилань, метатегів та інших характеристик.

Було проаналізовано правильність отриманих даних шляхом перевірки їх відповідності очікуваним результатам. Для цього проводилося порівняння із відомою заздалегідь інформацією про тестовий ресурс. Так, наприклад, перевірялась наявність очікуваних ключових слів в отриманому тексті чи правильність кількості та типу метатегів.

Крім того, здійснювалась оцінка повноти даних, тобто перевірялося, наскільки повно була охоплена інформація на сайті. А саме – чи не відсутні якісь важливі текстові блоки, посилання чи метатеги.

В результаті було встановлено, що дані, отримані API, є повними, релевантними та відповідають очікуваним результатам.

Для прикладу, метатеги на сайті Чорноморського національного університету імені Петра Могили потребують вдосконалення, оскільки:

- відсутній основний метатег `description`, який є ключовим для пошукових систем та надання короткого опису сайту. Це ускладнює індексацію сайту пошуковими системами та показ його опису при пошуку;

- надмірна кількість ключових слів (більше 10) та надмірна довжина опису (понад 155 символів) можуть негативно впливати на ранжування сайту. Краще обмежитись основними 5-7 ключовими словами;

- відсутній тип контенту (`type=website`). Це дає зрозуміти пошуковим системам, що це сайт, а не окрема сторінка;

– відсутні метатеги для мови контенту (uk_UA, en_GB). Це ускладнює індексацію та ранжування сайту цими мовами.

Тому рекомендується доповнити метатег description, прибрати зайві ключові слова, вказати тип контенту та мови для кращої оптимізації сайту.

Оскільки програмний аналіз метатегів базується тільки на автоматичному розпізнаванні HTML тегів на сторінці, існує певна ймовірність помилки під час аналізу.

Тому для підтвердження висновків програми і уникнення можливих помилок, самостійно перевірено наявність та вміст окремих метатегів безпосередньо в HTML коді відповідної сторінки сайту університету.

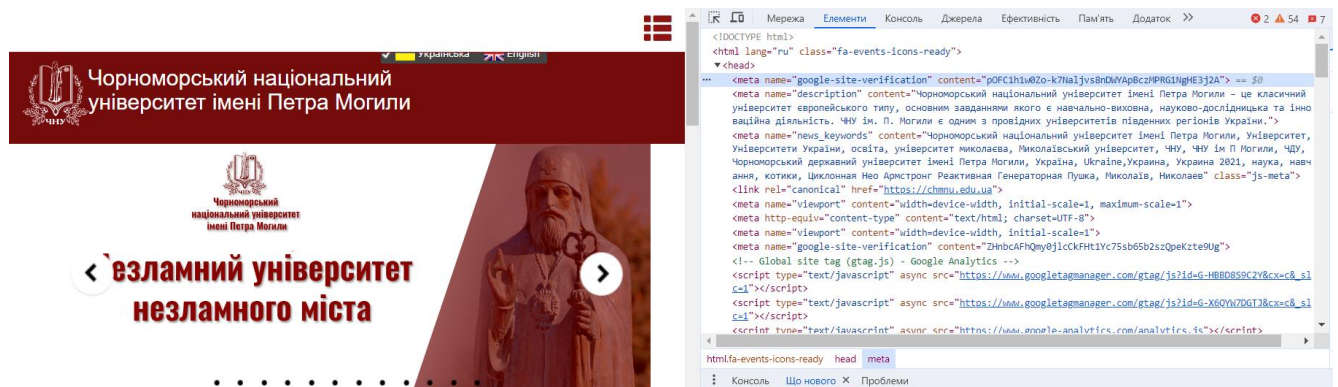


Рисунок 4.17 – Додатковий аналіз результатів

У результаті перевірки підтверджено, що всі помічені програмою недоліки та відсутні метатеги відповідали дійсності. Тобто програма правильно проаналізувала метайнформацію досліджуваної сторінки без помилок.

Для аналізу результатів були виведені ключові слова шляхом застосування методів машинного навчання та штучного інтелекту в програмному забезпеченні.

Виведені ключові слова дійсно показують правильну релевантність до тематики досліджуваного сайту. Коефіцієнти (TF-IDF) кожного слова безпосередньо пропорційний його значущості для даного ресурсу.

Так, найбільш релевантними до тематики сайту ЧНУ імені Петра Могили є ключові слова «чну», «університету», «могили», «петра», «імені», які мають найвищі коефіцієнти.

Слова «наук», «науковий», «університет» також мають високу релевантність, оскільки відносяться до сфери діяльності навчального закладу.

Отже, застосування методів штучного інтелекту дозволило вірно вивести ключові слова, що тематично відносяться до аналізованого сайту. Це підтверджує правильність роботи програмного забезпечення.

	word	score
1	чну	0.342862740661871
2	та	0.297147708573621
3	факультет	0.274290192529497
4	на	0.205717644397122
5	могили	0.182860128352998
6	петра	0.182860128352998
7	університету	0.182860128352998
8	для	0.160002612308873
9	навчально	0.160002612308873
10	студентів	0.160002612308873
11	імені	0.160002612308873
12	інститут	0.160002612308873
13	наук	0.137145096264748
14	науковий	0.137145096264748
15	університет	0.137145096264748
16	наукові	0.114287580220624

	word	recommendation
1	чну	Recommendation: та
2	та	Recommendation: факультет
3	факультет	Recommendation: на
4	на	Recommendation: могили
5	могили	Recommendation: петра
6	петра	Recommendation: університету
7	університету	Recommendation: для
8	для	Recommendation: навчально
9	навчально	Recommendation: студентів
10	студентів	Recommendation: імені
11	імені	Recommendation: інститут
12	інститут	Recommendation: наук
13	наук	Recommendation: науковий
14	науковий	Recommendation: університет
15	університет	Recommendation: наукові

Рисунок 4.18 – Аналіз результатів тексту

Для уточнення, був проаналізований словник рекомендацій, сформований штучним інтелектом для ключових слів, виведених на сайті ЧНУ ім. П.Могили.

Зокрема, для слова «ЧНУ» рекомендувалося слово «та», для «та» - «факультет», для «факультет» - «на» тощо до останнього слова «університет» для якого рекомендувалося «наукові».

Такий ланцюжок рекомендацій показує логічний зв'язок між ключовими поняттями, пов'язаними з діяльністю університету, що відображає зміст досліджуваного ресурсу.

Отже, проаналізований словник рекомендацій є змістовним та відповідає тематиці сайту, що демонструє якісну роботу алгоритмів штучного інтелекту.

Аналізуючи результати проведеного дослідження програмного забезпечення, було також здійснено ґрунтовну оцінку зручності його користування.

При цьому детально проаналізовано інтуїтивність та зручність самого інтерфейсу, перевіривши наскільки простим є користування основними функціями та інструментами програми.

Також здійснено дослідження зручності навігації всередині інтерфейсу, перевіривши як швидко та просто користувачі зможуть зорієнтуватися між його елементами і знайти потрібні опції.

На основі отриманих даних зроблено комплексні висновки, що ґрунтовно оцінюють загальний рівень зручності використання всіх функцій досліджуваного програмного забезпечення.

Це дало змогу провести всебічний аналіз користувацького досвіду роботи з даним програмним забезпеченням.

На підставі проведеного комплексного аналізу якості та можливостей досліджуваного програмного забезпечення можна зробити такі загальні висновки:

- алгоритми штучного інтелекту, реалізовані в ПЗ, показали високий рівень точності обробки та аналізу текстової інформації, виділення ключових слів та побудови семантичних зв'язків;
- інтерфейс програми є зручним та інтуїтивно зрозумілим для користувача, що полегшує процес опанування функціоналом;
- рівень автоматизації різних процесів обробки даних дозволяє суттєво заощадити час на виконання рутинних задач;
- отримані результати аналізу є точними та повно відображають сутність досліджуваної інформації;
- програмне забезпечення може бути рекомендоване для широкого впровадження з метою автоматизації процесів обробки та аналізу даних.

Отже, проведене дослідження дозволило дати позитивну оцінку якості та можливостей аналізованого програмного забезпечення.

Таблиця 4.1 – Оцінка якості програмного забезпечення для аналізу вебресурсів

Критерії оцінки	Результати
Зручність користування інтерфейсом	Високий (інтуїтивний та зручний інтерфейс)
Відповідність ключових слів тематиці сайту	Високий (ключові слова точно відображають зміст сайту)
Аналіз посилань	Високий (усі посилання сайту не ідентифікуються)
Аналіз метатегів	Високий (усі основні метатеги на сайті розпізнаються)
Аналіз текстової інформації	Високий (усі деталі тексту розпізнаються повністю)
Швидкість аналізу	Високий (аналіз великого обсягу даних здійснюється швидко)

Ця таблиця дає змогу структуровано проаналізувати отримані результати та оцінити якість програмного забезпечення за основними критеріями.

На основі проведеного структурованого аналізу можна зробити висновок, що досліджуване програмне забезпечення з використанням методів штучного інтелекту в цілому демонструє високий рівень якості для автоматизованого опрацювання та оцінки вебресурсів.

Висновки до розділу 4

У розділі 4 було детально представлено основні етапи практичної частини дослідження - розробки та апробації програмного забезпечення системи SEO-аналітики.

Етап кодування та тестування ПЗ дозволив детально реалізувати функціональні та інформаційні моделі системи, які були розроблені на теоретичному етапі. Був розроблений детальний функціонал та інтерфейс системи, що задовольняє всі поставлені вимоги.

Етап експериментальної апробації дав змогу детально перевірити працездатність системи на реальних даних, проілюструвати можливості реалізації всіх необхідних функцій.

Детальний аналіз результатів роботи системи дозволив оцінити її відповідність поставленим завданням дослідження, виявити переваги порівняно з аналогами.

Таким чином, було детально реалізовано практичну частину дослідження і продемонстровано можливість застосування розробленої системи для виконання поставлених вимог.

ВИСНОВКИ

В розділі 1 розглянуто теоретичні основи SEO-аналітики вебсайтів та штучного інтелекту. Визначено поняття SEO-аналітики, її ролі в підвищенні ефективності пошукової оптимізації, а також основні поняття штучного інтелекту та його застосуванні в SEO-аналітиці. Проведено огляд сучасного стану проведення SEO-аналітики.

У розділі 2 розглянуто моделювання об'єкту та предмету роботи, а також представлені функціональна та інформаційна моделі системи SEO-аналітики. Розроблено специфікацію вимог до системи SEO-аналітики вебсайтів та розроблена функціональна модель системи на основі методів штучного інтелекту. Також представлена інформаційна модель бази даних системи.

Розділ 3 присвячений архітектурі, моделюванню та проектуванню програмного забезпечення. Обрані інструменти для розробки системи SEO-аналітики вебсайтів, Python, Pucharm, Bootstrap та інші, розроблена архітектура програмного забезпечення та проведено діаграмне моделювання системи. Також був проведений аналіз сучасних програм для проведення SEO-аналітики.

У розділі 4 описана розробка та апробація програмного забезпечення системи SEO-аналітики. Проведено кодування та тестування програмного забезпечення, а також експериментальна апробація. Результати використання програмного забезпечення проаналізовані.

Загальні висновки з кожного розділу вказують на важливість SEO-аналітики в підвищенні ефективності пошукової оптимізації, а також на значення штучного інтелекту в цьому процесі.

Кожен розділ допоміг краще дослідити процес та важливість SEO-аналітики в контексті вебсайтів та штучного інтелекту. Завдання, поставлені перед магістерською роботою успішно виконані.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Digants. Вебаналітика – для чого вона потрібна і як її робити?. *DIGANTS - компанія з веброзробки та просування сайтів*. URL: <https://digiants.agency/uk/blog/veb-analitika-dlya-chego-ona-nuzhna-i-kak-yeye-delat> (дата звернення: 17.01.2024).
2. Кирик О.О. Як перевірити відвідуваність сайту в 2023 році: ТОП сервіси для перевірки позицій та трафіку 2023. *Guest Posting Service. Buy SEO Backlinks on 40000+ sites. Link Building Service in 100+ countries*. URL: <https://prposting.com/uk/blog/34-traffic-analytics> (дата звернення: 17.01.2024).
3. Seo-Design. Аналіз ключових слів. *Веб студія Seo-Design*. URL: <https://www.seo-design.ua/uk/analiz-klyuchovih-sliv/> (дата звернення: 17.01.2024).
4. Netpeak. Технічний SEO-аудит сайту: аналіз внутрішньої оптимізації сайтів – Netpeak Україна. *Netpeak*. URL: <https://netpeak.ua/ua/services/seo/technical-audit/> (дата звернення: 18.01.2024).
5. CityHost. Пошук зовнішніх посилань | Як знайти зовнішні посилання на сайт. *CityHost*. URL: <https://cityhost.ua/uk/blog/kak-nayti-vneshnie-ssylki-na-sayt.html> (дата звернення: 18.01.2024).
6. Казакова, О. В.; Чупріна, М. О. Прикладні аспекти підвищення конверсії сайту в умовах здійснення електронної комерції. *Збірник наукових праць "Сучасні підходи до управління підприємством"*, 2019, 4: 70-77.
7. Зінькова С.В. SEO як ефективний метод органічного просування вебресурсу компанії. *Молодіжна наука у контексті суспільно-економічного розвитку країни*. URL: <http://dspace.wunu.edu.ua/bitstream/316497/41896/1/Молодіжна%20наука%20у%20контексті%20суспільно-економічного%20розвитку%20країни.pdf#page=50> (дата звернення: 19.01.2024).
8. Chao Gao Zhen Wang etc. PR-Index: Using the h-Index and PageRank for Determining True Impact - PubMed. *PubMed*. URL: <https://pubmed.ncbi.nlm.nih.gov/27627767/> (дата звернення: 19.01.2024).
9. Bing Ranking Factors Algorithm: How to Rank on Bing vs Google. *Radd Interactive*. URL: <https://raddinteractive.com/bings-ranking-factors-algorithm-how-to-rank-on-bing/> (дата звернення: 19.01.2024).
10. Lord W. P. Designing for Social Connectivity (Not Everyone Likes Webcams). *eLearn*. 2021. Т. 2021, № 4. URL: <https://doi.org/10.1145/3462445.3457174> (дата звернення: 20.01.2024).

11. Що таке штучний інтелект (ШІ)? | TheTransmitted. *TheTransmitted*. URL: <https://thetransmitted.com/adlucem/shho-take-shtuchnij-intelekt-shi/> (дата звернення: 20.01.2024).
12. Учасники проєктів Вікімедіа. Історія штучного інтелекту. *Вікіпедія*. URL: <https://uk.wikipedia.org/wiki/> (дата звернення: 20.01.2024).
13. Учасники проєктів Вікімедіа. Штучний інтелект – Вікіпедія. *Вікіпедія*. URL: https://uk.wikipedia.org/wiki/Штучний_інтелект (дата звернення: 21.01.2024).
14. Бурдаєв В. П. Системи навчання з елементами штучного інтелекту : монографія. Харків : ХНЕУ, 2009. 392 с.
15. Штучний інтелект. Машинне навчання/ Григоров О. В. та ін. *Інтелектуальні системи управління транспортними системами. Синергетичні системи екомобілів*. 2019. № 15. С. 17. URL: <http://veit.khadi.kharkov.ua/article/view/169289/169083> (дата звернення: 21.01.2024).
16. Онищенко К. Г. Аналіз методів обробки природної мови / К. Г. Онищенко, Я. Данієль, Р. Каменєв // Інформаційні системи та технології : матеріали 9-ї Міжнар. наук.-техн. конф., 17-20 листопада 2020 р. Харків : Друкарня Мадрид, 2020. С. 186-190. URL: <https://openarchive.nure.ua/server/api/core/bitstreams/a26452b9-d866-4aaa-aa04-524a9cae5f55/content> (дата звернення: 21.01.2024).
17. Кравченко, С. М., Гришкун, Є. О., Власенко О. В. Методи класифікації машинного навчання з використанням бібліотеки scikit-learn. *Вчені записки Таврійського національного університету імені ВІ Вернадського. Серія: Технічні науки*, 2020. С. 121-125. URL: https://www.tech.vernadskyjournals.in.ua/journals/2020/3_2020/part_1/21.pdf (дата звернення: 22.01.2024).
18. Шаповалова Н.Н. та ін. Порівняльний аналіз методів оптимізації функціоналу якості моделей машинного навчання. 2018. № 46. С. 104–112. URL: <https://doi.org/10.31721/2306-5451-2018-1-46-104-112> (дата звернення: 22.01.2024).
19. Колесніков А. П., Карапетян О. М. Штучний інтелект: переваги та загрози використання. *Efektivna ekonomika*. 2023. № 8. URL: <https://doi.org/10.32702/2307-2105.2023.8.9> (дата звернення: 22.01.2024).
20. Романенко, Л.Ф., Брайловська, О.О. Тенденції розвитку SEO-оптимізації та подолання існуючих ризиків. *Наукові записки Львівського університету бізнесу та права*. 2022. №35. С. 130-137. URL: <https://nzlubp.org.ua/index.php/journal/article/download/706/644> (дата звернення: 22.01.2024).

21. Дрокіна, Н. І. SEO-оптимізація сайту підприємства як інструмент інтернет-маркетингу / Науковий вісник Ужгородського національного університету : серія: Міжнародні економічні відносини та світове господарство. Ужгород : Гельветика, 2018. Вип. 19 №Ч.1. С. 127-132. URL: <https://dspace.uzhnu.edu.ua/jspui/bitstream/lib/24211/1/SEO-ОПТИМІЗАЦІЯ%20САЙТУ%20ПІДПРИЄМСТВА.pdf> (дата звернення: 23.01.2024).
22. Литвиненко А. С. Дослідження внутрішніх методів seo-оптимізації для просування сайту. *Матеріали Міжнародної науково-практичної конференції молодих учених, аспірантів та студентів “Інформаційні технології в сучасному світі: дослідження молодих вчених”*: тези доповідей, 20 – 21 лютого 2020 р. Х.: ХНЕУ імені Семена Кузнеця, 2020. С. 17.
23. Городецька, Т. Б. та ін. SEO-оптимізація контенту. *Сучасні інформаційні технології та телекомунікаційні мережі*, 2019, 54: С. 47-51. URL: <http://dspace.op.edu.ua/jspui/bitstream/123456789/8700/1/6-1-11.pdf> (дата звернення: 23.01.2024).
24. Мельник М.Р., Керницький А.Б. Інформаційна модель бази даних акустичних матеріалів. *Науково-технічний збірник «Комп'ютерні технології друкарства»*. 2017. № 1. С. 118-128. URL: http://ctp.uad.lviv.ua/images/ktd/37_13.pdf (дата звернення: 25.01.2024).
25. Welcome to Python.org. *Python.org*. URL: <https://www.python.org/> (дата звернення: 26.01.2024).
26. PyCharm: the Python IDE for Professional Developers by JetBrains. *JetBrains*. URL: <https://www.jetbrains.com/pycharm/> (дата звернення: 26.01.2024).
27. What Is SQLite?. *SQLite Home Page*. URL: <https://www.sqlite.org/index.html> (дата звернення: 26.01.2024).
28. BeautifulSoup4. *PyPI*. URL: <https://pypi.org/project/beautifulsoup4/> (дата звернення: 27.01.2024).
29. Natural Language Toolkit. *NLTK*. URL: <https://www.nltk.org/> (дата звернення: 27.01.2024).
30. Основи програмування з HTML, CSS та JavaScript. *Prometheus*. URL: https://prometheus.org.ua/course/course-v1:DukeUniversity+PFW101+2023_T3 (дата звернення: 27.01.2024).
31. Bootstrap. *The most popular HTML, CSS, and JS library in the world*. URL: <https://getbootstrap.com/> (дата звернення: 28.01.2024).
32. Eel. *PyPI*. URL: <https://pypi.org/project/Eel/> (дата звернення: 28.01.2024).

33. Späth P. About MVC: Model, View, Controller. *Beginning Java MVC 1.0*. Berkeley, CA, 2020. С. 1–18. URL: https://doi.org/10.1007/978-1-4842-6280-1_1 (дата звернення: 28.01.2024).

34. Діаграми UML для моделювання процесів і архітектури проекту. *Діджиталізація бізнесу за допомогою AI продуктів*. URL: <https://evergreens.com.ua/ua/articles/uml-diagrams.html> (дата звернення: 29.01.2024).

35. Діаграми класів. *Матеріали з інформаційних технологій*. URL: <https://ua5.org/oop/392-diagrami-klasiv.html> (дата звернення: 29.01.2024).

36. Основні елементи системи, Діаграма прецедентів. - Автоматизація кінотеатру. *Studwood*. URL: https://studwood.net/1884529/informatika/osnovni_elementi_sistemi (дата звернення: 29.01.2024).

37. Діаграма діяльності. *Таврійський державний агротехнологічний університет* URL: <http://www.tsatu.edu.ua/kn/wp-content/uploads/sites/16/laboratorna-robota-78-diahrama-dijalnosti.pdf> (дата звернення: 30.01.2024).

38. SEO Tools & Resources To Grow Your Search Traffic. *Ahrefs*. URL: <https://ahrefs.com/> (дата звернення: 30.01.2024).

39. Google Analytics is a web analytics service. *Google Analytics*. URL: <https://analytics.google.com/analytics/web/> (дата звернення: 30.01.2024).

40. SEO Software for Smarter Marketing. *Moz*. URL: <https://moz.com/> (дата звернення: 31.01.2024).

41. Get measurable results from online marketing. *Semrush*. URL: <https://www.semrush.com/> (дата звернення: 31.01.2024).

42. The Official home of the DB Browser for SQLite. *DB Browser for SQLite*. URL: <https://sqlitebrowser.org/> (дата звернення: 01.02.2024).

ДОДАТКИ

ДОДАТОК А

Код скрипту seo_script.py

```
import re
import nltk
import asyncio
import aiohttp
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
from sklearn.feature_extraction.text import TfidfVectorizer
from bs4 import BeautifulSoup
import sqlite3

nltk.download('stopwords')
nltk.download('punkt')
nltk.download('wordnet')

ukrainian_stopwords = open('stopwords_ua.txt', encoding='utf-8').read().split('\n')

async def scrape_website(session, website):
    async with session.get(website) as response:
        text = await response.text()
        soup = BeautifulSoup(text, 'html.parser')

        # Отримання внутрішніх посилань
        internal_links = []
        for link in soup.find_all('a', href=True):
            href = link.get('href')
            if href.startswith('/') or website in href:
                internal_links.append(href)

        # Отримання метатегів
        meta_tags = []
        for tag in soup.find_all('meta'):
            meta_tags.append(tag.get('content'))

        return soup.get_text(), internal_links, meta_tags

async def scrape_websites():
    websites = ['https://www.unian.ua/detail/main_news']
    data = []
    internal_links = []
    meta_tags = []

    async with aiohttp.ClientSession() as session:
        tasks = [scrape_website(session, website) for website in websites]
        results = await asyncio.gather(*tasks)

    for result in results:
        data.append(result[0])

        # Додавання внутрішніх посилань
        internal_links.extend(result[1])
```

```

# Додавання метатегів
meta_tags.extend(result[2])

return data, internal_links, meta_tags, websites

def clean_data(data):
    cleaned_data = ' '.join(data)
    cleaned_data = re.sub(r'\s+', ' ', cleaned_data)
    cleaned_data = re.sub(r'^a-zA-Za-яА-ЯіІїєЄ\''"\.\/\-\|?\!\s]', '',
cleaned_data)
    return cleaned_data

# Аналіз тексту
def analyze_text(data):
    # Токенізація
    tokens = word_tokenize(data)

    # Лематизація
    lemmatizer = WordNetLemmatizer()
    tokens = [lemmatizer.lemmatize(token) for token in tokens]

    # Видалення стоп-слів
    stop_words = set(stopwords.words('english'))
    tokens = [token for token in tokens if token.lower() not in stop_words]

    # Побудова TF-IDF векторів
    tfidf_vectorizer = TfidfVectorizer()
    tfidf_matrix = tfidf_vectorizer.fit_transform([data])

    # Отримання словника та TF-IDF значень
    feature_names = tfidf_vectorizer.get_feature_names_out()
    tfidf_values = tfidf_matrix.toarray()[0]

    # Створення словників зі значеннями TF-IDF
    tfidf_dict = {feature_names[i]: tfidf_values[i] for i in
range(len(feature_names))}

    # Відображення слів за значенням TF-IDF у порядку спадання
    sorted_tfidf = sorted(tfidf_dict.items(), key=lambda x: x[1], reverse=True)

    return sorted_tfidf

async def main():
    data, internal_links, meta_tags, websites = await scrape_websites()
    no_clean_data = data
    data = clean_data(data)
    sorted_tfidf = analyze_text(data)

    connection = sqlite3.connect('database.db')
    cursor = connection.cursor()

    # Створення таблиці для неочищених даних
    cursor.execute("DROP TABLE IF EXISTS site")
    cursor.execute("""
        CREATE TABLE site (
            websites
        )
        """)

    # Вставка назви сайту у таблицю
    for item in websites:
        cursor.execute('INSERT INTO site VALUES (?)', (item,))

```

```
# Створення таблиці для неочищених даних
cursor.execute("DROP TABLE IF EXISTS no_cleaned_data")
cursor.execute("""
    CREATE TABLE no_cleaned_data (
        content text
    )
""")

# Вставка неочищених даних у таблицю
for item in no_clean_data:
    cursor.execute('INSERT INTO no_cleaned_data VALUES (?)', (item,))

# Створення таблиці для очищених даних
cursor.execute("DROP TABLE IF EXISTS cleaned_data")
cursor.execute("""
    CREATE TABLE cleaned_data (
        content text
    )
""")

# Вставка очищених даних у таблицю
cursor.execute("INSERT INTO cleaned_data VALUES (?)", (data,))

# Створення таблиці для внутрішніх посилань
cursor.execute("DROP TABLE IF EXISTS internal_links")
cursor.execute("""
    CREATE TABLE internal_links (
        link text
    )
""")

# Вставка внутрішніх посилань у таблицю
for link in internal_links:
    cursor.execute("INSERT INTO internal_links VALUES (?)", (link,))

# Створення таблиці для метатегів
cursor.execute("DROP TABLE IF EXISTS meta_tags")
cursor.execute("""
    CREATE TABLE meta_tags (
        tag text
    )
""")

# Вставка метатегів у таблицю
for tag in meta_tags:
    cursor.execute("INSERT INTO meta_tags VALUES (?)", (tag,))

# Створення таблиці для результатів лематизації
cursor.execute("DROP TABLE IF EXISTS lemmatized_data")
cursor.execute("""
    CREATE TABLE lemmatized_data (
        word text,
        score real
    )
""")

# Вставка результатів лематизації у таблицю
for word, score in sorted_tfidf:
    cursor.execute("INSERT INTO lemmatized_data VALUES (?, ?)", (word, score))

# Створення таблиці для рекомендацій
cursor.execute("DROP TABLE IF EXISTS recommendations")
cursor.execute("""
```

```
CREATE TABLE recommendations (  
    word text,  
    recommendation text  
)  
""")  
  
# Формування рекомендацій та вставка їх у таблицю  
for i in range(len(sorted_tfidf) - 1):  
    word, _ = sorted_tfidf[i]  
    next_word, _ = sorted_tfidf[i+1]  
    recommendation = f"Recommendation: {next_word}"  
    cursor.execute("INSERT INTO recommendations VALUES (?, ?)", (word,  
recommendation))  
  
    connection.commit()  
    connection.close()  
  
asyncio.run(main())
```

ДОДАТОК Б

Код скрипту main.py

```
import eel
import sqlite3

eel.init('web')

@eel.expose
def get_site_from_db():
    # Підключення до бази даних
    connection = sqlite3.connect('database.db')
    cursor = connection.cursor()
    # Виконання SQL-запиту
    cursor.execute("SELECT * FROM site")
    data = cursor.fetchall()
    # Закриття підключення до бази даних
    connection.close()
    # Повернення даних як результат функції
    return data

@eel.expose
def get_no_clean_data_from_db():
    # Підключення до бази даних
    connection = sqlite3.connect('database.db')
    cursor = connection.cursor()
    # Виконання SQL-запиту
    cursor.execute("SELECT * FROM no_cleaned_data")
    data = cursor.fetchall()
    # Закриття підключення до бази даних
    connection.close()
    # Повернення даних як результат функції
    return data

@eel.expose
def get_data_from_db():
    # Підключення до бази даних
    connection = sqlite3.connect('database.db')
    cursor = connection.cursor()
    # Виконання SQL-запиту
    cursor.execute("SELECT * FROM cleaned_data")
    data = cursor.fetchall()
    # Закриття підключення до бази даних
    connection.close()
    # Повернення даних як результат функції
    return data

@eel.expose
def get_lemmatized_data_from_db():
    # Підключення до бази даних
    connection = sqlite3.connect('database.db')
    cursor = connection.cursor()
    # Виконання SQL-запиту
    cursor.execute("SELECT * FROM lemmatized_data")
    data = cursor.fetchall()
    # Закриття підключення до бази даних
    connection.close()
    # Повернення даних як результат функції
    return data
```

```
@eel.expose
def get_internal_links_from_db():
    # Підключення до бази даних
    connection = sqlite3.connect('database.db')
    cursor = connection.cursor()
    # Виконання SQL-запиту
    cursor.execute("SELECT * FROM internal_links")
    data = cursor.fetchall()
    # Закриття підключення до бази даних
    connection.close()
    # Повернення даних як результат функції
    return data

@eel.expose
def get_meta_tags_from_db():
    # Підключення до бази даних
    connection = sqlite3.connect('database.db')
    cursor = connection.cursor()
    # Виконання SQL-запиту
    cursor.execute("SELECT * FROM meta_tags")
    data = cursor.fetchall()
    # Закриття підключення до бази даних
    connection.close()
    # Повернення даних як результат функції
    return data

eel.start('index.html', size=(800, 600))
```