

Міністерство освіти і науки України
Дрогобицький державний педагогічний університет імені Івана Франка
кафедра фізики та інформаційних систем

«До захисту допускаю»

завідувач кафедри фізики та інформаційних систем

кандидат фізико-математичних наук, доцент

_____ Віталій ГОЛЬСЬКИЙ «____» ____ 2025 р.

Розроблення хром-плагіна для пошуку помилок на веб-сторінці

Спеціальність: 122 Комп'ютерні науки

Випускова робота

на здобуття кваліфікації – «Бакалавр з комп'ютерних наук»

Автор роботи: **Кенц Анатолій Васильович** _____

підпис

Науковий керівник: **кандидат фізико-математичних наук,
доцент кафедри фізики та інформаційних систем**

Лешко Роман Ярославович _____

підпис

Дрогобич 2026

Захист кваліфікаційної (бакалаврської, магістерської) роботи

(тема роботи)

Оцінка за стобальною шкалою: _____

Оцінка за національною чотирибальною шкалою: _____

Коротка мотивація захисту:

Дата

прізвище, ім'я

Голова ЕК _____

підпис

Секретар ЕК _____

підпис

прізвище, ім'я

Анотація

Кенц А. В. Розроблення хром-плагіна для пошуку помилок на веб-сторінці.
Випускова робота, Дрогобицький державний педагогічний університет імені Івана Франка, Дрогобич, 2026 р.

Випускова робота присвячена актуальній науково-технічній задачі автоматизації комплексного технічного аудиту та SEO-моніторингу веб-сторінок безпосередньо у браузерному середовищі. Об'єктом дослідження є методи ідентифікації критичних помилок, аналізу продуктивності та оцінки якості контенту сучасних веб-додатків.

У ході дослідження проєктується та реалізується багатofункціональне розширення для браузера Google Chrome на базі архітектури Manifest V3 із використанням стека React та TypeScript. Розроблений програмний засіб забезпечує виконання автоматизованої перевірки метаданих, ієрархії заголовків та індексаційних директив, виявлення mixed content, «битих» посилань та відсутніх атрибутів доступності, збір метрик швидкодії через Performance API та ідентифікація маркетингових пікселів, розрахунок щільності ключових слів, «водності» тексту та загального Content Score.

Цінність роботи полягає у розробці автономної системи розрахунку інтегральної оцінки якості (scoring logic) та модуля локальної генерації деталізованих PDF-звітів. Створений інструмент дозволяє фахівцям проводити оперативну експертизу веб-ресурсів, оптимізуючи процеси QA-тестування та пошукового просування без залучення сторонніх серверних обчислень.

Ключові слова: веб-сторінка, Chrome-плагін, Manifest V3, SEO-аудит, продуктивність, технічна оптимізація, юзабіліті.

Abstract

Kents A. V. Development of a Chrome Extension for Detecting Errors on a Web Page. Graduation Thesis, Ivan Franko State Pedagogical University in Drohobych, Drohobych, 2026.

The graduation thesis is dedicated to the urgent scientific and technical problem of automating comprehensive technical audits and SEO monitoring of web pages directly within the browser environment. The object of the study is the methods for identifying critical errors, analyzing performance, and assessing content quality in modern web applications.

In the course of the research, a multifunctional extension for the Google Chrome browser is designed and implemented based on the Manifest V3 architecture using the React and TypeScript stack. The developed software tool ensures the execution of automated checks for metadata, heading hierarchy, and indexing directives; detection of mixed content, broken links, and missing accessibility attributes; collection of performance metrics via the Performance API; identification of marketing pixels; and calculation of keyword density, text text-to-padding ratio ("wateriness"), and the overall Content Score.

The value of the work lies in the development of an autonomous system for calculating an integral quality score (scoring logic) and a module for the local generation of detailed PDF reports. The created tool enables specialists to conduct rapid examinations of web resources, optimizing QA testing and search engine optimization processes without involving third-party server-side computations.

Keywords: web page, Chrome extension, Manifest V3, SEO audit, performance, technical optimization, usability.

Зміст

ВСТУП.....	6
Розділ 1 АНАЛІТИЧНИЙ ОГЛЯД ТА ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ.....	10
1.1. Сучасний стан та архітектурні особливості складних веб-систем.....	11
1.2. Класифікація детермінованих дефектів та їх вплив на конверсію веб- ресурсів.....	13
1.3. Компаративний аналіз існуючих інструментів технічного та SEO-аудиту	15
1.4. Обґрунтування доцільності розробки автономного браузерного розширення на базі.....	17
Розділ 2 ПРОЄКТУВАННЯ ТА ТЕОРЕТИЧНЕ ОБҐРУНТУВАННЯ СИСТЕМИ	19
2.1. Формування функціональних та технічних вимог до інтегрованого засобу аудиту.....	20
2.2. Обґрунтування вибору технологічного стека.....	23
2.3. Математична модель інтегрального оцінювання якості веб-сторінки.....	25
2.4. Алгоритмічне забезпечення семантичного аналізу та технічної діагностики.....	27
Розділ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ЕКСПЛУАТАЦІЙНІ ХАРАКТЕРИСТИКИ РОЗРОБЛЕНОЇ СИСТЕМИ.....	29
3.1. Особливості програмної реалізації модульної архітектури на базі стандарту Manifest V3.....	29
3.2. Розроблення адаптивного користувацького інтерфейсу та засобів інтерактивної візуалізації даних.....	30
3.3. Функціональна реалізація діагностичних модулів та механізмів обміну повідомленнями.....	34
3.4. Технологія локальної генерації аналітичної звітності та верифікація математичної моделі оцінювання.....	36
Висновки.....	38
Список використаних джерел.....	40

ВСТУП

Актуальність теми. Сучасний етап розвитку глобального інформаційного суспільства характеризується тотальною диджиталізацією економічних та комунікаційних процесів. Веб-додатки еволюціонували від статичних інформаційних сторінок до надскладних інтерактивних систем, що інтегрують у собі розгалужену бізнес-логіку, клієнтські сценарії та багат шарові структури даних. У контексті такої технологічної трансформації якість веб-ресурсу перестає бути суто технічним параметром і стає критичним фактором ринкової конкурентоспроможності.

Актуальність дослідження обумовлена декількома фундаментальними чинниками. По-перше, кореляція між технічною досконалістю сайту та користувацьким досвідом (UX) набула характеру прямої залежності. Наявність mixed content, порушення ієрархії заголовків або відсутність атрибутів доступності (alt-текстів зображень) не лише погіршує сприйняття контенту, а й призводить до деградації лояльності аудиторії. За статистичними даними, затримка завантаження сторінки понад 3 секунди спричиняє відмову понад 50% потенційних користувачів, що робить моніторинг продуктивності через Performance API першочерговим завданням. По-друге, алгоритми пошукових систем (SEO) висувають дедалі суворіші вимоги до семантичної валідності та технічної гігієни сторінок. Помилки в метаданих (Title, Description, Canonical), дублювання H1 або некоректні robots-директиви стають причиною песимізації ресурсів у видачі, що прямо впливає на економічну ефективність онлайн-бізнесу. По-третє, існуючий інструментарій аудиту (наприклад, Google Lighthouse [1] або Screaming Frog [2]) часто є або занадто складним для оперативного використання non-QA фахівцями, або вимагає передачі даних на зовнішні сервери, що створює ризики для конфіденційності. Вбачається гостра потреба у створенні автономних, локальних засобів аналізу, які інтегруються безпосередньо у робоче середовище браузера.

Вибір архітектури на базі Chrome Extension (Manifest V3) із використанням React [3] та TypeScript [4] дозволяє реалізувати безпосередній доступ до DOM-структури сторінки в реальному часі. Це забезпечує можливість не просто фіксації помилок, а й глибокого аналізу контенту (щільність ключових слів, «водність»), детекції маркетингових інструментів (tracking-пікселів) та локальної генерації звітів без залучення сторонніх обчислювальних ресурсів.

Таким чином, розробка інтегрованого плагіна для комплексного аудиту веб-сторінок є актуальною науково-практичною задачею, розв'язання якої дозволить оптимізувати процеси контролю якості, підвищити SEO-показники та забезпечити стабільність функціонування сучасних веб-систем у висококонкурентному середовищі.

Таким чином, **метою випускної роботи** є розроблення інтегрованого програмного комплексу у форматі розширення для браузера Google Chrome (на базі Manifest V3) для автоматизованого багатофакторного аудиту SEO-показників, технічної продуктивності та якості контенту веб-сторінок у реальному часі.

Для досягнення мети розв'язано такі **завдання**:

1. Проведено системний аналіз існуючих засобів браузерного моніторингу та ідентифіковано критичні технічні помилки, що деградують користувацький досвід та SEO-рейтинг веб-ресурсів.

2. Спроектовано модульну архітектуру розширення із чітким розподілом логіки сканування (SEO, Content, Performance, Tracking) та інтерфейсної частини.

3. Реалізовано алгоритми глибокого аналізу DOM-структури для виявлення дублікатів тегів, mixed content та відсутності атрибутів доступності.

4. Розроблено математичну модель розрахунку інтегральної оцінки (scoring logic) на основі штрафних балів за технічні та семантичні недоліки сторінки.

5. Реалізовано функціонал локальної генерації аналітичних PDF-звітів та ергономічний UI з можливостями інтерактивної взаємодії з контентом сторінки.

Об’єкт дослідження — програмна система автоматизованого багатофакторного аудиту та моніторингу технічного стану веб-ресурсів у браузерному середовищі.

Предмет дослідження — архітектура хром-плагіна, алгоритми аналізу структури та продуктивності сторінок, а також методи візуалізації та звітності результатів аудиту.

Методи та технології реалізації.

Програмна реалізація системи базується на стандарті Manifest V3, що забезпечує оптимальний баланс між безпекою та швидкодією в середовищі Chromium. Архітектурний каркас проєкту побудований на використанні бібліотеки React у поєднанні зі строго типізованою мовою TypeScript, тоді як процес збірки та стилізації оптимізовано за допомогою Vite [5] та Tailwind CSS [6]. Функціональна взаємодія між Popur UI та Content Script здійснюється через асинхронний механізм обміну повідомленнями за посередництва Background Service Worker. Автономне скануюче ядро має модульну структуру, де логіка SEO-аналізу, діагностики продуктивності та семантичного аудиту контенту відокремлена від інтерфейсної частини. Математична модель розрахунку інтегральної оцінки уніфікована в спільному програмному модулі, що гарантує валідність даних у всіх представленнях. Завершальним етапом обробки результатів є локальна генерація PDF-звітів засобами Canvas та маніпуляцій з PDF-байтами без залучення зовнішніх серверних ресурсів.

Практична цінність результатів дослідження. Науково-практична значущість розробленого програмного комплексу обумовлена створенням автономного, високопродуктивного інструментарію для багатофакторної діагностики веб-ресурсів, що інтегрується безпосередньо в операційне середовище фахівця. Практична цінність роботи полягає у наступних аспектах:

1. Оптимізація процесів технічного аудиту. Впровадження плагіна дозволяє суттєво скоротити часові витрати на проведення первинного SEO- та технічного аналізу сторінок. Завдяки прямому доступу до DOM-структури, фахівці отримують можливість ідентифікувати детерміновані дефекти

(відсутність метаданих, дублювання заголовків H1, порушення ієрархії тегів) безпосередньо у процесі перегляду сайту, що критично важливо для динамічних веб-додатків.

2. Забезпечення конфіденційності та автономності. На відміну від більшості існуючих хмарних сервісів, розроблена система функціонує виключно на стороні клієнта. Це гарантує повну безпеку технічних даних та інтелектуальної власності, оскільки аналіз коду, обробка метрик продуктивності та генерація звітів відбуваються локально, без передачі інформації на зовнішні сервери.

3. Висока точність діагностики продуктивності. Використання нативного Performance API браузера дозволяє зчитувати метрики реального користувацького досвіду (TTFB, DOM complete, час DNS-лукапу) у реальному часі. Це надає розробникам точні дані для усунення «вузьких місць» у швидкодії сайту, що безпосередньо впливає на зниження показника відмов та підвищення конверсії.

4. Автоматизація аналітичної звітності. Реалізований модуль локальної генерації PDF-звітів дозволяє миттєво трансформувати результати сканування у структурований документ. Це забезпечує спеціалістів готовою доказовою базою для формування технічних завдань на виправлення помилок або для надання звітів клієнтам та керівництву.

5. Універсальність і дидактичний потенціал. Розроблений інтерфейс із підтримкою багатомовності та компактним режимом підсвічування ключових слів робить інструмент зручним як для професійних QA-інженерів і SEO-спеціалістів, так і для використання в освіті. Плагін може слугувати практичним засобом при вивченні тем з цифрової грамотності, дозволяючи демонструвати принципи ергономіки та технічної валідності веб-сторінок.

Таким чином, результати роботи мають пряме прикладне застосування у сфері інформаційних технологій, забезпечуючи підвищення якості веб-контенту та стабільність функціонування сучасних інтерактивних систем.

Розділ 1

АНАЛІТИЧНИЙ ОГЛЯД ТА ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

Сучасна парадигма розвитку мережевих технологій демонструє стрімку еволюцію веб-застосунків у бік складних інтерактивних систем, де якість технічної реалізації прямо корелює з економічною ефективністю бізнесу. Технічні недоліки, такі як некоректна структура DOM, відсутність метаданих (Title, Description) або порушення ієрархії заголовків, виступають критичними факторами песимізації сайту в пошукових алгоритмах

Аналіз існуючих рішень, зокрема Google Lighthouse та PageSpeed Insights [7], виявив їхню високу точність у вимірюванні продуктивності, проте вони часто вимагають значних обчислювальних ресурсів або спеціалізованих знань для інтерпретації результатів. Інструменти на кшталт Screaming Frog забезпечують глибокий аудит, але функціонують як десктопні додатки, що ускладнює оперативний моніторинг безпосередньо в процесі серфінгу.

Таким чином, виникає науково-практична потреба у створенні легкого, але багатофункціонального інструменту, інтегрованого в браузерне середовище, який дозволяє проводити миттєву діагностику без передачі даних на зовнішні сервери. У цьому розділ описано особливості складних веб-систем, класифіковано різні помилки та їх вплив на конверсію веб-ресурсів, порівняно існуючі інструменти SEO аналізу, описано місце розробленого плагіна з поміж існуючих.

1.1. Сучасний стан та архітектурні особливості складних веб-систем

Сучасна парадигма розвитку глобального інформаційного середовища характеризується трансформацією веб-ресурсів від статичних інформаційних сторінок до надскладних інтерактивних систем, що інтегрують у собі розгалужену бізнес-логіку та багатoshарові структури даних. Еволюція веб-додатків обумовлена тотальною диджиталізацією економічних процесів, де якість технічної реалізації ресурсу безпосередньо корелює з його ринковою конкурентоспроможністю та здатністю генерувати прибуток.

На сучасному етапі веб-системи визначаються високою динамічністю та складністю реалізації клієнтської частини. Архітектурно вони базуються на принципах адаптивного дизайну, включають численні інтерактивні елементи та складні скриптові сценарії, що функціонують безпосередньо у браузері користувача. Проте зростаюча складність таких додатків неминує призводить до виникнення детермінованих дефектів: неефективного коду, помилок у роботі серверів, погано оптимізованих запитів до баз даних або порушень у DOM-структурі. У контексті підвищення вимог до користувацького досвіду UX, виявлення та усунення таких помилок на ранніх етапах розробки стає важливим завданням.

Основними технічними характеристиками, що визначають якість сучасної веб-системи, є семантична правильність структури, відповідність стандартам W3C, доступність контенту для людей з особливими потребами, оптимізація швидкодії та стійкість до високих навантажень. За даними аналітичних досліджень [8], затримка завантаження сторінки понад 3 секунди призводить до відмови більше ніж 50% користувачів, що робить продуктивність (метрики LCP, FCP, TTFB) визначальним фактором виживання продукту в мережі.

З огляду на архітектурну специфіку сучасних веб-додатків, найбільш ефективним інструментом для їхнього аналізу стають браузерні розширення.

Вибір архітектури на базі Google Chrome (Manifest V3) забезпечує прямий доступ до DOM-структури сторінки, що дозволяє виконувати перевірки безпосередньо в контексті браузера, де і відбувається взаємодія кінцевого користувача з продуктом. Модульна архітектура таких аналітичних засобів, що включає Content Scripts для роботи з елементами сторінки та Background Service Workers для координації асинхронних процесів, дозволяє реалізувати автономне скануюче ядро для оперативного SEO та технічного аудиту.

Таким чином, сучасний стан веб-індустрії вимагає впровадження інтегрованих систем автоматизованого моніторингу, здатних ідентифікувати технічні, семантичні та експлуатаційні недоліки в режимі реального часу, забезпечуючи стабільність функціонування складних веб-систем у висококонкурентному цифровому середовищі.

1.2. Класифікація детермінованих дефектів та їх вплив на конверсію веб-ресурсів

У контексті системного аналізу якості веб-інтерфейсів детерміновані дефекти розглядаються як ідентифіковані технічні та структурні недоліки, що виникають внаслідок помилок проєктування або розробки та мають прямий прогнозований негативний вплив на показники ефективності ресурсу. Кожна помилка, навіть мінімальна, має властивість кумулятивного ефекту, створюючи перешкоди для користувача та призводячи до деградації конверсійного шляху.

Для побудови цілісної моделі аудиту доцільно класифікувати дефекти за категоріями їхнього впливу на архітектуру та функціональність веб-системи [9-12]:

1. Технічні та інфраструктурні дефекти. До цієї групи належать помилки, що порушують базову працездатність ресурсу:

а) наявність «битих» посилань та некоректна обробка сторінок помилки 404, що дезорієнтує відвідувача та знижує довіру до ресурсу;

б) ланцюги та цикли редиректів, які сповільнюють швидкість завантаження та негативно впливають на показник PageRank;

в) помилки HTTP-протоколу (типу 4xx та 5xx), а також відсутність SSL-сертифіката, що сприймається пошуковими системами та браузерами як фактор ризику безпеки;

г) низька швидкість завантаження сторінок, зумовлена неоптимізованими скриптами або надмірною вагою зображень.

2. Структурні та SEO-дефекти. Ці недоліки перешкоджають коректній індексації ресурсу пошуковими алгоритмами:

а) нерелевантні, відсутні або дубльовані мета-теги (Title, Description) та заголовки H1;

б) порушення ієрархії підзаголовків H2–H3, що ускладнює семантичний аналіз контенту ботами та сприйняття інформації користувачами;

в) неправильна конфігурація файлу `robots.txt` або канонічних посилань (`rel=canonical`), що призводить до штучного блокування сторінок для сканування.

3. Дефекти юзабіліті (UX) та інтерфейсу (UI). Дана категорія дефектів створює «інформаційний шум» та бар'єри для взаємодії:

а) відсутність адаптивної верстки або мобільної версії, що загрожує втратою до 60% цільового трафіку;

б) перевантаженість інформацією та відсутність «вільного простору» (white space), що розсіює увагу та заважає фокусуванню на ключових закликах до дії;

в) інтуїтивно незрозуміла навігація та відсутність функцій внутрішнього пошуку, особливо критична для інтернет-магазинів;

г) некоректна робота веб-форм, надмірна кількість обов'язкових полів або складна валідація даних, що змушує користувача припинити процес реєстрації чи замовлення на «півдороги».

Вплив зазначених дефектів на конверсію веб-ресурсів може виразитися у прямій математичній залежності. Зокрема, встановлено, що затримка завантаження понад 3 секунди призводить до відмови більше ніж 50% відвідувачів. Дефекти, пов'язані з відсутністю актуальних цін («ціни за запитом») або складністю процедури оформлення замовлення, апріорі підвищують відсоток відмов, оскільки сучасний користувач схильний переходити на ресурси конкурентів із прозорішою політикою [8, 13].

Науково обґрунтований підхід до усунення детермінованих дефектів передбачає не лише виправлення технічного коду, а й глибоку оптимізацію користувацького шляху (Customer Journey), що дозволяє трансформувати відвідувачів у реальних покупців, підвищуючи рентабельність інвестицій (ROI) цифрового продукту [13]. Таким чином, систематичний аудит та класифікація дефектів є першочерговою стадією впровадження ефективної стратегії розвитку будь-якої веб-системи.

1.3. Компаративний аналіз існуючих інструментів технічного та SEO-аудиту

Для забезпечення високої технічної якості та конкурентоспроможності веб-ресурсів у сучасному цифровому середовищі використовується розгалужена екосистема аналітичних засобів. Кожен з існуючих інструментів має власну специфікацію, орієнтовану на вирішення конкретних завдань: від моніторингу швидкодії до глибокого семантичного аудиту та аналізу посилальної маси.

Чільне місце серед інтегрованих браузерних рішень посідає Google Lighthouse — автоматизований інструмент із відкритим вихідним кодом, що дозволяє проводити комплексний аудит продуктивності, доступності (Accessibility), дотримання технічних стандартів та SEO безпосередньо в середовищі Chrome DevTools. Паралельно з ним функціонує PageSpeed Insights, який фокусується на метриках завантаження сторінок (таких як FCP, LCP) та надає детальні рекомендації щодо стиснення зображень і мінімізації коду. Проте ці інструменти, попри їхню еталонність, часто вимагають спеціалізованих знань для інтерпретації результатів і здебільшого аналізують сайт лише на стороні клієнта.

Для глибокого дослідження зовнішніх чинників ранжування та семантичного ядра використовуються професійні платформи, такі як Ahrefs та Screaming Frog SEO Spider. Ahrefs [14] забезпечує моніторинг позицій у пошукових системах, аналіз беклінків та стратегій конкурентів, що є критичним для стратегічного маркетингового планування. У свою чергу, Screaming Frog функціонує як потужний кроулер, що ідентифікує технічні дефекти, такі як ланцюги редиректів, «биті» посилання та некоректні мета-теги Title і Description. Основним недоліком цих систем є їхня висока вартість та десктопна орієнтованість, що ускладнює оперативний аудит «в один клік».

Окремим вектором розвитку є інструменти аналізу користувачького досвіду та поведінкових патернів, зокрема Plerdy [15] та Crazy Egg [16]. Plerdy

пропонує інтегровані рішення, що включають теплові карти кліків, записи сеансів та SEO-checker у форматі розширення для Chrome, що дозволяє виявляти проблемні зони в UI/UX дизайні на основі реальної взаємодії користувача з інтерфейсом. Додаткову діагностику швидкодії з візуалізацією вузьких місць у завантаженні скриптів забезпечує сервіс GTmetrix [17].

Результати компаративного аналізу свідчать, що незважаючи на потужність існуючого інструментарію, більшість рішень є або занадто складними для широкого кола фахівців, або вимагають передачі даних на сторонні сервери. Виникає науково-практична потреба у створенні автономного, легкого хром-плагіна, який поєднує в собі функції миттєвої перевірки SEO-метрик, технічної продуктивності та якості контенту з можливістю локальної генерації звітів без порушення конфіденційності даних.

1.4. Обґрунтування доцільності розробки автономного браузерного розширення на базі

Синтез результатів компаративного аналізу існуючих рішень (див. підрозділ 1.3) дозволяє констатувати наявність суттєвого розриву між потужними аналітичними платформами та потребами фахівців у оперативній, конфіденційній і структурно точній діагностиці веб-сторінок. Більшість хмарних сервісів та десктопних додатків мають обмеження, пов'язані з необхідністю передачі даних на зовнішні сервери, що створює ризики для приватності та конфіденційності інтелектуальної власності. У цьому контексті проєктування інструмента у форматі розширення для браузера Google Chrome є найбільш обґрунтованим архітектурним рішенням.

Вибір браузерного розширення як платформи для реалізації системи аудиту обумовлений наступними чинниками:

1. Прямий доступ до DOM-структури. На відміну від зовнішніх кроулерів, плагін функціонує безпосередньо в середовищі браузера, що забезпечує безперешкодний доступ до об'єктної моделі документа (DOM) у реальному часі. Це дозволяє ідентифікувати помилки в ієрархії тегів, виявляти дублікати метаданих та аналізувати динамічний контент, який генерується клієнтськими скриптами і може бути недоступним для статичних аналізаторів.

2. Повна автономність та конфіденційність. Проєкт передбачає створення локального скануючого ядра, що виключає залежність від зовнішньої серверної інфраструктури. Весь процес обробки даних — від зчитування тегів до генерації PDF-звітів — відбувається на стороні клієнта, що гарантує збереження приватності користувача та безпеку веб-даних.

3. Використання стандарту Manifest V3. Реалізація плагіна на базі Manifest V3 є важливою для забезпечення відповідності сучасним вимогам Google щодо безпеки, продуктивності та енергоефективності. Використання сервіс-воркерів (background service workers) замість постійно активних фонових сторінок

дозволяє оптимізувати використання ресурсів пам'яті та забезпечити стабільну роботу розширення навіть при аналізі високонавантажених веб-додатків.

4. Інтеграція з Performance API. Браузерне середовище надає унікальну можливість безпосередньої взаємодії з вбудованими інструментами діагностики продуктивності. Це дозволяє збирати точні метрики завантаження (TTFB, LCP, DOM complete) безпосередньо в момент взаємодії користувача зі сторінкою, що є недосяжним для багатьох зовнішніх сервісів.

Отже, розробка автономного хром-плагіна на базі Manifest V3 є науково та практично доцільною, оскільки такий підхід поєднує в собі високу швидкість обробки даних, точність діагностики на рівні вихідного коду та повну незалежність від сторонніх обчислювальних ресурсів. Це дозволяє створити ергономічний та безпечний інструмент для SEO-спеціалістів, розробників та маркетологів, який органічно інтегрується в їхній повсякденний робочий процес.

Розділ 2

ПРОЄКТУВАННЯ ТА ТЕОРЕТИЧНЕ ОБҐРУНТУВАННЯ СИСТЕМИ

У контексті формування вимог до програмного комплексу визначено пріоритетність модульної архітектури, що забезпечує незалежне функціонування блоків SEO-аналізу, діагностики продуктивності та семантичного аудиту. Технологічний стек проекту базується на використанні бібліотеки React та строго типізованої мови TypeScript, що дозволяє досягти високої надійності коду та модульності компонентів. Вибір стандарту Manifest V3 обумовлений жорсткими вимогами до безпеки та енергоефективності сучасних браузерних розширень.

Математична модель оцінювання якості веб-сторінки базується на розрахунку інтегрального показника якості Q [13, 18], який обчислюється за формулою дедуктивної деградації балів. Початкове значення показника становить 100 одиниць, від якого віднімаються вагові штрафні коефіцієнти за виявлені технічні помилки E , наявність змішаного контенту M , повільні ресурси P та некоректні robots-директиви R . Семантична складова оцінюється через Content Score S , що враховує щільність ключових слів, індекс читабельності FRES та показник «водності» тексту. Такий підхід забезпечує об'єктивність оцінювання, оскільки враховує як технічну валідність коду, так і якість контенту для кінцевого користувача. Очікуваним результатом впровадження системи є скорочення часу на проведення первинного аудиту та підвищення точності ідентифікації прихованих помилок у DOM-структурі.

2.1. Формування функціональних та технічних вимог до інтегрованого засобу аудиту

Процес проектування інтегрованого засобу аудиту вимагає чіткої детермінації функціональних та технічних специфікацій, що забезпечують валідність аналізу в умовах високої динамічності сучасних веб-додатків. Враховуючи вплив технічних дефектів на конверсію та користувацький досвід UX, формування вимог базується на необхідності охоплення широкого спектра метрик — від семантичної коректності коду до показників продуктивності та маркетингової прозорості.

Функціональні вимоги до системи охоплюють чотири вектори аналізу.

Перший вектор — це технічний SEO-моніторинг, який повинен забезпечувати автоматизовану перевірку метаданих (title, meta description), канонічних URL-адрес, атрибутів Open Graph та індексаційних директив robots meta. Важливою функцією є ідентифікація критичних структурних помилок, таких як дублювання заголовків H1, відсутність атрибутів alt у зображень та наявність порожніх або «битих» посилань, що безпосередньо впливає на ранжування сторінки. Система має реалізовувати алгоритм групування однотипних проблем для оптимізації сприйняття звіту користувачем.

Другий функціональний блок стосується діагностики продуктивності та технічної гігієни. Інструмент повинен взаємодіяти з нативним Performance API браузера для фіксації метрик завантаження (TTFB, DNS lookup, TCP connection, DOM complete) та ідентифікації mixed content (HTTP-ресурси на HTTPS-сторінках), що є критичним фактором безпеки та швидкодії

Третій вектор включає семантичний аналіз контенту: розрахунок кількості слів, визначення «водності» тексту та щільності ключових фраз. Специфічною вимогою є забезпечення інтерактивності, зокрема можливості підсвічування обраних ключових слів безпосередньо в структурі сторінки.

Четвертий блок передбачає детекцію технологічного стека (CMS, CDN) та tracking-пікселів провідних рекламних систем (Google Ads, Facebook, TikTok), що дозволяє оцінити маркетингову оснащеність ресурсу.

Загалом можна подати Use Case діаграму і діаграму послідовностей.

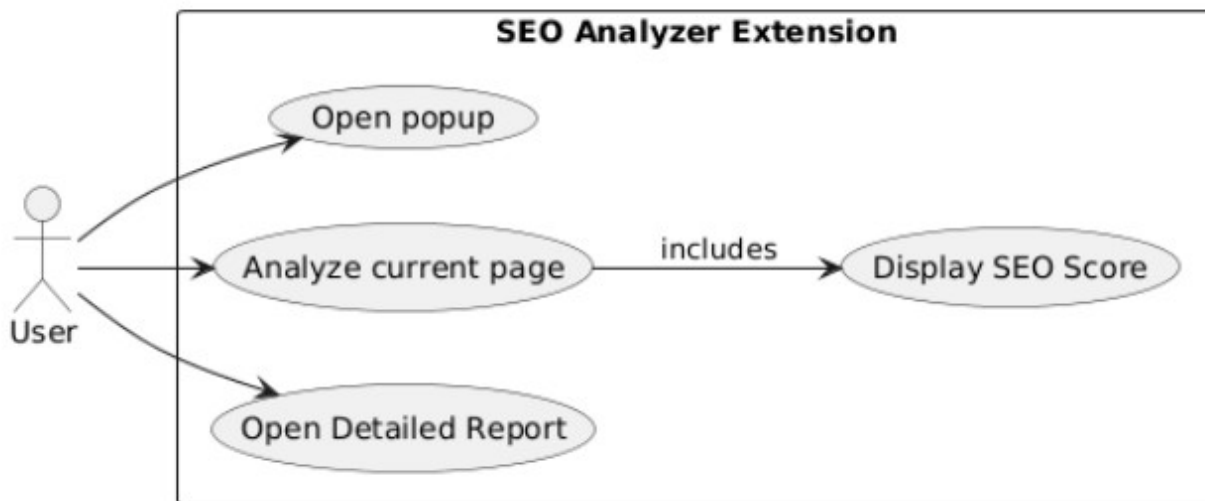


Рисунок 2.1. Діаграма випадків для плагіна.

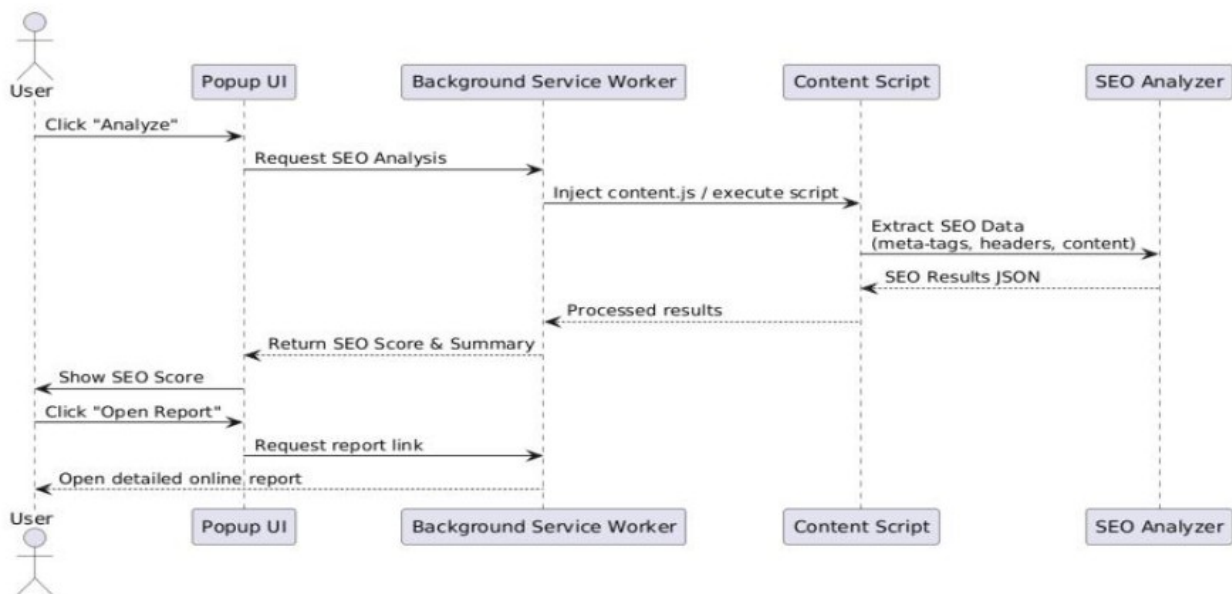


Рисунок 2.2. Sequence Diagram.

Технічні вимоги до програмного комплексу зумовлені архітектурними обмеженнями середовища Chromium та необхідністю забезпечення

конфіденційності даних. Фундаментальною вимогою є реалізація плагіна на базі стандарту Manifest V3, що гарантує підвищений рівень безпеки та ефективне використання ресурсів пам'яті через Background Service Workers. Архітектурний каркас має бути побудований з використанням бібліотеки React та мови TypeScript для забезпечення строгої типізації даних та модульності компонентів, що полегшує подальшу масштабованість системи.

Окремою технічною вимогою є автономність обробки даних: усі обчислення, включаючи розрахунок інтегральної оцінки Scoring Logic та генерацію PDF-звітів, повинні відбуватися локально на стороні клієнта за допомогою Canvas та PDF-маніпуляцій, без передачі інформації на зовнішні сервери. Система повинна підтримувати асинхронну взаємодію між Popup UI та Content Scripts через механізм Message Passing, забезпечуючи коректне зчитування DOM-структури активної вкладки без деградації продуктивності браузера. Інтерфейс плагіна має підтримувати двомовну локалізацію та адаптивний режим відображення для зручної роботи фахівця в контексті перегляду сторінки.

2.2. Обґрунтування вибору технологічного стека

Вибір технологічного базису для реалізації інтегрованого аналітичного засобу обумовлений необхідністю забезпечення високої продуктивності, надійності та відповідності сучасним стандартам безпеки браузерних екосистем. Архітектурний каркас проєкту базується на поєднанні бібліотеки React та строго типізованої мови TypeScript, що дозволяє реалізувати модульну структуру з чітким розподілом функціональних обов'язків між компонентами. Використання React забезпечує компонентний підхід до побудови користувацького інтерфейсу (Popur UI). Це важливо для підтримки динамічного відображення результатів аналізу та реалізації компактного режиму візуалізації. TypeScript, у свою чергу, надає інструментарій для суворої типізації даних, зокрема структурованих об'єктів результатів сканування, що мінімізує ймовірність виникнення помилок під час обміну повідомленнями між різними шарами розширення.

Фундаментальним архітектурним рішенням є перехід на стандарт Manifest V3, який наразі є галузевим еталоном конфігурації розширень для браузера Google Chrome. Використання Manifest V3 дозволяє значно підвищити безпеку та енергоефективність системи за рахунок впровадження фонових сервіс-воркерів (Background Service Workers) замість постійно активних фонових сторінок. Service Worker виступає центральним координатором, який асинхронно обробляє запити від інтерфейсу та контентних скриптів, підтримуючи стабільність роботи плагіна навіть при значних навантаженнях на DOM-структуру активної вкладки.

Для забезпечення безпосередньої взаємодії з веб-сторінкою в проєкті задіяно механізм Content Scripts, які функціонують безпосередньо в контексті браузера користувача. Це надає розширенню повний доступ до об'єктної моделі документа (DOM) та дозволяє виконувати глибокий технічний аналіз, включаючи перевірку ієрархії заголовків, атрибутів доступності та метаданих, без втручання в серверну частину досліджуваного ресурсу. Комунікація між

Popup UI та Content Script реалізована через Message Passing API, що забезпечує швидку та безпечну передачу зібраних даних для подальшої обробки скануючим ядром.

Процес збірки та стилізації проекту оптимізовано за допомогою сучасних інструментів розробки — Vite та Tailwind CSS. Використання Vite дозволяє досягти високої швидкості гарячого оновлення модулів (HMR) та мінімізувати обсяг фінального коду розширення, що позитивно впливає на швидкість його ініціалізації у браузері. Утилітарний CSS-фреймворк Tailwind CSS надає можливість реалізувати ергономічний та адаптивний дизайн інтерфейсу з дотриманням принципів юзабіліті, забезпечуючи коректне відображення аналітичних даних на пристроях з різною роздільною здатністю. Таким чином, обраний технологічний стек формує цілісну, високопродуктивну екосистему, здатну виконувати багатофакторний аудит веб-сторінок у реальному часі з дотриманням найвищих вимог до безпеки даних.

2.3. Математична модель інтегрального оцінювання якості веб-сторінки

У межах науково-технічного проектування інтегрованих засобів аудиту фундаментальне значення має математична формалізація процесу оцінювання, яка дозволяє трансформувати дискретні технічні параметри у континуальний показник якості. Для об'єктивної детермінації стану веб-ресурсу в розробленому плагіні було впроваджено концепцію інтегрального показника якості Q , що базується на принципі дедуктивної деградації балів [8, 13, 18]. Замість адитивного накопичення позитивних ознак, алгоритм використовує еталонне значення (100 балів), від якого віднімаються вагові “штрафні” коефіцієнти за кожен виявлену групу детермінованих дефектів. Такий підхід корелює з галузевими стандартами, зокрема з методологією Google Lighthouse, де оцінка формується на основі виявлених невідповідностей технічним вимогам. Математично модель можна представити у вигляді функціональної залежності:

$$Q = 100 - (\sum P_{issues} + P_{mixed\ content} + P_{performance} + P_{robots}) \cdot C_{content}, \quad (2.1)$$

де кожна змінна відображає окремий пласт технічного та семантичного аудиту.

Параметр штрафів за технічні проблеми $\sum P_{issues}$ включає сукупну вагу помилок, ідентифікованих сканером в DOM-структурі. До критичних факторів, що мають найбільшу питому вагу у деградації оцінки, належать відсутність метаданих (Title та Description), некоректне налаштування канонічних URL-адрес та дублювання заголовків першого рівня H1, що створює семантичний шум для пошукових алгоритмів. Додатково враховується відсутність атрибутів доступності, зокрема `alt` для зображень, що прямо впливає на юзабіліті та інклюзивність ресурсу. Штраф за змішаний контент $P_{mixed\ content}$ відображає стан безпеки та технічної гігієни: наявність HTTP-ресурсів на HTTPS-сторінках розцінюється як критична вразливість, що призводить до блокування елементів браузером та зниження довіри користувачів.

Діагностика продуктивності $P_{performance}$ інтегрує дані, отримані через нативний Performance API. Штрафні бали нараховуються за перевищення нормативних значень швидкодії, таких як Time to First Byte (TTFB), DOM interactive та загальний час завантаження. Оскільки затримка завантаження понад 2–3 секунди є критичною для утримання аудиторії, вага цього параметра в математичній моделі є значною. Особливістю алгоритму є впровадження корекційного коефіцієнта якості контенту $C_{content}$, який базується на семантичному аналізі. Він враховує «водність» тексту, щільність ключових слів та індекс читабельності FRES, що дозволяє системі не просто фіксувати наявність тегів, а оцінювати корисність інформаційного наповнення для кінцевого споживача.

Для забезпечення цілісності та верифікованості даних логіка розрахунку інтегральної оцінки винесена в окремий уніфікований програмний модуль ``src/shared/scoring.ts``. Це гарантує ідентичність обчислень як у динамічному Рорир-інтерфейсі плагіна, так і в генерованих PDF-звітах, усуваючи ризик розбіжностей при інтерпретації результатів. Таким чином, розроблена математична модель дозволяє отримати комплексну багатофакторну оцінку веб-сторінки, що враховує технічну валідність, швидкість функціонування та семантичну цінність контенту в єдиному цифровому показнику.

2.4. Алгоритмічне забезпечення семантичного аналізу та технічної діагностики

Фундаментальною основою функціонування розробленого плагіна є декомпозиція процесу аудиту на автономні алгоритмічні модулі, інтегровані у структуру content script. Це дозволяє реалізувати паралельну обробку різних пластів даних веб-сторінки, забезпечуючи високу детермінованість результатів при мінімальних витратах обчислювальних ресурсів клієнта. Алгоритмічне забезпечення системи розподілено за функціональними напрямками, що корелюють із ключовими модулями сканування.

Семантичний аналіз текстового наповнення сторінки реалізовано в межах модуля *contentScanner.ts*. Алгоритм аналізу базується на лінгвістичній обробці текстових вузлів DOM-дерева. Ключовими етапами цього процесу є очищення контенту від HTML-тегів, токенизація тексту та фільтрація «стоп-слів» для виявлення семантичного ядра. Система автоматично розраховує кількісні метрики: загальну кількість слів, щільність та релевантність ключових фраз, а також інтегральний показник «водності» контенту. Особливістю алгоритму є ідентифікація топ-20 ключових слів з можливістю їхньої подальшої інтерактивної візуалізації. Алгоритм підсвічування реалізовано через динамічну ін'єкцію стилів у DOM-структуру, що дозволяє користувачеві в компактному режимі роруп-інтерфейсу наочно оцінити розподіл ключових понять на сторінці.

Технічна діагностика продуктивності базується на алгоритмах модуля *performanceScanner.ts*, який здійснює пряму взаємодію з нативним Browser Performance API. На відміну від зовнішніх інструментів аналізу, даний підхід дозволяє зчитувати метрики безпосередньо з контексту завантаження активної вкладки. Алгоритм фіксує критичні часові мітки: час DNS-лукапу, TCP-з'єднання, Time to First Byte (TTFB), а також моменти досягнення станів DOM interactive та DOM complete. Додатково імплементовано алгоритм пошуку

повільних ресурсів, що дозволяє ідентифікувати конкретні об'єкти, які деградують загальну швидкість рендерингу.

Модуль *linkScanner.ts* забезпечує реалізацію алгоритмів верифікації цілісності гіперпосилань та безпеки контенту. Алгоритм технічного аудиту виконує ітеративний обхід атрибутів href та src для виявлення порожніх посилань та зображень без атрибута alt. Окремий критичний блок алгоритму присвячено детекції змішаного контенту (mixed content): система автоматично маркує HTTP-ресурси, що завантажуються в межах захищеного HTTPS-з'єднання, що є ключовим параметром безпеки в сучасних веб-стандартах.

Інтеграція результатів здійснюється через модуль *seoScanner.ts*, який консолідує дані всіх підсистем у типізований інтерфейс AnalyzerResult. Алгоритмічна відокремленість логіки сканування від користувацького інтерфейсу UI гарантує стабільність роботи системи: дані обробляються у фоновому режимі та передаються в роруп через асинхронні повідомлення, що виключає «заморожування» основної сторінки під час проведення аудиту. Таким чином, розроблене алгоритмічне забезпечення формує замкнутий цикл діагностики, що охоплює як технічну валідність сторінки, так і якість її семантичного наповнення.

Розділ 3

ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ЕКСПЛУАТАЦІЙНІ ХАРАКТЕРИСТИКИ РОЗРОБЛЕНОЇ СИСТЕМИ

3.1. Особливості програмної реалізації модульної архітектури на базі стандарту Manifest V3

Програмна реалізація розробленого плагіна базується на суворій модульній декомпозиції, що відповідає сучасним стандартам розробки розширень для браузерів сімейства Chromium. Основу проєкту складає директорія `src/`, структурована за функціональним призначенням компонентів, що дозволяє мінімізувати зв'язність коду та полегшує його сервісну підтримку.

Ключовим елементом архітектури є перехід на стандарт Manifest V3, що вимагає використання ***background/serviceWorker.ts*** замість застарілих фонових сторінок. Даний модуль виступає в ролі «подієвого шини» (event bus), яка асинхронно обробляє запити від інтерфейсу (***src/popup/***) до контентних скриптів (***src/content/***) через механізм `chrome.tabs.sendMessage`.

Процес збірки (build process) оптимізовано за допомогою інструментарію Vite, який здійснює транспіляцію TypeScript-коду та мініфікацію статичних ресурсів. Окрему увагу приділено модулю `src/shared/`, де винесено спільні інтерфейси (типи `AnalyzerResult`), логіку групування `issues` та єдину формулу розрахунку оцінки ***scoring.ts***, що гарантує цілісність даних при їх візуалізації як у рорир-вікні, так і в PDF-звіті. Така архітектурна побудова забезпечує високу швидкість ініціалізації плагіна та стабільність його роботи навіть на сторінках із надмірною кількістю DOM-елементів.

3.2. Розроблення адаптивного користувацького інтерфейсу та засобів інтерактивної візуалізації даних

Проектування інтерфейсу розробленого програмного комплексу базувалося на фундаментальних принципах юзабіліті та ергономіки, що є особливо важливим для забезпечення ефективної взаємодії користувача з аналітичними даними в реальному часі. Використання бібліотеки React у поєднанні з утилітарним CSS-фреймворком Tailwind CSS дозволило реалізувати компонентно-орієнтований підхід, забезпечуючи високу швидкість рендерингу та адаптивність візуальних елементів до різних конфігурацій браузерного вікна.

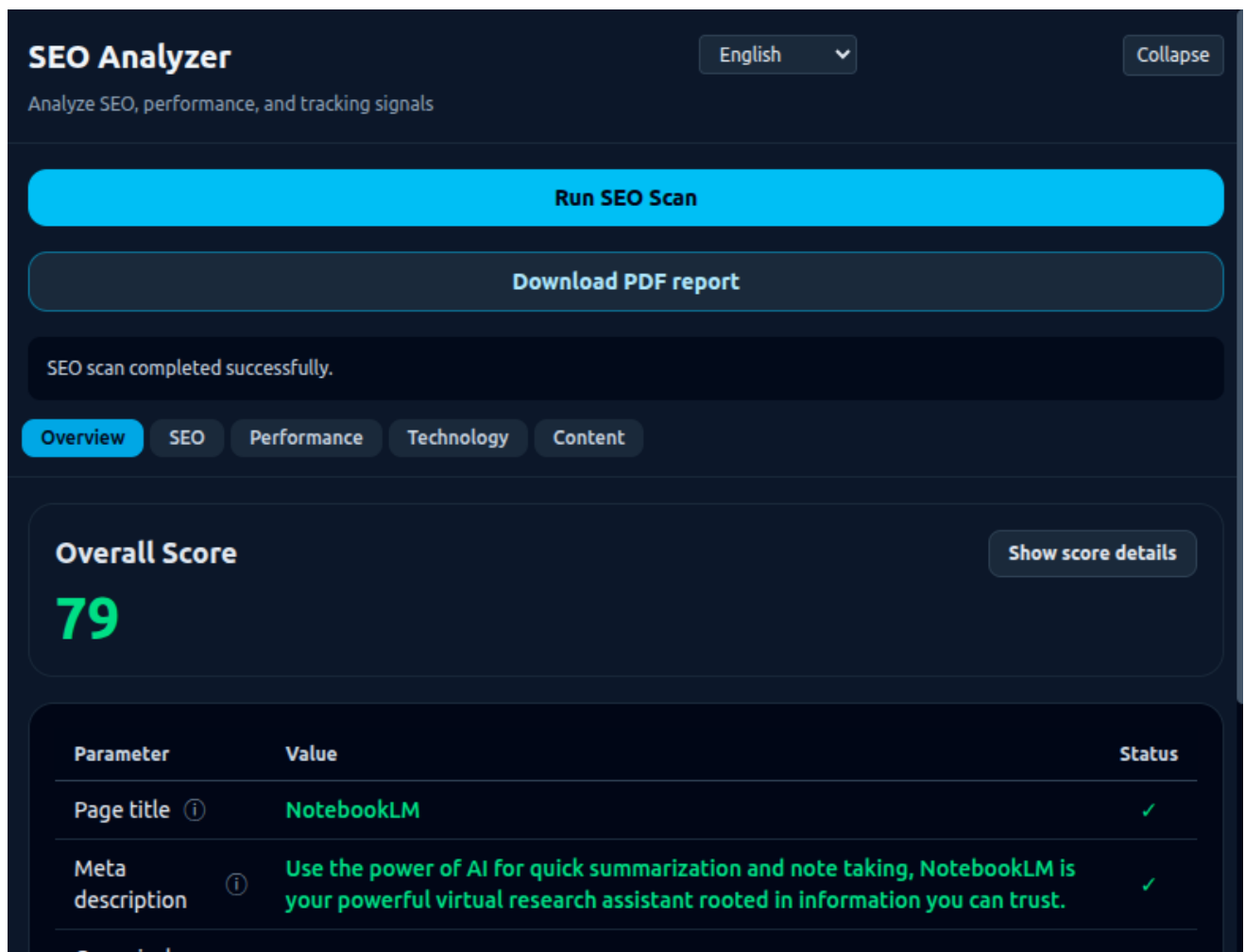


Рисунок 3.1. Вікно плагіна.

Центральним вузлом взаємодії є головне вікно Роруп-інтерфейсу (Рисунок 3.1), архітектура якого спроектована таким чином, щоб мінімізувати когнітивне навантаження на фахівця. На відміну від існуючих інструментів, що часто перевантажують користувача надмірним обсягом неструктурованої інформації, наш інтерфейс реалізує чітку логічну ієрархію відомостей.

Верхня частина Роруп-панелі відведена під візуалізацію інтегральної оцінки (SEO Score), що дозволяє миттєво верифікувати загальний технічний стан ресурсу. Нижче розташовані блоки детермінованих показників: метадані (Title, Description), канонічні URL та індексаційні директиви. Науково-практичною особливістю реалізації є інтелектуальний модуль групування помилок (issues grouping). Замість лінійного переліку всіх виявлених дефектів, система агрегує однотипні порушення, наприклад, відсутність атрибутів `alt` для групи зображень, відображаючи їх як єдиний запис із зазначенням кількісного показника, що значно спрощує процес прийняття рішень щодо оптимізації.

Окремим аспектом ергономічної організації інтерфейсу є впровадження системи тематичних вкладок (tabs), що забезпечує чітку логічну декомпозицію функціональних можливостей плагіна. Оскільки система проводить багатофакторний аналіз, що охоплює різні пласти даних — від технічного коду до семантичних метрик — використання табованої навігації дозволяє згрупувати результати за відповідними категоріями: «SEO-аудит», «Аналіз контенту», «Продуктивність» та «Маркетинговий трекінг».

Така інтерфейсна архітектура мінімізує когнітивне навантаження на користувача, надаючи можливість фокусуватися на конкретному векторі діагностики без візуального перевантаження основними масивами даних. Кожна вкладка акумулює вузькоспеціалізовані інструменти та графіки, що дозволяє фахівцеві послідовно верифікувати стан веб-ресурсу, зберігаючи при цьому цілісність контексту поточної сесії сканування. Це підкреслює високу адаптивність UI-рішення, розробленого засобами React, до складних аналітичних сценаріїв професійного аудиту.

Для забезпечення глибокого семантичного аналізу в інтерфейс інтегровано засіб інтерактивної візуалізації ключових лексем. На основі розрахунків модуля `contentScanner.ts` користувачеві надається перелік топ-20 ключових слів із показниками їхньої щільності та релевантності. Унікальною функціональною можливістю розробленого плагіна є механізм «зворотного зв'язку з контентом»: при натисканні на конкретне слово в списку воно автоматично підсвічується безпосередньо в DOM-структурі активної веб-сторінки.

З метою підвищення зручності такого візуального аудиту було розроблено компактний режим Роруп-інтерфейсу (Рисунок 3.2). Він дозволяє згорнути основну панель аналізу, залишаючи доступними лише елементи керування підсвічуванням та ключові метрики, що звільняє візуальний простір для дослідження самої сторінки.

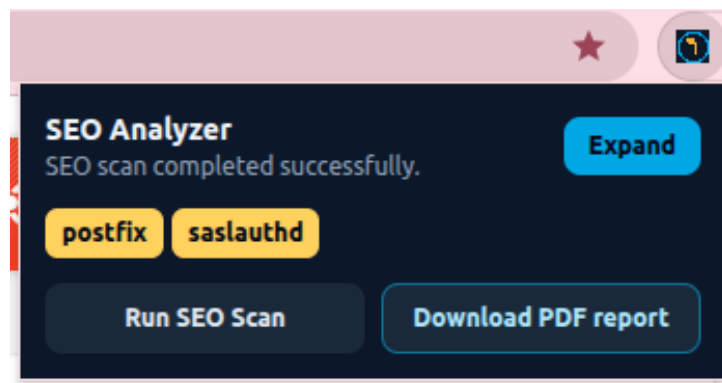


Рисунок 3.2. Компактний режим плагіна та інтерактивне підсвічування ключових слів

Технічна стабільність цих рішень забезпечується за рахунок суворої типізації інтерфейсів на TypeScript та асинхронного обміну повідомленнями з контентними скриптами, що унеможливорює виникнення візуального шуму або «заморожування» сторінки під час інтенсивної взаємодії з елементами UI. Такий підхід формує досконалий користувацький досвід UX, поєднуючи експертну глибину аналізу з ергономічною легкістю керування даними.

Окремим вектором функціональної досконалості системи є реалізація механізму багатомовної підтримки localization, що інтегрований безпосередньо

в інтерфейсну логіку. На поточному етапі розробки плагін забезпечує повну адаптацію контенту для українського та англійського сегментів, що дозволяє фахівцям оперувати валідним термінологічним апаратом незалежно від мовного контексту досліджуваного ресурсу. Архітектурне рішення, засноване на децентралізації лінгвістичних ресурсів у спеціалізованій директорії *src/locales/*, створює умови для практично необмеженого масштабування мовного стека.

Завдяки використанню суворої типізації TypeScript та уніфікованих JSON-структур для збереження перекладів, імплементація додаткових локалізацій не потребує реструктуризації основного коду. Процес розширення мовного охоплення зводиться до простого додавання нових словникових масивів, що забезпечує високу гнучкість системи та її готовність до міжнародної дистрибуції. Це відкриває значні перспективи для адаптації аналітичного інструментарію до специфічних потреб різних лінгвістичних груп у глобальному середовищі веб-розробки та SEO-аудиту

3.3. Функціональна реалізація діагностичних модулів та механізмів обміну повідомленнями

Програмна архітектура діагностичного ядра розширення базується на принципі децентралізації обчислювальної логіки, де основним середовищем виконання виступають контентні скрипти (*content scripts*), що інтегруються безпосередньо в об'єктну модель документа активної веб-сторінки. Вибір такої моделі обумовлений необхідністю отримання прямого доступу до DOM-структури для виконання глибокого технічного аудиту, який неможливо реалізувати через зовнішні інструменти аналізу. Взаємодія між інтерфейсною частиною та аналітичним ядром реалізована через асинхронний протокол обміну повідомленнями *chrome.tabs.sendMessage*, що дозволяє Роруп-вікну ініціювати сеанс сканування та отримувати структуровані дані у форматі *AnalyzerResult* без блокування основного потоку виконання браузера.

Діагностичний цикл розпочинається з роботи модуля *seoScanner.ts*, який здійснює верифікацію критичних метаданих, таких як заголовки сторінки, мета-опис та канонічні URL-адреси, а також аналізує індексаційні директиви *robots meta*. Окремим етапом функціонування даного модуля є звернення до *seoRules.ts*, де інкапсульовано бізнес-логіку виявлення структурних аномалій: наявності дублікатів заголовків першого рівня (H1), відсутності атрибутів *alt* у графічних об'єктів та ідентифікації порожніх посилань. Завдяки впровадженню алгоритмів агрегації, система групує ідентифіковані проблеми за типами, що дозволяє уникнути надмірності інформації при великій кількості однотипних дефектів на складних веб-сторінках.

Технічний аудит швидкодії забезпечується модулем *performanceScanner.ts*, який використовує нативний Performance API браузера для збору метрик реального користувацького досвіду, включаючи час встановлення TCP-з'єднання, Time to First Byte (TTFB) та моменти досягнення станів *DOM interactive* і *DOM complete*. Паралельно з цим модуль *linkScanner.ts* виконує ревізію внутрішніх та зовнішніх гіперпосилань, а також

проводить діагностику змішаного контенту (mixed content), виявляючи ресурси, що завантажуються через незахищений протокол HTTP, що є важливим фактором кібербезпеки сучасних систем.

Завершальна стадія аналізу реалізується модулями `contentScanner.ts` та `trackingScanner.ts`. Перший здійснює лінгвістичну обробку текстового масиву, розраховуючи показники щільності ключових слів, індекс читабельності FRES та коефіцієнт «водності» тексту, що дозволяє сформуванню об'єктивний Content Score сторінки. Модуль `trackingScanner.ts` у свою чергу виконує детекцію сторонніх сигнатур маркетингових інструментів, ідентифікуючи tracking-пікселі рекламних мереж Google, Meta, TikTok та LinkedIn, а також визначаючи технологічний стек сайту (CMS та CDN-провайдерів). Консолідований результат усіх сканерів передається назад до фонового сервіс-воркера, де відбувається фінальна обробка та підготовка даних для візуалізації в Popur-інтерфейсі розширення.

3.4. Технологія локальної генерації аналітичної звітності та верифікація математичної моделі оцінювання

Завершальним етапом функціонування розробленого програмного комплексу є трансформація результатів динамічного сканування у верифікований аналітичний документ. Науково-технічна новизна реалізації цього модуля полягає у відмові від традиційних серверних архітектур на користь повної клієнтської автономності. Використання технологій маніпулювання PDF-байтами та засобів рендерингу HTML5 Canvas дозволяє генерувати деталізовані звіти безпосередньо у середовищі браузера, що гарантує абсолютну конфіденційність технічних даних та виключає залежність від зовнішніх API-сервісів.

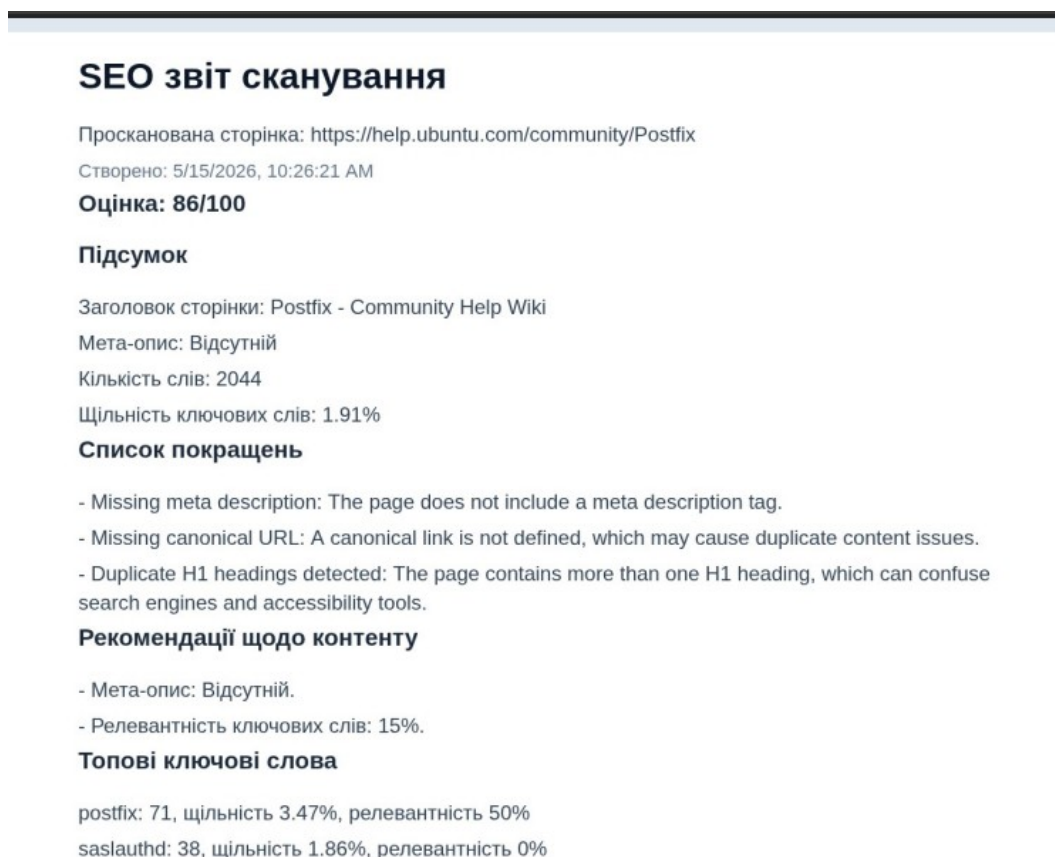


Рисунок 3.3. Приклад згенерованого PDF-звіту

Фундаментальною основою для формування звітності є програмний модуль ``src/shared/scoring.ts``. Застосована у ньому математична модель, що базується на дедуктивній деградації балів, забезпечує ідентичність розрахунків для всіх представлень системи. Верифікація моделі в межах звіту дозволяє користувачеві не просто побачити фінальну цифру, а проаналізувати логіку нарахування штрафних балів за окремими категоріями: технічні дефекти (issues), змішаний контент (mixed content), затримки продуктивності (performance) та некоректні robots-директиви.

Згенерований PDF-звіт (Рисунок 3.3) має чітку структуру, що включає:

Summary сканування з загальними відомостями про перевірену сторінку та час проведення аудиту.

Score breakdown з детальною декомпозицією інтегральної оцінки з візуалізацією вагових коефіцієнтів кожного порушення.

Список покращень у формі структурованого перелік технічних рекомендацій, згрупований за пріоритетністю виправлення.

Content-рекомендації і семантичний аналіз, що включає перелік top keywords та візуалізацію їх розташування у тексті (вибрані підсвічені лексеми).

Реалізація локальної звітності підтверджує високу експлуатаційну ефективність плагіна, перетворюючи його з інструменту оперативної діагностики на повноцінну аналітичну платформу. Такий підхід дозволяє фахівцям інтегрувати результати автоматизованого аудиту безпосередньо в документацію проектів, забезпечуючи доказову базу для проведення подальших робіт з оптимізації веб-ресурсів.

Висновки

Основні наукові та практичні результати роботи полягають у наступному:

1. На основі системного аналізу існуючих засобів моніторингу (Google Lighthouse, Screaming Frog, PageSpeed Insights) ідентифіковано критичну потребу фахівців у інструменті, який поєднує прямий доступ до DOM-структури з повною автономністю обробки даних. Доведено, що використання стандарту Manifest V3 забезпечує необхідний рівень безпеки та швидкодії для сучасних браузерних систем.

2. Спроектовано та реалізовано модульну архітектуру розширення, що базується на технологічному стеку React + TypeScript. Чіткий розподіл функцій між Popup UI, Background Service Worker та Content Scripts дозволив відокремити логіку складних обчислень від інтерфейсної частини, забезпечуючи стабільність роботи плагіна на високонавантажених сторінках.

3. Розроблено та імплементовано алгоритми глибокого аналізу, які охоплюють технічні SEO-метрики (метадані, канонічні URL, robots meta), параметри продуктивності через Performance API та детекцію маркетингових tracking-пікселів. Особливу увагу приділено виявленню детермінованих дефектів, таких як mixed content, дублювання заголовків H1 та відсутність атрибутів доступності зображень.

4. Сформульовано та верифіковано математичну модель** розрахунку інтегральної оцінки якості (Scoring Logic). Застосований принцип дедуктивної деградації балів (старт зі 100 одиниць із наступним відніманням вагових штрафів) забезпечує об'єктивність оцінювання технічного та семантичного стану ресурсу.

5. Реалізовано інноваційний функціонал локальної генерації аналітичних PDF-звітів засобами Canvas та маніпуляцій з байтами без залучення зовнішніх серверів, що гарантує абсолютну конфіденційність даних. Ергономічність системи підтверджена впровадженням інтерактивної візуалізації (підсвічування ключових слів) та підтримкою багатомовності інтерфейсу.

Практична значущість проєкту полягає у створенні цілісного інструментарію для QA-інженерів, SEO-спеціалістів та веб-розробників, який дозволяє суттєво скоротити час на проведення первинного технічного аудиту та підвищити якість кінцевого веб-продукту. Подальші перспективи дослідження можуть бути спрямовані на інтеграцію модулів нейромережевого аналізу контенту для глибшої оцінки релевантності та читабельності текстів.

Список використаних джерел

1. Lighthouse [Електронний ресурс] — Режим доступу: URL: <https://developer.chrome.com/docs/lighthouse/overview>.
2. Screaming Frog SEO Spider. [Електронний ресурс] — Режим доступу: URL: <https://www.screamingfrog.co.uk/seo-spider/>
3. React. Learn React. [Електронний ресурс] — Режим доступу: URL: <https://react.dev/learn>.
4. TypeScript for JavaScript Programmers. [Електронний ресурс] — Режим доступу: URL: <https://www.typescriptlang.org/docs/handbook/typescript-in-5-minutes.html>
5. Vite. [Електронний ресурс] — Режим доступу: URL: <https://v7.vite.dev/>
6. Tailwind CSS [Електронний ресурс] — Режим доступу: URL: <https://v2.tailwindcss.com/docs>.
7. PageSpeed Insights [Електронний ресурс] — Режим доступу: URL: <https://pagespeed.web.dev/>.
8. Шведов І. Дослідження інструментів аналізу продуктивності веб-додатків для ідентифікації та вирішення проблем з продуктивністю. Herald of Khmelnytskyi National University. Technical Sciences. 2024. Т. 347, No 1. P. 595-602. DOI: <https://doi.org/10.31891//2307-5732-2025-347-83>.
9. Помилки інтерфейсу: як їх уникнути та збільшити лояльність клієнтів [Електронний ресурс] — Режим доступу: URL: <https://torgsoft.ua/articles/stati/pomylky-interfejsu/>.
- 10.7 помилок UX, які мимоволі роблять усі дизайнери [Електронний ресурс] — Режим доступу: URL: <https://ideadigital.agency/blog/7-pomilok-ux-yaki-mimovoli-roblyat-usi-dizajneri/>.
11. ТОП-10 помилок UX дизайну [Електронний ресурс] — Режим доступу: URL: <https://www.plerdy.com/ua/blog/top-10-ux-mistake/>.

- 12.20+ помилок при створенні сайту, які слід уникати [Електронний ресурс] — Режим доступу: URL: <https://impulse-design.com.ua/ua/20-oshibok-pri-sozdanii-sajta.html>.
13. Литовченко І. Л. Інтернет-маркетинг. Навчальний посібник – К.: Центр учбової літератури, 2011. – 332 с.
14. Ahrefs — маркетингова ШІ-платформа на базі Big Data. [Електронний ресурс] — Режим доступу: URL: <https://ahrefs.com/uk>.
15. Plerdy [Електронний ресурс] — Режим доступу: URL: <https://www.plerdy.com/>.
16. Crazy Egg [Електронний ресурс] — Режим доступу: URL: <https://www.crazyegg.com/>.
17. Gtmatrix [Електронний ресурс] — Режим доступу: URL: <https://gtmatrix.com/>.
18. Технічна оптимізація сайту: типові помилки та їх усунення [Електронний ресурс] — Режим доступу: URL: <https://web-promo.ua/ua/blog/tehnicna-optymizacziya-sajtu-typovi-pomylky-ta-yih-usunennya/>.