

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»  
Навчально-науковий інститут прикладного системного аналізу  
Кафедра системного проектування**

До захисту допущено:  
Завідувач кафедри  
\_\_\_\_\_ Вадим МУХІН  
« \_\_\_\_ » \_\_\_\_\_ 2023 р.

**Дипломна робота**  
на здобуття ступеня бакалавра  
за освітньо-професійною програмою  
«Інтелектуальні сервіс-орієнтовані розподілені обчислювання»  
зі спеціальності 122 "Комп'ютерні науки"  
на тему: «Пошукова оптимізація веб-сайтів для підвищення  
продуктивності та швидкості роботи»

Виконав:  
студент ІV курсу, групи ДА-91  
Марко Вадим Вячеславович \_\_\_\_\_

Науковий керівник:  
Доцент, к.т.н.  
Безносик Олександр Юрійович \_\_\_\_\_

Консультант з економіки:  
Доцент, к.е.н.  
Рощина Надія Василівна \_\_\_\_\_

Рецензент:  
Доц. каф. ЦТЕ НН ІАТЕ, к.т.н. доц.  
Шаповалова Світлана Ігорівна \_\_\_\_\_

Засвідчую, що у цій дипломній роботі  
немає запозичень з праць інших авторів  
без відповідних посилань.  
Студент \_\_\_\_\_

**Національний технічний університет України**  
**«Київський політехнічний інститут імені Ігоря Сікорського»**  
**Навчально-науковий інститут прикладного системного аналізу**  
**Кафедра системного проектування**

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 122 «Комп'ютерні науки»

Освітньо-професійна програма – «Інтелектуальні сервіс-орієнтовані розподілені обчислювання»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ В. Є. Мухін

«\_\_» \_\_\_\_\_ 2023 р.

**ЗАВДАННЯ**

**на дипломну роботу студенту**

**Марку Вадиму Вячеславовичу**

1. Тема роботи: «Пошукова оптимізація веб-сайтів для підвищення продуктивності та швидкості роботи», керівник роботи Безносик Олександр Юрійович, доцент, к.т.н., затверджені наказом по університету від «30» травня 2023 р. No 2065-с.
2. Термін подання студентом роботи – 14 червня 2023 р.
3. Вихідні дані до роботи: система WordPress 6.2.1, PHP, JavaScript, HTML, CSS, Google Lighthouse, середовище розробки WebStorm, Chrome DevTools
4. Зміст роботи:
  1. Дослідження основ пошукової оптимізації
  2. Аналіз наявних проблем веб-сайту
  3. Реалізація методів пошукової оптимізації
  4. Огляд результатів оптимізації веб-сайту
  5. Функціонально-вартісний аналіз програмного продукту

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо):

1. Презентація до захисту роботи

6. Консультанти розділів роботи.

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
Економічний	Рощина Н. В.		

7. Дата видачі завдання \_\_\_\_\_

#### Календарний план

№ з/п	Назви етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання	31.01.2023	
2	Збір інформації	06.04.2023	
3	Ознайомлення з літературою	16.04.2023	
4	Дослідження основ пошукової оптимізації	30.04.2023	
5	Реалізація методів оптимізації для підвищення продуктивності веб-сайту	14.05.2023	
6	Аналіз отриманих результатів оптимізації	28.05.2023	
7	Розробка економічної частини	05.06.2023	
8	Оформлення дипломної роботи	12.06.2023	
9	Отримання допуску до захисту та подача роботи в ДЕК	15.06.2023	

Студент

Вадим МАРКО

Керівник

Олександр БЕЗНОСИК

## АНОТАЦІЯ

до дипломної роботи Марка Вадима Вячеславовича на тему  
«Пошукова оптимізація веб-сайтів для підвищення продуктивності та  
швидкості роботи»

У сучасному цифровому світі, де конкуренція в Інтернеті зростає, пошукова оптимізація веб-сайтів стає необхідною складовою стратегії просування товарів та послуг. Завдяки пошуковій оптимізації підприємства та установи можуть забезпечити видимість своєї онлайн-присутності, залучити більше цільової аудиторії та покращити свої результати.

Метою даної дипломної роботи є дослідження методів пошукової оптимізації веб-сайтів та їх практичного застосування на прикладі веб-сайту кафедри.

У роботі були розглянуті основи пошукової оптимізації, різноманітні способи покращення рейтингу веб-сайту в пошукових системах та варіанти їх реалізації. Також для сайту кафедри був здійснений аудит показників та діагностика проблем, що впливають на рейтинг веб-сайту в пошукових системах. Далі була проведена робота по усуненню всіх знайдених недоліків і оцінка отриманих результатів.

Для оптимізації веб-сайту кафедри використовувалися методи пошукової оптимізації на сторінці. В результаті їх впровадження оцінка сайту пошуковими системами стала кращою, а отже вірогідність потрапляння сайту на вищі позиції на сторінках пошукових видач збільшиться.

**Загальний обсяг роботи: 75 с., 38 рис., 8 таблиць, 20 джерел.**

**Ключові слова:** пошукова оптимізація, пошукові системи, оптимізація на сторінці, пошукові павуки, індексація, посилання, веб-сайт, веб-сторінка

## ABSTRACT

for the bachelor's thesis of Marko Vadym Viacheslavovych on “Search Engine Optimization of Websites for Improving Performance and Speed”

In today's digital world, where online competition is rising, search engine optimization of websites is becoming a necessary component of a strategy for promoting products and services. Through search engine optimization, businesses and institutions can ensure the visibility of their online presence, attract more target audiences, and improve results.

The purpose of this thesis is to study the methods of search engine optimization of websites and their practical application on the example of the department's website.

The paper examines the basics of search engine optimization, various ways to improve website rankings in search engines, and options for their implementation. Also, an audit of indicators and diagnostics of problems affecting the ranking of the website in search engines was carried out for the department's website. Next, the work was carried out to eliminate all the shortcomings found and evaluate the results.

On-page search engine optimization methods were used to optimize the department's website. As a result of their implementation, the evaluation of the site by search engines has become better, and therefore the probability of getting the site to higher positions on the pages of search results will increase.

**The total volume of work: 75 p., 38 figures, 8 tables, 20 references.**

**Keywords:** search engine optimization, search engines, on-page optimization, web crawlers, website, webpage, indexing, links

## ЗМІСТ

ВСТУП.....	8
<b>1 ОСНОВИ ПОШУКОВОЇ ОПТИМІЗАЦІЇ .....</b>	<b>9</b>
1.1 Цінність зворотних посилань.....	9
1.2 Павуки пошукових систем .....	10
1.3 Тактика чорного SEO.....	10
1.4 Методи оптимізації поза сторінкою .....	11
1.5 Методи оптимізації на сторінці .....	11
1.6 Роль контенту в пошуковій оптимізації.....	12
1.7 Використання ключових слів.....	12
1.8 Чисті URL-адреси .....	12
1.9 Технічне SEO.....	13
1.10 Оптимізація коду сторінки.....	14
1.11 Огляд Lighthouse .....	15
1.12 Створення звітів Lighthouse .....	16
1.13 Оцінка продуктивності .....	16
1.14 Оцінка доступності .....	18
1.15 Оцінка найкращих практик .....	19
1.16 SEO-інформація.....	20
1.17 Висновки до розділу .....	20
<b>2 АНАЛІЗ НАЯВНИХ ПРОБЛЕМ ВЕБ-САЙТУ .....</b>	<b>21</b>
2.1 Огляд проблем продуктивності .....	22
2.2 Огляд проблем доступності .....	25
2.3 Огляд найкращих практик.....	27
2.4 Огляд проблем SEO-метрик.....	29
2.5 Висновки до розділу .....	32
<b>3 РЕАЛІЗАЦІЯ ПОШУКОВОЇ ОПТИМІЗАЦІЇ ТА АНАЛІЗ РЕЗУЛЬТАТІВ</b>	<b>33</b>
3.1 Стиснення текстових файлів.....	33
3.2 Мінімізація кумулятивного зсуву розмітки .....	34
3.3 Удосконалення доступності.....	36
3.4 Усунення змішаного контенту.....	37

3.5	Створення мета-описів .....	39
3.6	Усунення посилань, що не індексуються .....	40
3.7	Створення XML-карти веб-сайту .....	41
3.8	Впровадження структурованої розмітки даних Schema.org .....	44
3.9	Lighthouse аудит результатів оптимізації .....	46
3.10	Semrush аудит результатів оптимізації .....	48
3.11	Висновки до розділу .....	51
4	ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОДУКТУ .....	53
4.1	Постановка задачі проектування .....	53
4.2	Обґрунтування функцій програмного продукту .....	54
4.3	Обґрунтування системи параметрів програмного продукту .....	57
4.4	Аналіз експертного оцінювання параметрів .....	59
4.5	Аналіз рівня якості варіантів реалізації функцій .....	62
4.6	Економічний аналіз варіантів розробки ПП .....	64
4.7	Вибір кращого варіанту ПП техніко-економічного рівня .....	69
4.8	Висновки до розділу .....	70
	ВИСНОВКИ .....	71
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	73

## ВСТУП

Пошук інформації завжди був невід'ємною частиною людського життя. З розвитком технологій з'являлися нові джерела інформації, що потребували нових методів навігації по ним. Сьогодні найбільшим джерелом інформації є мережа інтернет, що за даними на 2023 рік складається близько з 1.13 мільярдів веб-сайтів та більш ніж 50 мільярдів веб-сторінок [4]. Зрозуміло, що для такої кількості інформації потрібні інструменти, що дозволить швидко знайти те, що потрібно користувачу. Такими інструментами є пошукові системи, а їх програмну компоненту називають пошуковими рушіями (search engines).

Пошукові системи для відображення корисних користувачу результатів використовують алгоритми, що оцінюють їх доречність відповідно до введеного запиту. Проте цей процес починається ще до того, як користувач надсилає свій запит. Формування результатів пошукової видачі розпочинається з моменту розміщення веб-сайту в мережі Інтернет і його оцінки.

Оптимізація веб-сайтів задля покращення їх рейтингу в пошукових системах називається пошуковою оптимізацією (search engine optimization). Вона складається з великої кількості методів, основною метою яких є збільшення трафіку веб-сайту, завдяки підвищенню його привабливості для користувачів та пошукових систем.

# 1 ОСНОВИ ПОШУКОВОЇ ОПТИМІЗАЦІЇ

Через запеклу конкуренцію з кожним днем все більше компаній звертають увагу на SEO. Добре реалізована SEO-стратегія дозволяє випередити конкурентів та опинитися на перших позиціях в пошуковій видачі. Алгоритми пошукових систем часто оновлюються для покращення результатів видачі та боротьби зі спамом. Беручи до уваги велику кількість факторів, які враховуються при ранжуванні веб-сайтів, встигати слідкувати за оновленнями пошукових систем є нетривіальною задачею. Серед цих факторів можна, наприклад, назвати якість та кількість зворотних посилань і ключових слів, швидкість роботи веб-сайту, зручність використання на мобільних пристроях, доступність, тощо. До того ж деталі роботи алгоритмів пошуку приховуються їхніми розробниками, щоб позбутися шкідливих маніпуляцій зі сторони утримувачів та розробників веб-сайтів. Попри всі складності дуже важливо слідкувати за змінами в пошукових системах, адже ці зміни роблять веб-простір більш корисним, зручним та доступним для користувачів.

## 1.1 Цінність зворотних посилань

Велика кількість вхідних посилань з інших веб-сайтів часто вважається показником його популярності та авторитетності. Ці вхідні посилання, які також називають зворотними, сигналізують пошуковим системам про те, що веб-сайт, на який посилаються, заслуговує на довіру і є цінним. Пошукові системи, такі як Google, розглядають зворотні посилання як вирішальний фактор ранжування, оскільки вони вказують на впливовість веб-сайту. Однак не всі зворотні посилання є рівноцінними. Коли авторитетні веб-сайти посилаються на певний сайт, це свідчить про те, що його контент або ресурси також є авторитетними та релевантними. Високоякісні зворотні посилання з таких джерел мають більшу вагу і вплив на видимість та рейтинг сайту. Поява сильного профілю зворотних посилань вимагає поєднання створення цінного

контенту, зміцнення відносин з іншими веб-сайтами та використання ефективних стратегій для залучення органічних посилань.

## **1.2 Павуки пошукових систем**

Пошукові павуки – це програми, які розроблені для постійного сканування та індексування сторінок в мережі та передачі цих даних пошуковим системам. Пошукові павуки систематично відвідують веб-сайти, переходять по посиланням та аналізують структуру, релевантність та контекст веб-сторінок. Вони збирають інформацію з усіх доступних елементів сторінок для визначення рейтингу веб-сайту у пошуковій видачі. Павуки відіграють ключову роль в підтриманні свіжості індексу пошукових систем. Фахівці з SEO звертають особливу увагу на те, щоб веб-сайти добре сканувались та індексувались пошуковими павуками, адже вони слугують очами та вухами для пошукових систем.

## **1.3 Тактика чорного SEO**

Деякі фахівці використовують неетичні методи, відомі як "чорне SEO", щоб маніпулювати пошуковим ранжуванням. Подібні методи порушують правила пошукових систем та мають на меті досягнути хороших показників оптимізації, обманюючи пошукові системи. Ці тактики включають:

- Перенасичення ключовими словами;
- Прихований текст і посилання;
- Ферми посилань та платні посилання;
- Маскування контенту;
- Використання чужого контенту;
- Створення версії сайту для пошукових систем.

Однак пошукові системи карають веб-сайти, які використовують чорні методи, що призводить до зниження рейтингу або навіть до зникнення з

результатів пошуку.

#### **1.4 Методи оптимізації поза сторінкою**

Сам процес оптимізації веб-сайтів складається з багатьох елементів і рідко буває одноразовим. Виділяють два основних типи оптимізації: поза сторінкою та на сторінці. Обидва варіанти є надзвичайно важливими для просування сайту і, відповідно, залучення нових клієнтів. Пошукова оптимізація сайту поза сторінкою включає методи, головним призначенням яких є покращення сприйняття сайту користувачами та пошуковими системами з точки зору довіри, авторитету, популярності та релевантності.

Розвиваючи SEO поза сторінкою, можна досягнути значного повернення уваги до веб-сайту, в наслідок чого інші сайти та користувачі будуть посилатися на його контент. Прикладами оптимізації поза сторінкою є:

- Соціальний медіа маркетинг;
- Онлайн реклама та спонсоровані посилання;
- Спільні партнерські проекти;
- Підтримка відгуків клієнтів.

Звідси можна зробити висновок, що SEO поза сторінкою, націлене на реалізацію маркетингових стратегій та отримання результатів в довготривалій перспективі. Метою ж цієї роботи є покращення технічних аспектів веб-сайту, що впливають на його продуктивність та швидкість, тому надалі в цій роботі методи оптимізації поза сторінкою розглядатися не будуть.

#### **1.5 Методи оптимізації на сторінці**

Оптимізація на сторінці (on-page optimization) є важливою складовою пошукової оптимізації, яка полягає у внесенні змін безпосередньо до сторінок веб-сайту, включаючи його контент та технічну складову. При реалізації

методів пошукової оптимізації на сторінці, найбільше значення має те, наскільки добре пошуковий павук оцінить кожен сторінку веб-сайту. Тому при реалізації її технічної складової особливу увагу надають доступності сторінок для сканування.

## **1.6 Роль контенту в пошукової оптимізації**

Головною частиною веб-сайтів, що забезпечує їх привабливість для користувача є його контент. Відповідно для пошукових систем релевантний та цінний контент також має найбільше значення. Якісний контент окрім того, що сам по собі приваблює користувачів, ще й дозволяє пошуковим системам розуміти мету веб-сайту і додатково його просувати. Хороший контент приваблює зворотні посилання з інших веб-сайтів, що ще сильніше зміцнює авторитетність та довіру до ресурсу, а також сприяє позитивному користувацькому досвіду, відповідаючи на питання та вирішуючи проблеми.

## **1.7 Використання ключових слів**

Проведення комплексного дослідження ключових слів є важливим аспектом оптимізації веб-сайту для пошукових систем та ефективного охоплення цільової аудиторії. Цей процес передбачає визначення та аналіз релевантних ключових слів, які відповідають змісту веб-сайту і часто використовуються цільовою аудиторією. Ключові слова, окрім, звісно, основного контенту, можна додавати і в інші елементи сторінок, такі як заголовки, мета-теги та URL-адреси.

## **1.8 Чисті URL-адреси**

Важливе значення має оптимізація URL-адрес сторінок. Вони мають бути коротким, описовим і містити ключові слова. Також корисно уникати

використання непотрібних символів або цифр в URL-адресах.

На рисунку 1.1 URL містить параметри і спеціальні символи. Такі URL-адреси не є зручними і можуть бути складними для інтерпретації як користувачами, так і пошуковими системами. Вони не містять описових ключових слів і надають мало інформації про вміст сторінки.

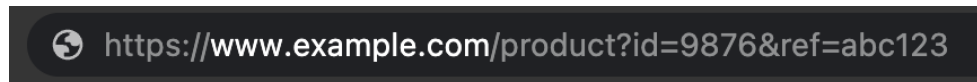


Рисунок 1.1 – Приклад поганої URL-адреси

На рисунку 1.2 URL-адреса чиста, описова і легка для розуміння. Вона містить релевантні ключові слова(продукти, електроніка, смартфони), які вказують на зміст сторінки. Ця URL-адреса зручна для користувача, забезпечує чітку ієрархію і, швидше за все, буде проіндексована і добре оцінена пошуковими системами.

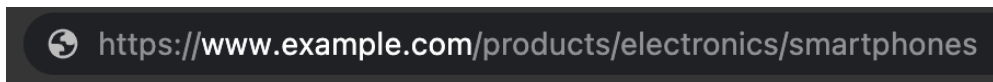


Рисунок 1.2 – Приклад хорошої URL-адреси

## 1.9 Технічне SEO

Технічне SEO передбачає оптимізацію технічних аспектів веб-сайту для покращення його продуктивності та доступності для пошукових систем. Методи технічного SEO включають, наприклад, забезпечення належної індексації веб-сторінок, виправлення непрацюючих посилань та впровадження структурованої розмітки даних. Технічна SEO-оптимізація допомагає пошуковим системам краще розуміти і оцінювати сайт. Серед звичних методів технічного SEO можна назвати:

- Впровадження структурованої розмітки даних за стандартом Schema.org [11];

- Створення XML-карт для надання пошуковим системам інформації про структуру сторінок веб-сайту;
- Оптимізація файлу robots.txt для контролю сканування та індексації пошуковими системами;
- Реалізація динамічних URL, що індексуються пошуковими системами.

## 1.10 Оптимізація коду сторінки

Вихідний код має велике значення в ранжуванні веб-сайту пошуковими системами. Він вказує на структуру сторінок, надає мета-інформацію щодо їх вмісту, а його якість може безпосередньо впливати на швидкість та правильність роботи.

1. Правильне використання HTML заголовків – заголовки повинні бути розташовані ієрархічно, починаючи з заголовка H1 (який зазвичай використовується для заголовка сторінки), потім H2, H3 і так далі. Використання більш ніж одного заголовка H1 на сторінці вважається поганою практикою, тому цього потрібно уникати. Також важливо використовувати ключові слова в назвах, оскільки вони важливі для SEO.

```
<h1>Головний заголовок сторінки</h1>  
<h2>Заголовок другого рівня</h2>  
<h3>Заголовок третього рівня</h3>
```

Рисунок 1.3 – Приклад ієрархії заголовків

2. Правильне використання мета-тегів – мета-теги надають додаткову інформацію про сторінку для пошукових систем, що допомагає підвищувати її видимість та налаштувати результати пошуку. Тому потрібно використовувати мета-теги, такі як мета-опис і мета-ключові

слова, щоб вказати короткий опис сторінки та важливі ключові слова. Важливо, щоб мета-опис був привабливим і актуальним для користувачів.

```
<!--приклад мета-опису-->  
<meta name="description" content="Це опис сторінки з використанням ключових слів.">  
<!--приклад мета-ключових слів-->  
<meta name="keywords" content="ключове_слово_1, ключове_слово_2, ключове_слово_3">
```

Рисунок 1.4 – Приклад мета-опису та мета-ключових слів

3. Якісний JavaScript код – погано написаний JavaScript код може призвести до суттєвих затримок в роботі сайту. Проблеми в роботі JavaScript коду можуть виникати через неправильну взаємодію з об'єктною моделлю документа(DOM), надмірну кількість HTTP-запитів, відсутність мінімізації коду, неефективну роботу програмних циклів, неправильне керування ресурсами або відсутністю кешування.

## 1.11 Огляд Lighthouse

Google Lighthouse – це інструмент з відкритим вихідним кодом, який перевіряє та оцінює веб-сайти за різними параметрами. Він надає вичерпні звіти про ефективність, доступність, найкращі практики та SEO. Lighthouse генерує дієві поради та визначає проблемні місця веб-сторінок, що допомагає оптимізувати веб-сайт. Плагін використовує набір попередньо визначених метрик для оцінки різних аспектів ефективності веб-сайту. Ці показники базуються на галузевих стандартах і найкращих практиках, забезпечуючи об'єктивну оцінку продуктивності веб-сайту та користувацького досвіду. Lighthouse моделює реальні сценарії, імітуючи роботу веб-сайту на різних пристроях, в різних мережевих умовах і за різної швидкості з'єднання. Це дозволяє розробникам покращити роботу веб-сайту для різних користувачів, гарантуючи, що він буде добре працювати в різних середовищах. Також

великим плюсом є інтегрованість в Chrome DevTools, що робить його легкодоступним під час процесу розробки та налагодження.

## 1.12 Створення звітів Lighthouse

Доступ до плагіну Google Lighthouse можна отримати безпосередньо через Chrome DevTools або встановивши розширення для браузера. Після встановлення Lighthouse дозволяє проводити аудит окремих веб-сторінок. Для цього треба в головному меню створення Lighthouse звіту обрати режим, тип пристрою та категорії метрик, а потім натиснути кнопку «Проаналізувати завантаження сторінки». Lighthouse пропонує наступні режими аналізу веб-сторінок:

- Навігація – основний режим, який дозволяє імітувати навігацію користувача веб-сайтом. Цей режим дає уявлення про те, як веб-сайт працює під час реальної взаємодії з користувачем, допомагаючи визначити області, які потрібно покращити з точки зору швидкості реакції на його дії.
- Часовий проміжок – режим, який дозволяє проводити аудит продуктивності протягом певного періоду часу. Він дає уявлення про те, як веб-сайт працює та поводить себе протягом тривалого періоду. Цей режим особливо корисний для моніторингу змін продуктивності та виявлення проблем, що виникають з часом.
- Знімок – режим, що дозволяє оцінити продуктивність в конкретний момент роботи веб-сайту. Корисний для швидкої оцінки або під час тестування певних змін, що були внесені на веб-сайт.

## 1.13 Оцінка продуктивності

Розділ продуктивності Google Lighthouse оцінює швидкість веб-сайту та час завантаження. Він враховує наступні показники:

- First contentful paint(FCP) – показник, що вказує на час який потрібен для того, щоб закінчився рендеринг першого елементу сторінки. Він вказує на те, чи побачить користувач щось на сторінці до того як закінчиться її завантаження.
- Largest contentful paint(LCP) – показник, що вказує на час, який потрібен найбільшому елементу на сторінці для того, щоб стати видимим для користувача. Він дозволяє оцінити наскільки швидко завантажуються головний контент веб-сайту.
- Total blocking time(TBT) – показник, що вимірює загальну кількість часу від закінчення рендерингу першого елементу на сторінці до моменту коли головний потік виконання буде вільним для того, щоб відповідати на ввід користувача. Він визначає моменти під час завантаження сторінки, які призводять до затримок в зворотньому зв'язку з користувачем, що виникають під час виконання JavaScript коду.
- Time to interactive(TTI) – показник, що вимірює час який потрібен щоб повністю завантажити сторінку та вказує на момент коли сторінка починає реагувати на всі дії користувача на ній.
- Cumulative layout shifts(CLS) – показник, що вимірює кількість неочікуваних змін в розмітці під час завантаження сторінки. Він вказує на те чи користувач побачить “стрибки” контенту на сторінці під час її загрузки.

Досить велика кількість факторів можуть впливати на ці показники. Перш за все це час, який потрібен серверу для відповіді на запит користувача. Повільний час відгуку сервера може затримати доставку початкового контенту(FCP), найбільшого елемента контенту(LCP) і доступність ресурсів, необхідних для інтерактивності(TBT і TTI). Наступним фактором є розмір сторінки та оптимізація ресурсів. Загальний розмір веб-сторінки, включаючи HTML, CSS, JavaScript та медіа файли відіграють вирішальну роль у визначенні FCP, LCP, TBT і TTI. Великі розміри файлів і надмірні запити на

ресурси можуть збільшити час, необхідний для рендерингу початкового контенту(FCP), завантаження найбільшого елемента контенту(LCP) і обробки необхідних скриптів для інтерактивності(TBT і TTI). Ще одним фактором є неефективне виконання JavaScript коду, що може спричинити затримки в рендерингу та інтерактивності. Складний JavaScript код, який вимагає тривалої обробки, може збільшити TBT і TTI, оскільки браузеру потрібно більше часу, щоб виконати скрипти і зробити сторінку інтерактивною. Послідовність дій, які виконує браузер для відображення веб-сторінки, також впливає на FCP, LCP і TTI. Оптимізація критичного шляху рендерингу шляхом мінімізації ресурсів, що блокують рендеринг, використання асинхронного завантаження та оптимізації доставки ресурсів може покращити ці показники. Швидкість і якість інтернет-з'єднання користувача впливають на FCP, LCP, TBT і TTI. Користувачі з повільним або нестабільним інтернет-з'єднанням можуть відчувати затримки в отриманні ресурсів, що призводить до збільшення часу завантаження і впливає на ці показники продуктивності. Впровадження належних методів кешування і стиснення для статичних ресурсів може покращити FCP і LCP. Кешування дозволяє браузеру зберігати і повторно використовувати ресурси, зменшуючи потребу в повторних запитах. Стиснення зменшує розмір файлів, що призводить до швидшого завантаження та рендерингу. Складність об'єктної моделі документа(DOM) може впливати на TTI. Складна структура DOM, спричинена надмірною вкладеністю, великою кількістю елементів або неефективним кодом, може збільшити час, необхідний для того, щоб сторінка стала повністю інтерактивною.

### **1.14 Оцінка доступності**

Lighthouse оцінює доступність веб-сайту, щоб переконатися, що ним можуть користуватися люди з обмеженими можливостями. Він звертає увагу на наступні показники:

- Атрибути ARIA(Accessible Rich Internet Applications) – це атрибути HTML-елементів сторінок веб-сайтів, які дозволяють контенту бути більш доступним для людей з обмеженими можливостями, що використовують допоміжні технології такі як зчитувачі екрану.
- Коефіцієнт контрастності – для забезпечення хорошої видимості тексту на сторінці проводиться перевірка його контрастності в порівнянні з фоном.
- Доступність клавіатури – проводиться перевірка на те, чи всі інтерактивні елементи на сторінці можуть бути активовані користувачем за допомогою клавіатури. Таким чином забезпечується доступність для людей, що не можуть використовувати миші чи інші вказівні пристрої.
- Alt-атрибути зображень – проводиться перевірка на те, чи всі HTML елементи зображень мають атрибут «alt». В alt-атрибуті вказують опис того, що показано на зображенні. Таким чином забезпечується доступність для користувачів з вадами зору.
- Заголовки та структура – цей показник потрібен для оцінки правильності використання заголовків та структури сторінки. Правильне використання заголовків та хороша структура полегшують користування веб-сайтом людям, що користуються зчитувачами екранів
- Управління фокусом – цей показник забезпечує наявність фокус-ефекту(ефекту виділення) на активних елементах сторінки. Людям що використовують клавіатуру для взаємодії це допомагає знайти активний елемент на сторінці.
- 

### **1.15 Оцінка найкращих практик**

У розділі "Найкращі практики" веб-сайт перевіряється на відповідність галузевим стандартам. Перевіряються заходи безпеки, такі як використання HTTPS, правильні конфігурації заголовків і коректний код HTML і CSS.

Lighthouse також надає рекомендації щодо оптимізації завантаження ресурсів, адаптивного дизайну та використання функцій прогресивних веб-додатків(PWA), гарантуючи, що веб-сайт відповідає сучасним найкращим практикам.

### **1.16 SEO-інформація**

Google Lighthouse включає SEO-аудит, який аналізує різні аспекти пошукової оптимізації сайту. Він вивчає метадані, можливість сканування пошуковими системами, зручність для мобільних пристроїв та багато іншого. SEO-аудит висвітлює сфери, які потребують покращення, дозволяючи оптимізувати SEO-елементи веб-сайту та підвищити його у рейтингах пошукових систем.

### **1.17 Висновки до розділу**

Отже, пошукова оптимізація є невід'ємною частиною сучасного веб-простору. Її принципи лежать в основі успіху всіх найпопулярніших веб-сайтів. Розуміння цих засад є першим кроком до того, аби створювати власні веб-сайти, якими користуватимуться мільйони людей. Основні принципи SEO включають дослідження та оптимізацію ключових слів, оптимізацію на сторінці, технічну оптимізацію, створення якісного контенту, оптимізацію користувацького досвіду та створення посилань. Також у сфері пошукової оптимізації дуже важливим є вміння користуватися інструментами оцінки роботи веб-сайтів у пошукових системах.

## 2 АНАЛІЗ НАЯВНИХ ПРОБЛЕМ ВЕБ-САЙТУ

В цьому розділі на прикладі сайту кафедри були розглянуті проблеми пошукової оптимізації та причини їх виникнення. Для цього, використовуючи плагін Lighthouse, перевіримо показники пошукової оптимізації основних сторінок веб-сайту кафедри та занесемо їх числові значення в таблицю 2.1. Оцінки представлені за 100-бальною шкалою, де 0 це найгірший результат, а 100 – найкращий.

Таблиця 2.1 – Результати аудиту сторінок за допомогою Lighthouse

Сторінка	Продуктивність	Доступність	Найкращі практики	SEO
home/	76	96	83	85
news/	75	95	92	80
about-us/official-information/	75	94	92	85
about-us/history/	74	94	92	85
about-us/teachers/	76	94	92	85
admission/bachelor/	74	94	92	85
admission/magistracy/	74	94	92	83
admission/second-degree/	76	94	92	85
admission/correspondence-tuition/	76	94	92	85
admission/postgraduate/	62	94	92	85
admission/employment/	76	94	92	85
admission/why-sd/	76	95	75	85
student/osvitni-progamy/	77	94	92	85
student/navchalnij-plan/	76	94	92	85
admission/catalogs/	100	94	92	85
student/syllabuses/	100	95	92	85
student/rozklad-zanjat/	100	94	92	85
student/navchalna-dokumentacija/	100	94	92	85
student/opytuvannia/	100	94	92	85
student-life/polozhennja-pro-imenni-stipendii/	99	94	75	85

Продовження таблиці 2.2

scientific-school/	100	94	92	85
science/specializovana-vchena-rada/	96	91	75	85
science/naukov-naprjamki/	99	94	92	85
science/dosjagnennja/	100	94	75	85
science/proekti-shhovikonujutsja/	100	94	92	85
category/konferencii/	100	94	92	85
science/publkacii/	100	94	75	77
science/studentsk-publac/	100	95	75	85
science/diplomni-proekti/	100	94	75	84
science/temi-doslidzen/	99	94	75	85
science/temi-doslidzen-aspirantiv/	99	94	92	77
zahist-vypusknih-robot/	99	94	92	85
partnership/international-science-cooperation/	100	94	92	77
about-us/contacts/	100	91	83	85

Можемо бачити, що сайт потребує заходів з оптимізації, особливо в плані підвищення продуктивності та SEO-метрик.

## 2.1 Огляд проблем продуктивності

На мою думку, показники продуктивності Lighthouse не завжди репрезентативні. Це пояснюється тим, що при визначенні продуктивності роботи оцінюється фактичний час завантаження елементів сторінки. Таким чином швидкість і стабільність інтернет-підключення може напряму впливати на результати аудиту. Незважаючи на цей факт, Lighthouse надає поради щодо покращення не тільки на основі часових метрик, але й оцінки коду веб-сайту.

Як можемо бачити на рисунку 2.1, Lighthouse пропонує стискати текстові файли. Стиснення тексту – це метод, який використовується для зменшення розміру текстових файлів, що дозволяє ефективно зберігати та передавати їх. Існує кілька популярних алгоритмів і форматів файлів, що використовуються для стиснення тексту, зокрема gzip, deflate і brotli.

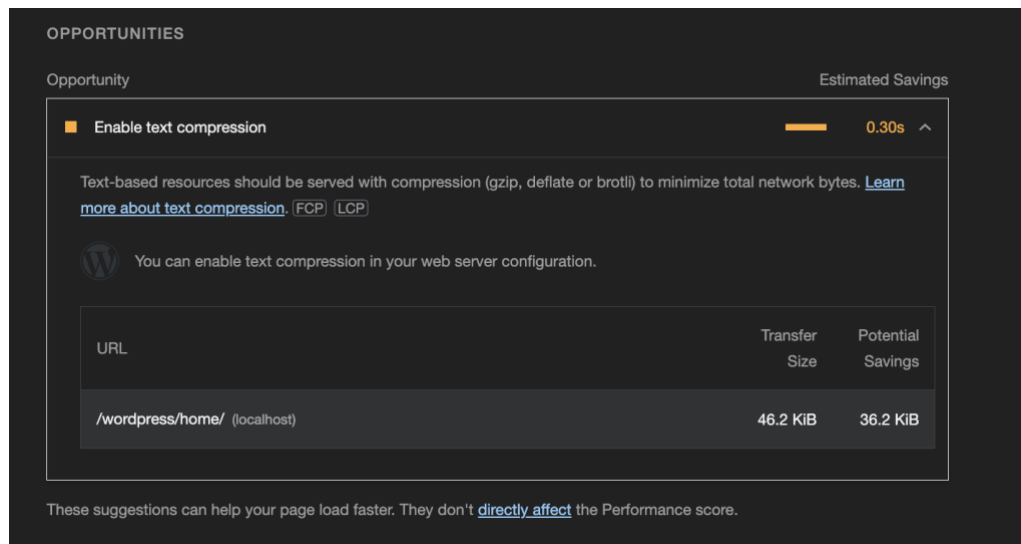


Рисунок 2.1 – Порада щодо стиснення текстових файлів

Deflate – це алгоритм стиснення без втрат, який видаляє надлишковість текстових даних, замінюючи повторювані послідовності посиланнями. Gzip, з іншого боку, це формат файлів, який використовує стиснення deflate для зменшення розміру файлів. При стисненні текстового файлу за допомогою gzip алгоритм спочатку застосовує метод стиснення deflate до вхідного тексту. Це передбачає виявлення повторюваних шаблонів і заміну їх більш короткими представленнями. Gzip також включає додаткові метадані та контрольну суму для забезпечення цілісності даних під час стиснення та розпакування.

Brotli – це новітній алгоритм стиснення, розроблений компанією Google. Він призначений для забезпечення більш високого ступеня стиснення порівняно з gzip і deflate при збереженні швидкості декомпресії. Порівняно з gzip і deflate, brotli зазвичай пропонує вищий ступінь стиснення і кращу продуктивність. Однак, компроміс полягає в тому, що для стиснення і

розпакування brotli вимагає більше обчислювальних ресурсів, що може вплинути на час обробки.

Наступною проблемою, як можемо побачити на рисунку 2.2, є відсутність явного значення ширини та висоти зображень на сторінці. Вона є одною з першопричин поганої оптимізації кумулятивного зсуву розмітки(CLS).

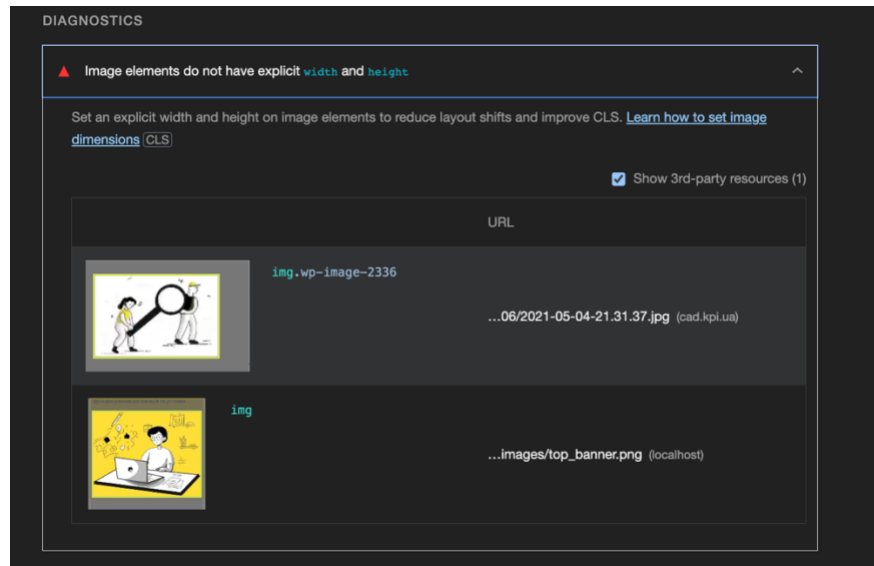


Рисунок 2.2 – Попередження про відсутність явних розмірів зображень

Також серед основних причин виникнення проблем з CLS можна назвати:

- вбудовані елементи;
- реклама;
- iframes;
- динамічно введений контент;
- веб-шрифти.

Вбудовані елементи, реклама, iframes та динамічно введений контент без явно заданих розмірів спочатку не враховуються при визначенні положень елементів на сторінці. Після загрузки вони можуть штовхати сусідні елементи, що спричиняє «стрибки» контенту на сторінці. Це негативно впливає на

користувацький досвід, бо користувачу потрібно заново шукати те місце, на якому він зупинився, що відволікає та заважає. Така ж проблема виникає і при завантаженні веб-шрифтів. Якщо не вказати резервний системний шрифт, текст може займати різну кількість місця до і після завантаження веб-шрифту, що може призвести до зміщення розмітки.

Числове значення кумулятивного зсуву розмітки в Lighthouse для забезпечення хорошого користувацького досвіду не повинен перевищувати 0.1 для щонайменше 75% відвідувачів веб-сторінки [5].



Рисунок 2.3 – Часова оцінка кумулятивного зсуву розмітки [5]

## 2.2 Огляд проблем доступності

Найпоширенішими проблемами доступності на веб-сайтах з поганою пошуковою оптимізацією є звісно відсутність атрибутів, що покращують доступність, та неправильне використання HTML-тегів. На рисунку 2.4 зображений фрагмент звіту про доступність веб-сайту, де вказано, що посилання мають мати прив'язаний текст, який описує їх кінцеве направлення та мету.

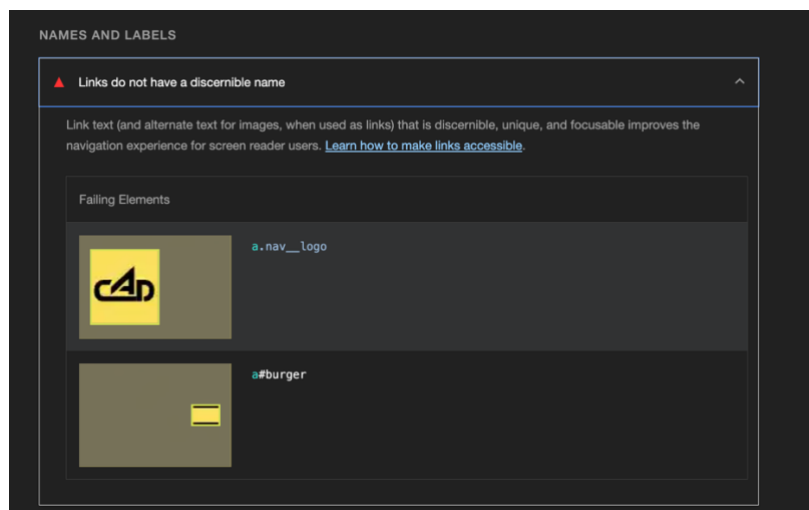


Рисунок 2.4 – Попередження про відсутність назви посилань

Так само, як і атрибути ARIA, атрибут title для iframe(рис 2.5) слугує для надання додаткової інформації та покращення доступності на веб-сторінках. Атрибути ARIA, такі як aria-label, aria-labelledby та aria-describedby, використовуються для покращення доступності різних елементів шляхом надання альтернативного або додаткового тексту, який зчитується допоміжними технологіями. З іншого боку, атрибут title для iframe дозволяє розробникам надавати заголовок або опис вмісту iframe. Хоча обидва типи атрибутів сприяють підвищенню доступності, вони відрізняються за своїм використанням.

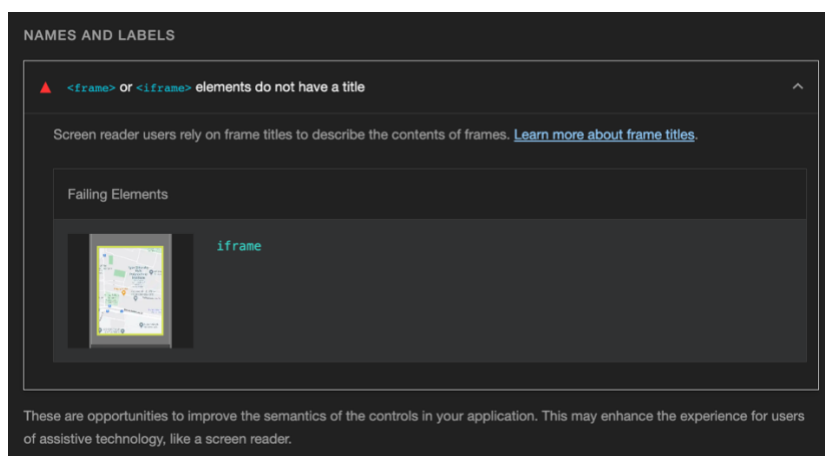


Рисунок 2.5 – Попередження про відсутність атрибуту-заголовку в iframe

Пошукові системи надають велике значення HTML-заголовкам. Використовуючи заголовки в логічному порядку можна створити чітку і змістовну структуру, яка покращує читабельність і зручність користування. Правильно використані заголовки окрім того, що надають важливу інформацію пошуковим системам та звичайним користувачам про організацію веб-сторінок, відіграють значну роль в роботі допоміжних технологій(рис. 2.6). Зчитувачі екранів покладаються на структуру заголовків для зрозумілої подачі контенту користувачам з вадами зору. Розробники веб-сайтів без достатніх знань чи досвіду часто використовують HTML-теги заголовків виключно для стилістичного оформлення тексту сторінки. Проте це може негативно вплинути на пошукову оптимізацію та доступність. Тому необхідно звертати увагу на те, щоб на сторінці заголовки використовувалися тільки за призначенням та в правильному порядку.

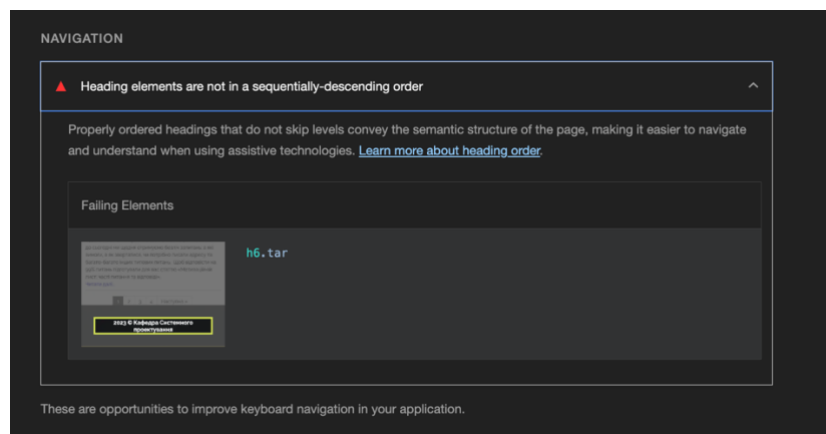


Рисунок 2.6 – Попередження про неправильний порядок заголовків

### 2.3 Огляд найкращих практик

Як можна побачити на рисунку 2.7 веб-сайт не завжди використовує HTTPS протокол в запитах, що є помилкою. Використання протоколу HTTPS є важливим навіть для веб-сайтів, які безпосередньо не збирають і не обробляють інформацію про користувачів. Перш за все HTTPS забезпечує автентичність вмісту веб-сайту. Він не дозволяє зловмисникам впроваджувати

шкідливий код на сторінки веб-сайту, захищаючи його цілісність і підтримуючи його стабільну роботу. Цей протокол надає додатковий рівень захисту від кібератак, таких як “людина посередині”, під час якої зловмисник має змогу слідкувати та змінювати дані, що передаються між сервером веб-сайту та пристроєм користувача. HTTPS шифрує всі дані, якими обмінюється сервер веб-сайту та пристрій користувача, знижуючи ризик шкідливого втручання ззовні. Також факт використання захищеного протоколу підвищує довіру користувачів до веб-сайту, тому що такі сайти позначаються піктограмою з зображенням замка в адресному рядку браузера.

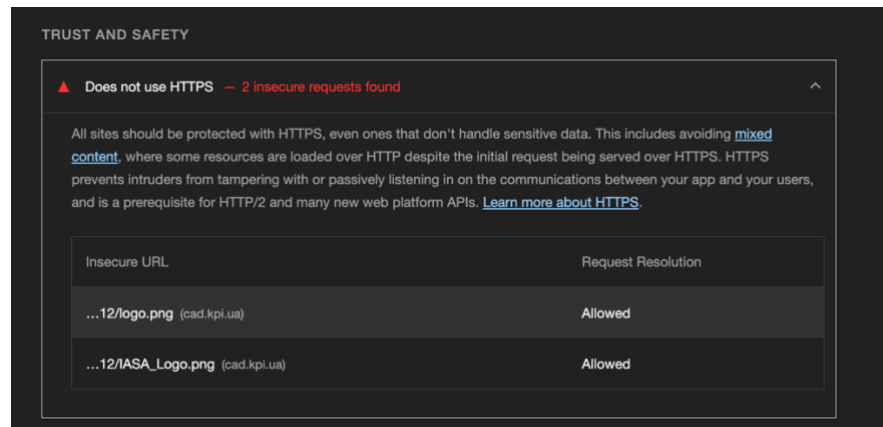


Рисунок 2.7 – Попередження про використання незахищеного HTTP протоколу

Будь-які помилки в консолі веб-сайту впливають на його оцінку пошуковими системами. Помилки свідчать про те, що певні частини веб-сайту можуть працювати неправильно або взагалі не працювати і це може погано вплинути на користувацький досвід(рис. 2.8). Тому потрібно завжди слідкувати за виводом консолі і усувати помилки, що виникають в процесі розробки, навіть якщо явно на роботу сайту вони не впливають.

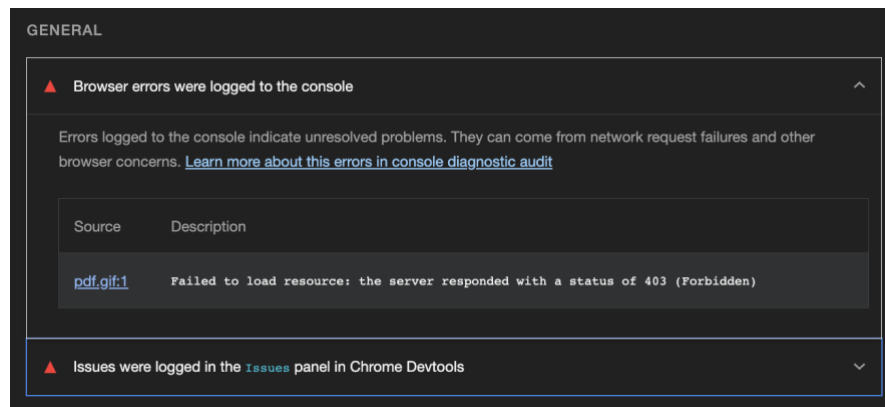


Рисунок 2.8 – Попередження про наявність помилок в консолі браузера

Схожа але менш критична ситуація з повідомленнями у вкладці «Проблеми» браузера. В ній можуть бути повідомлення про неправильну роботу функціоналу веб-сайту, який впливає, наприклад, на персоналізацію контенту або збір статистики. На ці повідомлення також варто звертати увагу але тільки за відсутності більш серйозних проблем.

## 2.4 Огляд проблем SEO-метрик

Найчастіше серед проблем Lighthouse SEO можна зустріти саме відсутність мета-описів(рис. 2.9). Мета-описи – це спеціальні мета теги, в атрибутах яких зберігається короткий опис веб-сторінки. Оптимальною довжиною мета опису вважається інтервал в 150-160 символів. Важливим фактом є те, що добре підібраний мета-опис може підвищити показник частоти кліків(click-through rate) сторінки. Якщо мета-опис на сторінці результатів пошуку зацікавить багато користувачів, відвідуваність сайту неодмінно збільшиться. Проте, якщо мета-опис нецікавий та неінформативний, він може навпаки відштовхнути потенційного відвідувача веб-сайту. Враховуючи те, що пошукові системи можуть автоматично підбирати мета-описи для сторінок веб-сайтів з наявного контенту, до створення мета-описів потрібно підходити з розумом. Крім того, пошукові системи можуть використовувати мета-опис як основу для відображення релевантної інформації в тематичних фрагментах,

панелях знань або інших розширених результатах пошуку, підвищуючи видимість і залучаючи більше органічного трафіку.

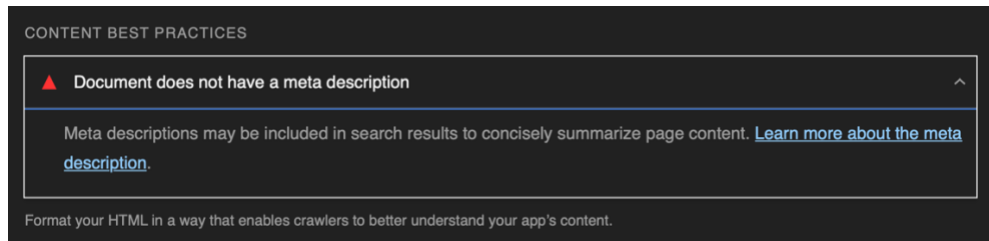


Рисунок 2.9 – Попередження про відсутність мета-опису на сторінці

Наявність посилань, по яким не можуть перейти павуки пошукових систем, призводять до проблем видимості та індексації сторінок веб-сайту. Найпоширенішим прикладом є посилання, що додаються за допомогою JavaScript коду. Павуки пошукових систем при перевірці веб-сайтів часто не можуть виконувати скрипти, що робить такі посилання для них невидимими. У нашому ж випадку на рисунку 2.10 посилання, які не можуть бути проіндексованими, виникли через неправильне використання HTML-тегів.

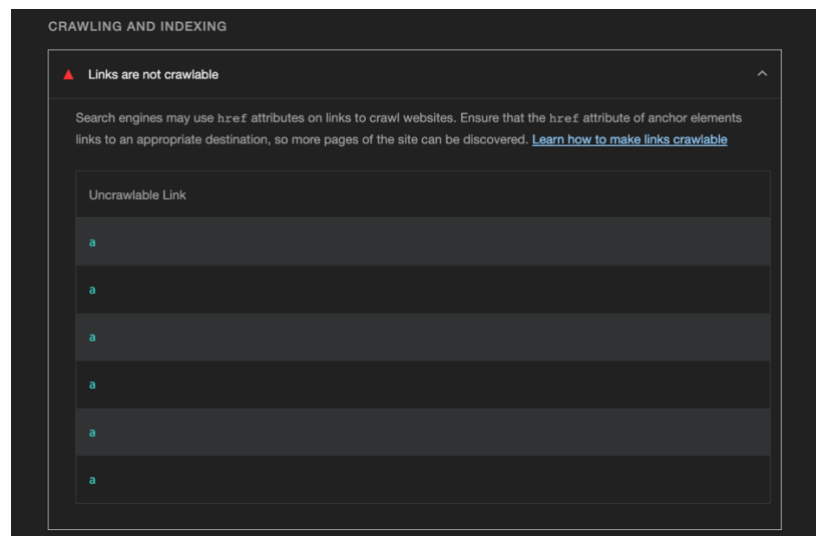


Рисунок 2.10 – Попередження про посилання, які не скануються

Окремим питанням є звісно оптимізація під мобільні пристрої і одним з найважливіших її аспектів є встановлення розміру елементів на сторінках.

Маючи обмежений простір на мобільних пристроях, необхідно впевнитись в тому, що такі основні елементи сторінок як текст, посилання та кнопки мають комфортний для користувача розмір. Приклади таких елементів можна побачити на рисунку 2.11 та рисунку 2.12. Якщо елемент сторінки занадто малий, користувачу буде складно його розгледіти та взаємодіяти. Натомість, якщо елемент занадто великий, він буде неефективно займати місце, на якому міг би знаходитись корисний контент сторінки, та потенційно буде перешкоджати навігації. Тому в цьому випадку потрібно обирати золоту середину, що забезпечить зручність користування та доступність вмісту сторінок

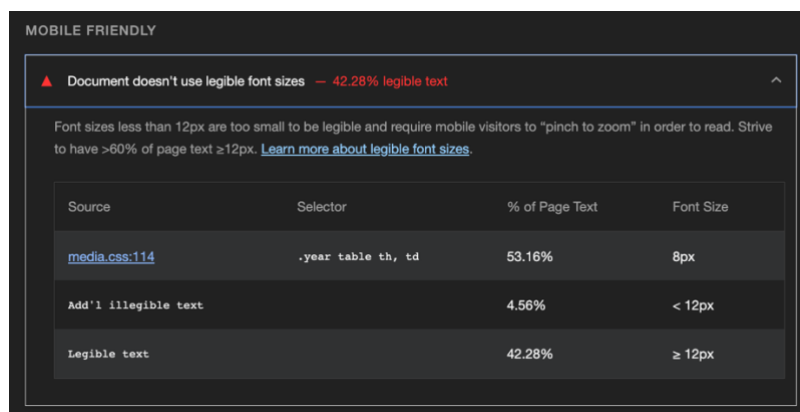


Рисунок 2.11 – Попередження про занадто малий шрифт тексту

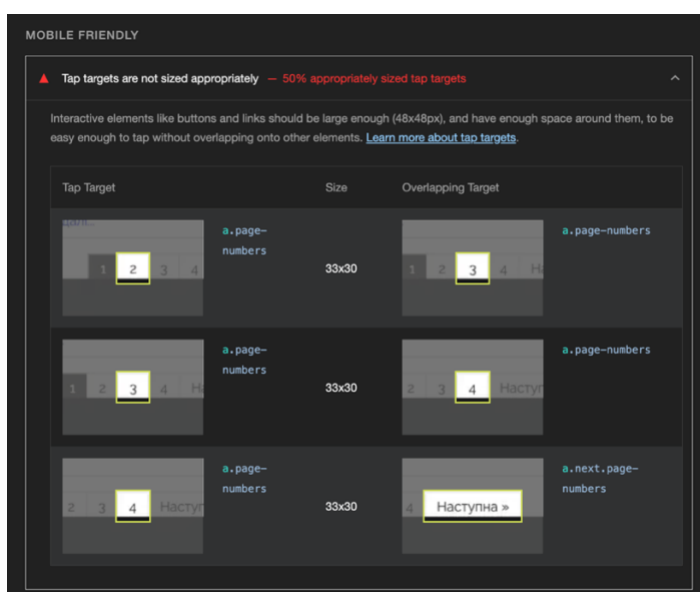


Рисунок 2.12 – Попередження про занадто малі розміри кнопок

## **2.5 Висновки до розділу**

Отже, в цьому розділі був проведений аудит роботи веб-сайту та були розглянуті реальні проблеми пошукової оптимізації веб-сторінок. Наявність цих проблем негативно впливає як і на пошукову оптимізацію, так і на загальний досвід користування сайтом. Розробникам веб-сайтів важливо знати причини їх виникнення, щоб ще на етапі створення сайту не поховати його на далеких сторінках пошукової видачі.

## 3 РЕАЛІЗАЦІЯ ПОШУКОВОЇ ОПТИМІЗАЦІЇ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

В цьому розділі були реалізовані методи пошукової оптимізації на прикладі веб-сайту кафедри Системного проектування. Також була проведена перевірка їх впливу на оцінку веб-сайту пошуковими системами.

### 3.1 Стиснення текстових файлів

Як вже було зрозуміло з минулого розділу, стиснення текстових файлів веб-сайту є дуже ефективним методом підвищення продуктивності. Для реалізації стиснення файлів був обраний найбільш поширений Gzip формат. Gzip широко застосовується в WordPress плагінах для стиснення файлів з декількох причин:

- Висока ефективність в зменшенні розміру файлів;
- Підтримка всіма сучасними браузерами;
- Стиснення відбувається тільки на стороні сервера.

Переважає більшість веб-сайтів досі використовує саме стиснення Gzip, а не більш сучасний Brotli. Незважаючи на те, що Brotli пропонує краще стиснення, не всі браузери на даний момент підтримують його з коробки. Це означає, що не всі користувачі будуть отримувати покращення продуктивності від стиснення і менша кількість розробників засобів для автоматизації стиснення буде зацікавлена в його використанні. Також стиснення алгоритмом Brotli вимагає більшої обчислювальної потужності сервера, що змушуватиме утримувачів веб-сайтів витратити додаткові кошти на вдосконалення серверного обладнання.

На рисунку 3.1 зображена реалізація стиснення файлів за допомогою WordPress плагіна SG-Optimizer. Як можемо бачити, Gzip формат застосовує саме алгоритм Deflate для стиснення

```

# GZIP enabled by SG-Optimizer
<IfModule mod_deflate.c>
  <IfModule mod_filter.c>
    AddOutputFilterByType DEFLATE "application/atom+xml" \
      "application/javascript" \
      "application/json" \
      "application/ld+json" \
      "application/manifest+json" \
      "application/rdf+xml" \
      "application/rss+xml" \
      "application/schema+json" \
      "application/vnd.geo+json" \
      "application/vnd.ms-fontobject" \
      "application/x-font-ttf" \
      "application/x-javascript" \
      "application/x-web-app-manifest+json" \
      "application/xhtml+xml" \
      "application/xml" \
      "font/eot" \
      "font/opentype" \
      "image/bmp" \
      "image/svg+xml" \
      "image/vnd.microsoft.icon" \
      "image/x-icon" \
      "text/cache-manifest" \
      "text/css" \
      "text/html" \
      "text/javascript" \
      "text/plain" \
      "text/vcard" \
      "text/vnd.rim.location.xloc" \
      "text/vtt" \
      "text/x-component" \
      "text/x-cross-domain-policy" \
      "text/xml"
  </IfModule>
</IfModule>
# END GZIP

```

Рисунок 3.1 – Модуль Gzip стиснення текстових файлів

## 3.2 Мінімізація кумулятивного зсуву розмітки

Аудит сторінок веб-сайту показав, що найпоширенішою причиною високого показника кумулятивного зсуву розмітки є саме відсутність явного значення розмірів зображень. Існує декілька різних способів задати розміри зображень. Один з них полягає у визначенні атрибутів ширини та висоти безпосередньо в HTML-тегах зображень. Цей метод корисно застосовувати в ситуаціях, коли відомо, що розмір зображень змінюватись не буде. Було вирішено використовувати ці атрибути для зображень в розмітці постів, яку генерує WordPress(рис 3.2).

```

<!-- wp:image {"align":"left","id":2448} -->
<figure class="wp-block-image alignleft"></figure>
<!-- /wp:image -->

```

Рисунок 3.2 – Використання атрибутів розмірів елемента в редакторі коду WordPress

Інший метод полягає у використанні CSS для встановлення розмірів зображень(рис 3.3). Очевидною перевагою є можливість задати однакові значення розмірів декільком різним елементам та використовувати медіа-запити для забезпечення адаптивності.

```

640 .banner .banner-img img{
641     width: 536px;
642     height: 419px;
643     max-width: 100%;
644 }
645
646 @media screen and (max-width: 1250px) {
647     .banner .banner-text {
648         flex-basis: auto;
649         flex: 1;
650     }
651     .banner .banner-text p {
652         max-width: 100%;
653     }
654     .banner .banner-img{
655         flex-basis: auto;
656     }
657 }
658
659 @media screen and (max-width: 600px) {
660     .banner {
661         padding: 12px;
662         justify-content: center;
663     }
664
665     .banner .banner-img img{
666         width: 388px;
667         height: 303px;
668         max-width: 100%;
669         margin: auto;
670     }
671 }

```

Рисунок 3.3 – Встановлення розмірів зображень за допомогою CSS

Крім того, CSS забезпечує гнучкість розмірів елементів за допомогою відносних одиниць, таких як відсотки, а за допомогою співвідношення сторін(aspect-ratio) можна встановити залежність висоти від ширини чи навпаки. Такий підхід часто застосовується задля забезпечення плавної зміни

розміру елементів сторінки відносно розмірів вікна перегляду браузера чи інших елементів сторінки. На рисунку 3.4 можна побачити застосування відносних одиниць ширини вікна перегляду(vw) та співвідношення сторін(aspect-ratio), що дозволяє зображенню при зміні розміру вікна перегляду розтягуватись чи стискатись, зберігаючи співвідношення сторін.

```
831  .text-page .entry-content .why-table {  
832      width: 74vw;  
833      aspect-ratio : 1.52 / 1;  
834      max-width: 896px;  
835      max-height: 591px;  
836  }
```

Рисунок 3.4 – Застосування відносних одиниць в CSS

### 3.3 Удосконалення доступності

Для покращення доступності веб-сайту кафедри було проведено роботу по усуненню проблем зі структурою веб-сторінок та додавання допоміжних атрибутів для зображень, посилань та iframe.

Нижче на рисунку 3.5 видно, що в розмітці нижнього колонтитула веб-сторінок до оптимізації використовується HTML-тег, який позначає заголовок шостого рівня. При правильній реалізації структури заголовків, наявність заголовка шостого рівня має свідчити про те, що на сторінці точно є заголовки п'ятого, четвертого, третього, другого та першого рівнів. Але в нашому випадку не всі сторінки веб-сайту, що містять в собі нижній колонтитул, мають ці заголовки. Це означає, що використовувати даний тег в тому місці не можна, бо це порушує ієрархію заголовків всіх сторінок веб-сайту.

<code>&lt;footer id="colophon" class="site-footer"&gt;</code>	17 17	<code>&lt;footer id="colophon" class="site-footer"&gt;</code>
<code>&lt;div class="info-row"&gt;</code>	18 18	<code>&lt;div class="info-row"&gt;</code>
<code>&lt;h4 class="tab"&gt;</code>	19 19	<code>&lt;span class="tab"&gt;</code>
<code>&lt;?php echo date("Y"); ?&gt; © Кафедра Системного проектуванн</code>	20 20	<code>&lt;?php echo date("Y"); ?&gt; © Кафедра Системного проектуванн</code>
<code>&lt;/h4&gt;</code>	21 21	<code>&lt;/span&gt;</code>
<code>&lt;/div&gt;</code>	22 22	<code>&lt;/div&gt;</code>
<code>&lt;/footer&gt;&lt;!-- #colophon --&gt;</code>	23 23	<code>&lt;/footer&gt;&lt;!-- #colophon --&gt;</code>

Рисунок 3.5 – Усунення неправильно використаного HTML-заголовка

В загальному семантика не тільки заголовків але й інших HTML-тегів відіграє значну роль в роботі допоміжних технологій. Наприклад, розглянемо рисунок 3.6, де для реалізації кнопки показу бічної панелі меню був використаний тег для посилань. Користувач з вадами зору при взаємодії з цією кнопкою скоріш за все прийняв би її за посилання. До того ж у цієї кнопки відсутній атрибут `aria-label`, який би міг пояснити користувачу що це за елемент сторінки. Очевидно, для виправлення цієї помилки потрібно замінити тег посилання на тег кнопки, вказавши також значення атрибуту `aria-label` та тип кнопки.

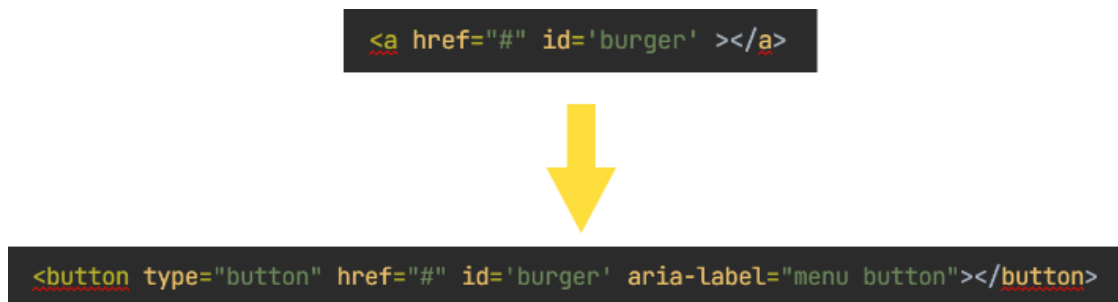


Рисунок 3.6 – Заміна на семантично-підходящий тег та використання `aria-label`

### 3.4 Усунення змішаного контенту

Типовою проблемою сайтів з поганою пошуковою оптимізацією є наявність змішаного контенту. Змішаний контент означає, що веб-сайт використовує одночасно як захищені (HTTPS), так і незахищені (HTTP) запити. Використання на захищених сайтах, незахищених запитів свідчить, що безпеці внутрішнього контенту сайту приділяли мало уваги, тому пошукові

системи знижують їх рейтинг. При цьому пошукові системи не звертають уваги на те, чи збирає сайт користувацькі дані чи ні. Запити, які спричиняють проблему змішаного контенту, це найчастіше запити на отримання зображень та інших файлів. Для пошуку запитів, що використовують протокол HTTP був використаний SEO аудит сервісу Semrush. Він виявив більше 200 випадків змішаного контенту серед 500 перевірених сторінок(рис. 3.7).

Page URL	Resource URL	Resource Type
Сайт кафедри САПР <a href="https://cad.kpi.ua/">https://cad.kpi.ua/</a>	<a href="http://cad.kpi.ua/wp-content/uploads/2021/12/IASA_Logo.png">http://cad.kpi.ua/wp-content/uploads/2021/12/IASA_Logo.png</a>	img
Сайт кафедри САПР <a href="https://cad.kpi.ua/">https://cad.kpi.ua/</a>	<a href="http://10.12.46.38/wp-content/uploads/2021/06/2021-05-04-21.31.37.jpg">http://10.12.46.38/wp-content/uploads/2021/06/2021-05-04-21.31.37.jpg</a>	img
Сайт кафедри САПР <a href="https://cad.kpi.ua/">https://cad.kpi.ua/</a>	<a href="http://cad.kpi.ua/wp-content/uploads/2021/12/logo.png">http://cad.kpi.ua/wp-content/uploads/2021/12/logo.png</a>	img
about-us – Сайт кафедри САПР <a href="https://cad.kpi.ua/about-us/">https://cad.kpi.ua/about-us/</a>	<a href="http://cad.kpi.ua/wp-content/uploads/2021/12/IASA_Logo.png">http://cad.kpi.ua/wp-content/uploads/2021/12/IASA_Logo.png</a>	img
about-us – Сайт кафедри САПР <a href="https://cad.kpi.ua/about-us/">https://cad.kpi.ua/about-us/</a>	<a href="http://cad.kpi.ua/wp-content/uploads/2021/12/logo.png">http://cad.kpi.ua/wp-content/uploads/2021/12/logo.png</a>	img
Контакти – Сайт кафедри САПР <a href="https://cad.kpi.ua/about-us/contacts/">https://cad.kpi.ua/about-us/contacts/</a>	<a href="http://cad.kpi.ua/wp-content/uploads/2021/12/logo.png">http://cad.kpi.ua/wp-content/uploads/2021/12/logo.png</a>	img
Контакти – Сайт кафедри САПР <a href="https://cad.kpi.ua/about-us/contacts/">https://cad.kpi.ua/about-us/contacts/</a>	<a href="http://cad.kpi.ua/wp-content/uploads/2021/12/IASA_Logo.png">http://cad.kpi.ua/wp-content/uploads/2021/12/IASA_Logo.png</a>	img
Історія кафедри – Сайт кафедри САПР <a href="https://cad.kpi.ua/about-us/history/">https://cad.kpi.ua/about-us/history/</a>	<a href="http://cad.kpi.ua/wp-content/uploads/2021/11/image003.jpeg">http://cad.kpi.ua/wp-content/uploads/2021/11/image003.jpeg</a>	img
Історія кафедри – Сайт кафедри САПР <a href="https://cad.kpi.ua/about-us/history/">https://cad.kpi.ua/about-us/history/</a>	<a href="http://cad.kpi.ua/wp-content/uploads/2021/11/image002.jpeg">http://cad.kpi.ua/wp-content/uploads/2021/11/image002.jpeg</a>	img
Історія кафедри – Сайт кафедри САПР <a href="https://cad.kpi.ua/about-us/history/">https://cad.kpi.ua/about-us/history/</a>	<a href="http://cad.kpi.ua/wp-content/uploads/2021/11/image005.jpeg">http://cad.kpi.ua/wp-content/uploads/2021/11/image005.jpeg</a>	img
Історія кафедри – Сайт кафедри САПР	<a href="http://cad.kpi.ua/wp-content/uploads/2021/12/IASA_Logo.png">http://cad.kpi.ua/wp-content/uploads/2021/12/IASA_Logo.png</a>	img

Рисунок 3.7 – Список виявлених джерел змішаного контенту

Виправити проблему змішаного контенту зазвичай дуже просто, але бувають винятки. По перше, якщо змішаний контент походить зі сторонніх джерел, які в своєму API не підтримують використання захищеного протоколу, усунути його буде досить важко. По друге, деякі веб-сайти дозволяють користувачам розміщувати власний контент. В цьому випадку у власника веб-сайту обмежені можливості щодо контролю контенту. Поганим рішенням буде заборонити користувачам розміщувати незахищений контент на веб-сайті, адже це призведе до більших втрат. По третє, проблема

змішаного контенту може виникнути при використанні систем управління контентом, які для генерації посилань можуть використовувати тільки HTTP протокол. Жодна з цих проблем не стосується веб-сайту, який оптимізується в даній роботі, тому достатньо буде змінити протокол посилань в шаблонах сторінок на HTTPS як зображено на рисунку 3.8.

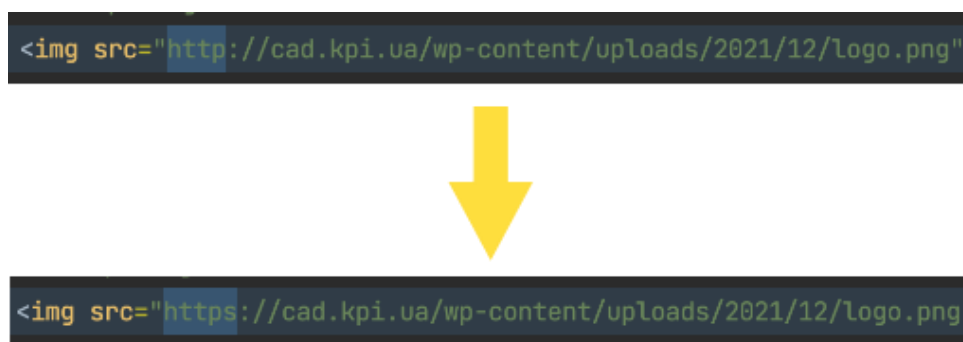


Рисунок 3.8 – Усування змішаного контенту

### 3.5 Створення мета-описів

Найефективнішим методом створення мета-описів для WordPress веб-сайту є використання плагінів. Серед них найпопулярнішим є плагін Yoast SEO. Він дозволяє за допомогою зручного графічного інтерфейсу(рис. 3.9) створювати мета-описи для окремих сторінок, категорій та постів. Під час написання тексту мета-опису цей плагін перевіряє його довжину та надає рекомендації щодо розміру тексту.

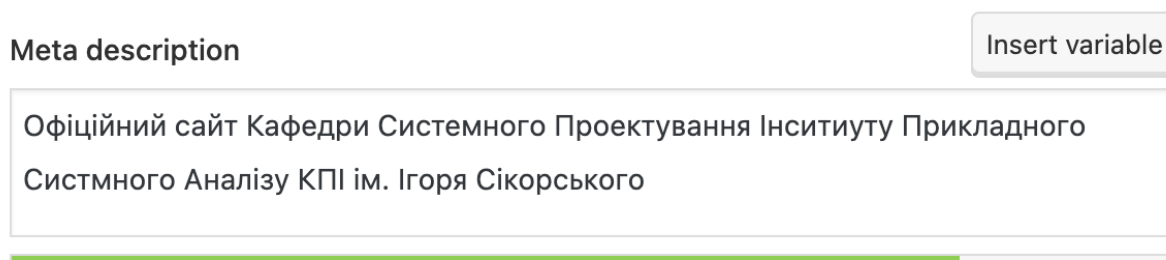


Рисунок 3.9 – Графічний інтерфейс для створення мета-описів

Також Yoast надає можливості по автоматизації створення мета-описів.

Для цього в Yoast існують змінні, які дозволяють автоматично додавати в мета-описи таку інформацію як заголовок сторінки чи посту, дату публікації, назву категорії, тощо.

### **3.6 Усунення посилань, що не індексуються**

Неможливість індексації посилань на сторінках веб-сайтів вважається великою проблемою для пошукових систем. Через неправильні посилання потенційно корисний контент може бути недоступним для користувачів. Переваги коректної роботи стосуються не тільки внутрішніх посилань, але й зовнішніх, адже пошукові системи заохочують поширення релевантного стороннього контенту. Проте розміщувати сторонні посилання на веб-сайті досить ризиковано. Випадки, коли утримувачі веб-сайтів, що поширюють між собою контент, надають інформацію про зміни в своєму ресурсі трапляються не так часто. Через це важко передбачити чи будуть сторонні посилання працювати в майбутньому. Тому дуже важливо перевіряти роботу всіх сторонніх посилань, і за потреби замінювати їх на працюючі чи подібні за контентом. На щастя, сучасні інструменти пошукової оптимізації дозволяють легко знаходити посилання, які не індексуються.

Ще одною причиною появи на веб-сторінках посилань, які не індексуються, може бути неправильне використання тегів. Наприклад, головній навігації сайту кафедри елементи, що позначають категорії зі спадним меню, використовують теги посилань, але вони не мають обов'язкового для посилань атрибуту «href» з адресою. Через це вони позначаються пошуковими системами як ті, що не індексуються. Навігаційне меню на цьому сайті генерується спеціальними засобами WordPress, тому просто змінити тег в розмітці верхнього колонтитула не вийде. У цієї проблеми є два потенційних рішення: модифікувати PHP код, який використовує WordPress для створення меню, або ж створити JavaScript функцію, яка буде змінювати DOM дерево сторінки. Можна одразу сказати,

що другий варіант має значно менше шансів на успіх, бо, як вже було згадано, павуки пошукових систем мають труднощі з виконанням скриптів, а отже при оцінці пошуковими системами цієї сторінки не буде враховано факт заміни тегів. Натомість WordPress заздалегідь генерує розмітку на сервері, а отже пошукові системи оцінюють кінцевий варіант сторінки.

На рисунку 3.10 зображена реалізація класу Custom\_Walker\_Nav\_Menu, який буде генерувати нову розмітку навігаційного меню.

```
class Custom_Walker_Nav_Menu extends Walker_Nav_Menu {
    public function start_el(&$output, $item, $depth = 0, $args = array(), $id = 0) {
        $indent = ($depth > 0 ? str_repeat("\t", $depth) : '');
        if ($args->walker->has_children && $depth === 0) {
            $output .= $indent . '<li';
            $output .= $this->get_attributes($item->classes);
            $output .= '<span>';
        } else {
            $output .= $indent . '<li';
            $output .= $this->get_attributes($item->classes);
            $output .= '<a';
            $output .= ' href="' . esc_attr($item->url) . '"';
            $output .= '>';
        }
        $output .= apply_filters('the_title', $item->title, $item->ID);
        if ($args->walker->has_children && $depth === 0) {
            $output .= '</span>';
        } else {
            $output .= '</a>';
        }
    }
    private function get_attributes($classes) {
        $attributes = '';
        if (!empty($classes)) {
            $attributes .= ' class="' . esc_attr(implode(' ', $classes)) . '"';
        }
        return $attributes;
    }
}
```

Рисунок 3.10 – Клас Custom\_Walker\_Nav\_Menu для створення нової навігації

### 3.7 Створення XML-карти веб-сайту

XML-карти – це спеціальні файли, що зберігають в собі структуру веб-сайту, надаючи павукам пошукових систем цінну інформацію про його організацію. XML-карти сайту включають таку інформацію, як URL-адреси, рівні пріоритету та дати останньої модифікації сторінок, що дозволяє

пошуковим системам ефективно орієнтуватися і розуміти ієрархію веб-сайту та відповідно оновлювати свій індекс. Це гарантує, що всі сторінки будуть відскановані та проіндексовані, покращуючи видимість веб-сайту в результатах пошуку. XML-карти сайту також допомагають швидше індексувати нові та оновлені сторінки, інформуючи пошукові системи про зміни.

Розмітка XML-карт сайту складається з таких елементів [15]:

- <xml> – обов'язковий елемент, що вказує на версію XML-файлу та його кодування;
- <urlset> – обов'язковий елемент, що вказує на стандарт поточного протоколу та обгортає набір записів сторінок;
- <url> – обов'язковий елемент, що обгортає запис сторінки;
- <loc> – обов'язковий елемент, що зберігає URL-адресу сторінки;
- <lastmod> – необов'язковий елемент, що зберігає дату та час останньої модифікації сторінки;
- <priority> – необов'язковий елемент, що вказує на пріоритет даної сторінки відносно інших сторінок веб-сайту для пошукових систем. Можна встановити значення від 0.0 до 1.0;
- <changefreq> – необов'язковий елемент, що вказує на частоту впровадження змін до даної сторінки. Можна встановити значення: постійно(always), погодинно(hourly), щотижнево(weekly), щомісячно(monthly), щорічно(yearly), ніколи(never).

Існує кілька методів створення XML-карт. Одним із підходів є створення XML-карт сайту вручну за допомогою текстового редактора або спеціального програмного забезпечення. Іншим варіантом є використання плагінів для системи керування контентом, які автоматично генерують XML-карти сайту. WordPress в тому числі пропонує плагіни, які динамічно генерують і оновлюють XML-карти сайту, коли додається або змінюється його вміст. Третім варіантом створення XML-карт веб-сайту є використання сторонніх

сервісів. Знаючи адресу домену сайту або адресу стартової сторінки, ці сервіси проходяться по всім сторінкам веб-сайту та додають їх у файл карти. Вибір способу створення залежить від технічних обмежень веб-сайту, але найефективнішим, на мою думку, варіантом є використання плагінів, які тісно інтегровані в систему подачі контенту веб-сайту. Тому для створення XML-карти веб-сайту кафедри був використаний плагін Yoast для WordPress(рис. 3.11), який атоматично створює окремі карти сайту для кожного типу сторінок. Також цей плагін генерує файл robots.txt, в якому можна задавати правила індексації сторінок веб-сайту пошуковими системами, забороняючи чи дозволяючи доступ до сторінок павукам пошукових систем.

## XML Sitemap

Generated by **Yoast SEO**, this is an XML Sitemap, meant for consumption by search engines.

You can find more information about XML sitemaps on [sitemaps.org](https://sitemaps.org).

This XML Sitemap Index file contains 7 sitemaps.

Sitemap	Last Modified
/wordpress/post-sitemap.xml	2023-05-09 16:23 +00:00
/wordpress/page-sitemap.xml	2023-06-10 02:14 +00:00
/wordpress/syllabus-post-sitemap.xml	2023-02-09 12:04 +00:00
/wordpress/category-sitemap.xml	2023-05-09 16:23 +00:00
/wordpress/diploma-sitemap.xml	2023-04-08 03:27 +00:00
/wordpress/syllabus-sitemap.xml	2023-02-09 12:04 +00:00
/wordpress/author-sitemap.xml	2023-06-06 19:24 +00:00

Рисунок 3.11 – XML-карта сайту

Додатково ефективність XML-карт веб-сайту можливо підвищити, надсилаючи їх напряму пошуковим системам. Наприклад, щоб надіслати карту пошуковій системі Google, існує сервіс Google Search Console, який окрім того дозволяє відслідковувати показники роботи веб-сайту в даній пошуковій системі.

### 3.8 Впровадження структурованої розмітки даних Schema.org

Розмітка Schema.org – це стандартизована розмітка, яка використовується для додавання структурованих даних на веб-сторінки, покращуючи їхню видимість у результатах пошукової видачі та дозволяючи пошуковим системам краще розуміти та представляти релевантну інформацію користувачам. Schema.org був розроблений у співпраці між найбільшими пошуковими системами, такими як Google, Bing та Yahoo. Ця розмітка додає контекст і значення елементам веб-сторінки, допомагаючи пошуковим системам точно інтерпретувати і класифікувати інформацію. Schema.org підтримує різноманітні типи контенту, включаючи статті, події, продукти, місцеві компанії, рецепти, огляди та багато іншого. Використовуючи розмітку Schema.org, можна надавати чітку інформацію про різноманітні параметри свого контенту, такі як назва, опис, автор, дата, місцезнаходження, рейтинги та ціни. Розмітка схеми може бути реалізована в трьох різних форматах:

1. JSON-LD(JavaScript Object Notation for Linked Data) – це найпопулярніший формат для розмітки завдяки своїй простоті та легкості реалізації. Він дозволяє вбудовувати розмітку в вигляді скриптів в HTML. Найбільшою перевагою формату є те, що він прописується в невидиму для користувача частину HTML-документа. Цей формат легко зчитується пошуковими системами і не вимагає модифікації існуючої структури HTML. JSON-LD рекомендований Google для реалізації структурованої розмітки даних.
2. Microdata – це специфікація HTML5, яка дозволяє вбудовувати структуровані дані в HTML-елементи за допомогою спеціальних атрибутів. Вона дозволяє додавати розмітку схеми безпосередньо до тегів HTML, які представляють контент. Розмітка Microdata концентрується на наданні якомога детальніших характеристик об'єктів.
3. RDFa(Resource Description Framework in Attributes) – це ще одна специфікація HTML5, яка дозволяє включати структуровані дані в атрибути HTML. Подібно до Microdata, RDFa дозволяє додавати

розмітку безпосередньо до тегів HTML. RDFa забезпечує більш гнучкий спосіб представлення складних зв'язків між даними ніж Microdata.

Структуровану розмітку для сайту можна створити вручну, користуючись шаблонами та валідатором Schema.org, або ж за допомогою SEO плагіну для системи керування контентом. Плагін Yoast SEO має функцію автоматичного створення структурованої розмітки з вибором типів сторінок та постів. Структурована розмітка даних для сайту кафедри, що була створена вручну та перевірена валідатором(рис. 3.12), дозволяє вказати інформацію, яку плагіни не можуть отримати автоматично, що є великою перевагою в випадку потреби в розширених результатах пошукової видачі.

```

<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "EducationalOrganization",
  "name": "Кафедра Системного проектування",
  "foundingDate": "1972",
  "description": "Кафедра СП здійснює підготовку фахівців для науково-дослідної, проектної і організаційно-управлінської діяльності",
  "url": "https://cad.kpi.ua",
  "hasOfferCatalog": {
    "@type": "OfferCatalog",
    "name": "Research Areas",
    "itemListElement": [
      {
        "@type": "ListItem",
        "position": 1,
        "item": {
          "@type": "Thing",
          "name": "Application of modern information technologies in the design of the objects of the artificial human environment"
        }
      },
      {
        "@type": "ListItem",
        "position": 2,
        "item": {
          "@type": "Thing",
          "name": "Informatization of the functioning of organizations and institutions"
        }
      }
    ]
  }
}
</script>

```

Рисунок 3.12 – Структурована розмітка, що створена вручну

Натомість структурована розмітка даних для сайту кафедри, що була згенерована плагіном(рис. 3.13), підтягує з мета-даних веб-сайту велику кількість інформації, яка може бути використана саме пошуковими системами для просування сайту.



Рисунок 3.13 – Структурована розмітка, що створена плагіном

### 3.9 Lighthouse аудит результатів оптимізації

В таблиці 3.1 зібрані результати аудиту основних сторінок за допомогою плагіну Lighthouse після проведення заходів з оптимізації. Оцінки представлені за 100-бальною шкалою, де 0 це найгірший результат, а 100 – найкращий.

Таблиця 3.1 – Результати аудиту Lighthouse сторінок сайту після оптимізації

Сторінка	Продуктивність	Доступність	Найкращі практики	SEO
home/	100	100	100	100
news/	100	100	100	100
about-us/official-information/	100	100	100	100
about-us/history/	99	100	100	100
about-us/teachers/	99	100	100	100
admission/bachelor/	99	100	100	100
admission/magistracy/	100	100	100	99

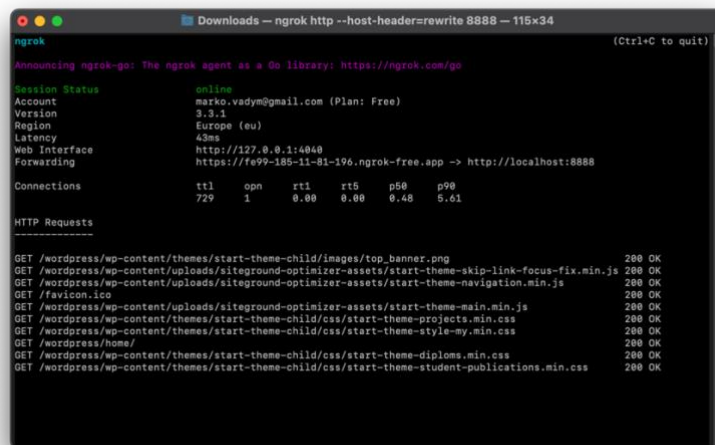
## Продовження таблиці 3.2

admission/second-degree/	100	100	100	100
admission/correspondence-tuition/	100	100	100	100
admission/postgraduate/	100	100	100	100
admission/employment/	100	100	100	100
admission/why-sd/	100	100	100	100
student/osvitni-progamy/	100	100	100	100
student/navchalnij-plan/	100	100	100	100
admission/catalogs/	100	100	100	100
student/syllabuses/	100	100	100	100
student/rozklad-zanjat/	100	100	100	100
student/navchalna-dokumentacija/	99	100	100	100
student/opytuvannia/	100	100	100	100
student-life/polozhennja-pro-imenni-stipendii/	100	100	100	98
scientific-school/	100	100	100	100
science/specializovana-vchena-rada/	100	100	100	100
science/naukov-naprjamki/	100	100	100	100
science/dosjagnennja/	100	100	100	100
science/proekti-shhovikonujutsja/	99	100	100	100
category/konferencii/	100	100	100	100
science/publkacii/	100	100	100	100
science/studentsk-publkac/	100	100	100	100
science/diplomni-proekti/	100	100	100	99
science/temi-doslidzen/	99	100	100	100
science/temi-doslidzen-aspirantiv/	100	100	100	100
zahist-vypusknih-robot/	100	100	100	100
partnership/international-science-cooperation/	100	100	100	100
about-us/contacts/	100	100	92	100

Як можемо бачити, показники пошукової оптимізації згідно оцінки Lighthouse майже максимальні. Це свідчить про те, що практично всі основні поради, які впливають на оцінку, були реалізовані.

### 3.10 Semrush аудит результатів оптимізації

Щоб впевнитися в репрезентативності результатів, було прийнято рішення використати SEO аудит сервісу Semrush. Цей сервіс вважається одним з кращих за функціоналом та кількістю корисної інформації про веб-сайт. Аби розпочати перевірку, спочатку потрібно розмістити оптимізовану версію веб-сайту в публічний доступ. З цим допоміг сервіс ngrok, який дозволяє отримати тимчасовий безкоштовний домен для хостингу сайту(рис. 3.14).



```

ngrok
Announcing ngrok-go: The ngrok agent as a Go library: https://ngrok.com/go

Session Status      online
Account             marko.vadym@gmail.com (Plan: Free)
Version             3.3.1
Region              Europe (eu)
Latency              43ms
Web Interface        http://127.0.0.1:4040
Forwarding           https://fe99-185-11-81-196.ngrok-free.app -> http://localhost:8888

Connections         ttl  opn  rt1  rt5  p50  p99
                   729  1    0.00 0.00 0.48 5.61

HTTP Requests
-----
GET /wordpress/wp-content/themes/start-theme-child/images/top_banner.png 200 OK
GET /wordpress/wp-content/uploads/siteground-optimizer-assets/start-theme-skip-link-focus-fix.min.js 200 OK
GET /wordpress/wp-content/uploads/siteground-optimizer-assets/start-theme-navigation.min.js 200 OK
GET /favicon.ico 200 OK
GET /wordpress/wp-content/uploads/siteground-optimizer-assets/start-theme-main.min.js 200 OK
GET /wordpress/wp-content/themes/start-theme-child/css/start-theme-projects.min.css 200 OK
GET /wordpress/wp-content/themes/start-theme-child/css/start-theme-style-my.min.css 200 OK
GET /wordpress/home/ 200 OK
GET /wordpress/wp-content/themes/start-theme-child/css/start-theme-diploms.min.css 200 OK
GET /wordpress/wp-content/themes/start-theme-child/css/start-theme-student-publications.min.css 200 OK

```

Рисунок 3.14 – Вікно роботи веб-серверу ngrok

Тепер коли Semrush може отримати доступ до сайту, проведемо аудит двох його версій: без оптимізацій та з оптимізаціями. Спочатку розглянемо показники сайту без оптимізацій(рис. 3.15). Можемо бачити, що веб-сайт кафедри має трохи нижчу за середню для такого типу веб-сайтів оцінку здоров'я, а саме 72%. За даними Semrush середня оцінка здоров'я веб-сайтів у сфері освіти 74%. Також сервіс може підраховувати кількість критичних

проблем веб-сайту, що несуть суттєвий негативний вплив на його роботу, окремо попереджати про менш важливі проблеми, які впливають на його оптимізацію, а також повідомляти про можливі покращення. Серед окремих показників варто звернути увагу на відсоток проіндексованих сторінок. Сервісу вдалося просканувати тільки 82% всіх сторінок, що не можна назвати хорошим результатом, адже з цього випливає, що контент близько 18% сторінок не буде включений в результати видачі та не сприятиме підвищенню рейтингу ресурсу в пошукових системах. Наступним показником, на який варто звернути увагу, це відсоток використання захищеного протоколу HTTPS на сторінках веб-сайту. Судячи з значення в 78%, на сайті є явні проблеми зі змішаним контентом. Показник продуктивності роботи веб-сайту теж далеко не найкращий, а отже у веб-сайту є досить великий простір для вдосконалення.

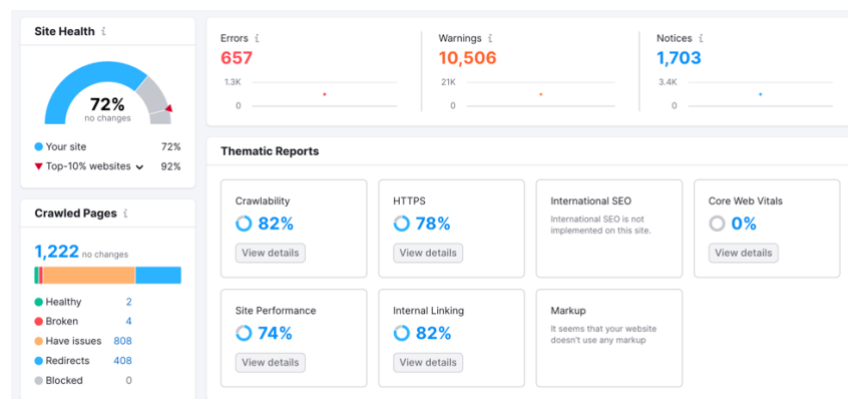


Рисунок 3.15 – Показники Semrush аудиту сайту без оптимізацій

Тепер розглянемо результати аудиту оптимізованої версії веб-сайту (рис. 3.16). В результаті проведених заходів вдалося підняти загальний бал роботи сайту з 72% до 90%, що є досить непоганим результатом. За даними Semrush тільки 10% найкращих веб-сайтів мають оцінку 92% і вище. Крім того вдалося позбутися всіх найбільш суттєвих та більшості менш важливих проблем. Серед менш важливих залишились проблеми малої кількості текстового контенту та відсутності заголовків першого рівня на багатьох сторінках веб-сайту. Показник сканування сторінок виріс на 11%, а отже тепер павуки пошукових систем матимуть доступ до значно більшої кількості контенту веб-

сайту. Проблема наявності змішаного контенту була вирішена, так як сайт тепер повністю використовує тільки захищені запити. Оцінка продуктивності наразі показує ідеальний результат, але вона може не завжди відобразити реальний стан справ, адже як і в випадку плагіну Lighthouse вона залежить від фактичного часу завантаження сторінок веб-сайту, на що можуть вплинути зовнішні фактори, такі як зміни в маршрутизації інтернет-трафіку.

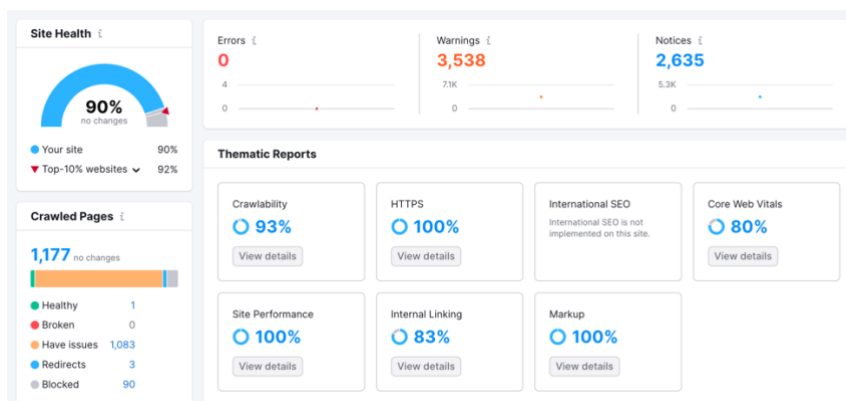


Рисунок 3.16 – Показники Semrush аудиту сайту з оптимізаціями

Додатково можемо поглянути на статистику покриття веб-сайту структурованою розміткою до і після оптимізації(рис. 3.17). Вона вказує на те, що до оптимізації на всіх сторінках сайту, які вдалося відсканувати, структурована розмітка була відсутня. А в оновленій версії всі сторінки веб-сайту мають структуровану розмітку Schema.org і додаткову розмітку Open Graph та Twitter Cards, яка дозволяє налаштовувати вигляд посилань на сторінки веб-сайту в соціальних мережах.

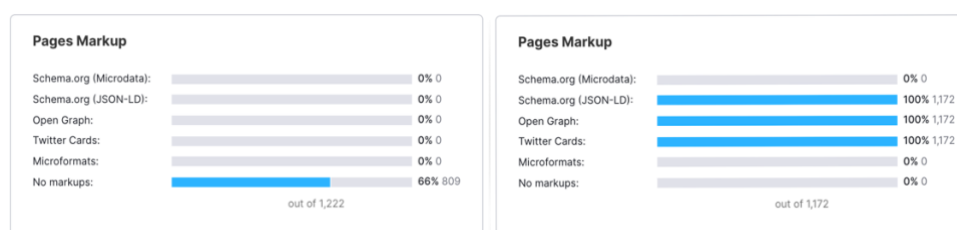


Рисунок 3.17 – Статистика покриття веб сайту структурованою розміткою до оптимізації(ліворуч) та після оптимізації(праворуч)

На рисунку 3.18 зображені діаграми статусних кодів HTTP сторінок сайтів відповідно до та після впровадження методів пошукової оптимізації. Як можемо бачити, до оптимізації третина всіх сторінок повертала статусні коди перенаправлення. Це пов'язано з тим, що посилання на ці сторінки використовували протокол HTTP. Тому сторінки сайту повертали код 301 і автоматично перенаправляли користувача на захищену HTTPS версію сторінки. Таким чином усунення змішаного контенту дозволило позбутися від лишніх перенаправлень.

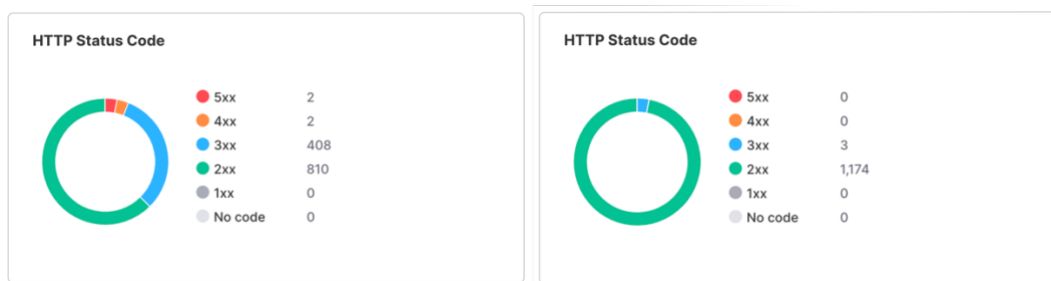


Рисунок 3.18 – Діаграми статусних кодів HTTP сторінок до оптимізації(ліворуч) та після оптимізації(праворуч)

Єдиним параметром, який практично не змінився після оптимізації залишається оцінка внутрішньої структури посилань. Вона враховує розподіл посилань на сторінках веб-сайту та відстань між сторінками. Чим менше кліків потрібно, щоб дістатися від одної частини сайту до іншої, і чим менше непрацюючих внутрішніх посилань, тим вища оцінка. Так як до структури веб-сайту не вносилися ніякі зміни, а непрацюючих внутрішніх посилань було порівняно небагато, оцінка майже не змінилася.

### 3.11 Висновки до розділу

Отже, в результаті проведених заходів з пошукової оптимізації були покращені різноманітні показники роботи веб-сайту, включаючи його

продуктивність, видимість для пошукових систем, доступність і роботу на мобільних пристроях. Реальні результати в вигляді збільшення трафіку чи зворотних посилань будуть помітні тільки з часом на реальному домені, адже пошуковим системам потрібно заново проіндексувати всі сторінки сайту та оновити свій внутрішній рейтинг.

Технології, що використовують веб-сайти можуть дуже відрізнятись, а отже ті, чи інші методи оптимізації можуть бути актуальні по-різному. Наприклад, веб-сайти, що генерують розмітку на сервері, потребують менше зусиль з оптимізації на стороні клієнта для хорошої продуктивності. Якщо код на сервері, працює достатньо швидко, то не потрібно займатися його реінжинірінгом для оптимізації на слабких системах користувачів. Натомість таким веб-сайтам необхідно зосереджуватись в цілому на прискоренні швидкості відповіді сервера, що може включати як і прискорення виконання коду, так і швидкість отримання та передачі даних через мережу. В випадку сайту кафедри використовувалася система керування контентом, що генерувала розмітку веб-сайту на сервері. Тому основні проблеми продуктивності сайту виникали саме через повільну роботу сервера та мережі.

З іншого боку, відмінності в реалізації методів також залежать від використаних веб-сайтом технологій. Наприклад, для редагування розмітки на веб-сайті кафедри необхідно було користуватися в основному інструментами системи керування контентом, а не середовищем розробки. Це можна пояснити тим, що розмітка сторінок веб-сайту зберігалася не в вигляді окремих HTML-документів, чи в документах-шаблонах, а в вигляді записів у базі даних.

## **4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОДУКТУ**

В даному розділі було проведено оцінювання основних характеристик комплексу програмного забезпечення для покращення показників пошукової оптимізації веб-сайтів на WordPress.

Даний програмний комплекс дозволяє проводити аудит веб-сайту, знаходити проблеми в його оптимізації та вирішувати їх. Буде розглянуто різні варіанти вибору програмних компонент та обрано ті, які найефективніше зможуть вирішити поставлені перед ними задачі.

Функціонально-вартісний аналіз (ФВА) передбачає собою технологію, що дозволяє оцінити реальну вартість продукту або послуги незалежно від організаційної структури компанії. ФВА проводиться з метою виявлення резервів зниження витрат шляхом ефективніших варіантів виробництва, кращого співвідношення між споживчою вартістю виробу та витратами на його виготовлення. Для проведення аналізу використовується економічна, технічна та конструкторська інформація.

Алгоритм функціонально-вартісного аналізу містить визначення послідовності етапів розробки продукту, визначення повних витрат (річних) та кількості робочих годин, визначення джерел витрат та кінцевий розрахунок вартості програмного продукту.

### **4.1 Постановка задачі проектування**

У роботі застосовується метод ФВА для проведення техніко-економічного аналізу розробки системи прогнозу стійкості фінансових показників. Оскільки рішення стосовно проектування та реалізації компонентів, що розробляється, впливають на всю систему, кожна окрема підсистема має її задовольняти. Тому фактичний аналіз являє собою аналіз функцій програмного продукту, призначеного для збору, обробки та проведення аналізу даних по компанії.

Технічні вимоги до програмного продукту є наступні:

- функціонування на персональних комп'ютерах зі стандартним набором компонентів;
- зручність та зрозумілість для користувача;
- швидкість обробки даних та доступ до інформації в реальному часі;
- можливість зручного масштабування та обслуговування;
- мінімальні витрати на впровадження програмного продукту.

## 4.2 Обґрунтування функцій програмного продукту

Головною функцією в даному випадку є покращення показників пошукової оптимізації WordPress веб-сайту. Опираючись на це, можна виділити наступні функції:

$F_1$  – вибір SEO інструменту .

$F_2$  – вибір мови програмування.

$F_3$  – вибір середовища розробки.

Кожна з цих функцій має декілька варіантів реалізації:

1) Функція  $F_1$ :

а) Semrush SEO Toolkit.

б) Moz Pro: SEO Software.

2) Функція  $F_2$ :

а) JavaScript.

б) PHP.

3) Функція  $F_3$ :

а) WebStorm.

б) PhpStorm.

Варіанти реалізації основних функцій наведені у морфологічній карті системи (рис. 4.1).

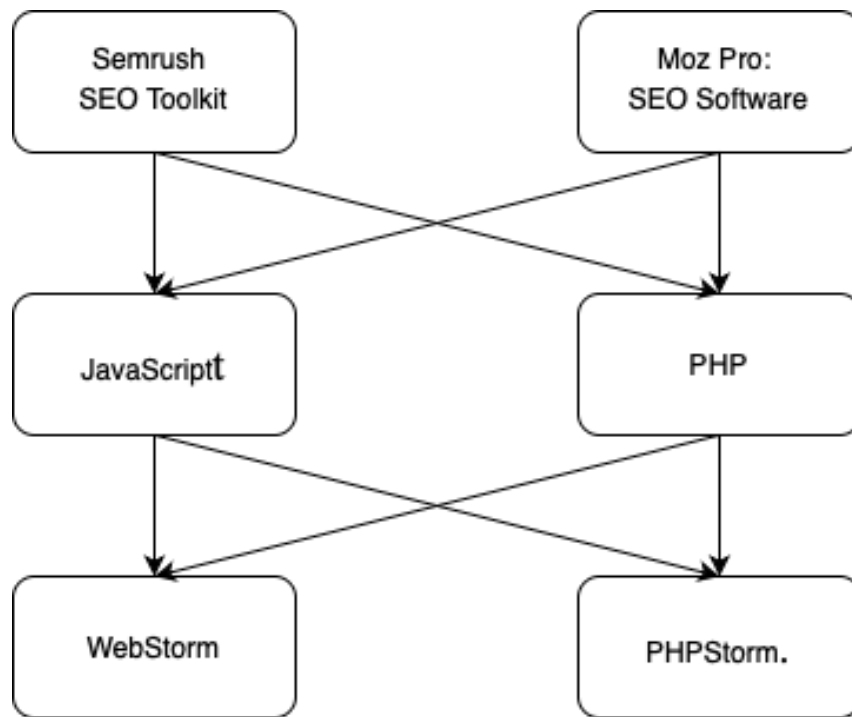


Рисунок 4.1 – Морфологічна карта

Морфологічна карта відображає множину всіх можливих варіантів основних функцій. Позитивно-негативна матриця показана в таблиці 4.1.

На основі аналізу позитивно-негативної матриці робимо висновок, що при розробці програмного продукту деякі варіанти реалізації функцій варто відкинути, тому, що вони не відповідають поставленим перед програмним продуктом задачам. Ці варіанти відзначені у морфологічній карті.

1) Функція  $F_1$ :

Обидва варіанти підходять

2) Функція  $F_2$ :

Незважаючи на те, використовуються дві мови, PHP код відіграє значно більшу роль в пошуковій оптимізації WordPress сайту ніж JavaScript код. Тому вибираємо варіант Б.

3) Функція  $F_3$ :

PHPStorm дозволяє зручно писати код для розширення WordPress, вказуючи на помилки та проблеми і надаючи інструменти автоматизації. Беремо варіант Б.

Таблиця 4.1 – Позитивно-негативна матриця

Функції	Варіанти реалізації	Переваги	Недоліки
$F_1$	А	Пропонує широкий спектр інструментів SEO, включаючи дослідження ключових слів, аналіз конкурентів, аналіз зворотних посилань, аудит сайту, відстеження рейтингу та оптимізацію контенту. Хороша інтеграція з WordPress	Деякі користувачі повідомляють про випадкові розбіжності або неточності в певних показниках
	Б	Відомий своєю точністю даних, надаючи надійну інформацію про рейтинг ключових слів, зворотні посилання, оптимізацію сторінок тощо.	Можливості пошуку ключових слів та налаштування звітів можуть бути відносно обмеженими, менш інтуїтивно зрозумілий інтерфейс
$F_2$	А	Універсальна мова програмування, яку можна використовувати як для фронтенд-, так і для бекенд-розробки	Відсутність строгої типізації може ускладнити підтримку та налагодження коду у великих проектах.
	Б	Має чудові можливості інтеграції з базами даних та веб-серверами	Проблеми з обробкою великого трафіку та масштабуванням, невідповідності в іменах вбудованих функцій
$F_3$	А	Легко інтегрується з популярними веб-технологіями, такими як HTML, CSS, TypeScript, і фреймворками, такими як React, Angular і Vue.js	Слабка підтримка PHP
	Б	Забезпечує надійну підтримку PHP-фреймворків, інтеграцію з базами даних, модульне тестування та контроль версій.	Окрім PHP має слабку підтримку інших мов програмування

Таким чином, будемо розглядати такі варіанти реалізації:

$F_1A - F_2B - F_3B$

$$F_1Б - F_2Б - F_3Б$$

Для оцінювання якості розглянутих функцій обрана система параметрів, описана нижче.

### 4.3 Обґрунтування системи параметрів програмного продукту

На основі даних, розглянутих вище, визначаються основні параметри вибору, які будуть використані для розрахунку коефіцієнта технічного рівня.

Для того, щоб охарактеризувати програмний продукт, будемо використовувати наступні параметри:

- X1 – швидкодія мови програмування;
- X2 – об'єм пам'яті для обчислень та збереження даних;
- X3 – час, який витрачається на виконання коду;
- X4 – потенційний об'єм програмного коду.

Гірші, середні і кращі значення параметрів вибираються на основі вимог замовника й умов, що характеризують експлуатацію програмного продукту, як показано у таблиці 4.2.

Таблиця 4.2 – Основні параметри програмного продукту

Назва параметру	Умовні позначення	Одиниці виміру	Значення параметра		
			гірші	середні	кращі
Швидкодія мови програмування	X1	оп/мс	90	150	250
Об'єм пам'яті	X2	Мб	1500	900	500
Час виконання коду	X3	мс	7000	5000	2000
Потенційний об'єм програмного коду	X4	к-сть рядків коду	5000	2500	1000

За даними таблиці 4.2 будуються графічні характеристики параметрів – рис. 4.2 – рис. 4.5.

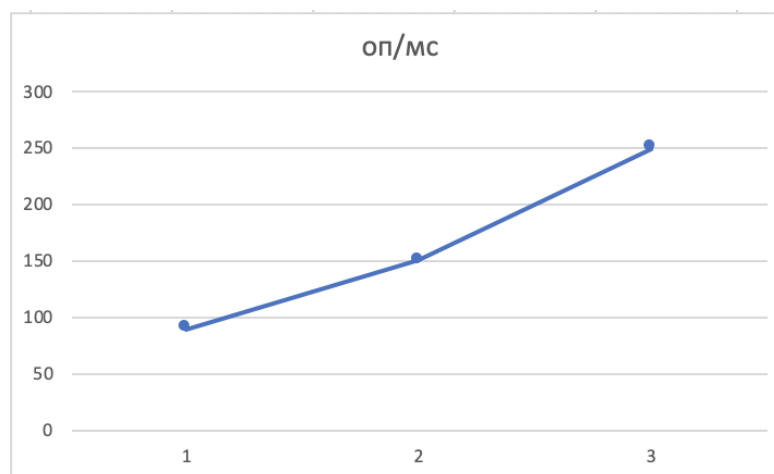


Рисунок 4.2 – X1, швидкодія мови програмування

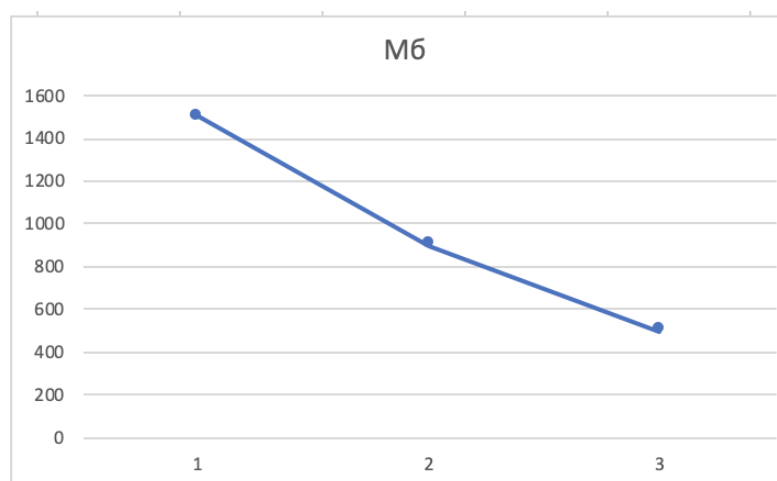


Рисунок 4.3 – X2, об'єм пам'яті

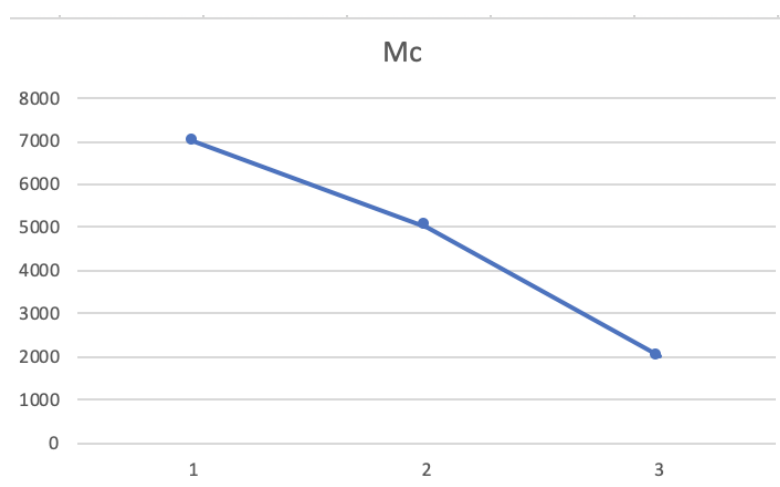


Рисунок 4.4 – X3, час попередньої обробки даних

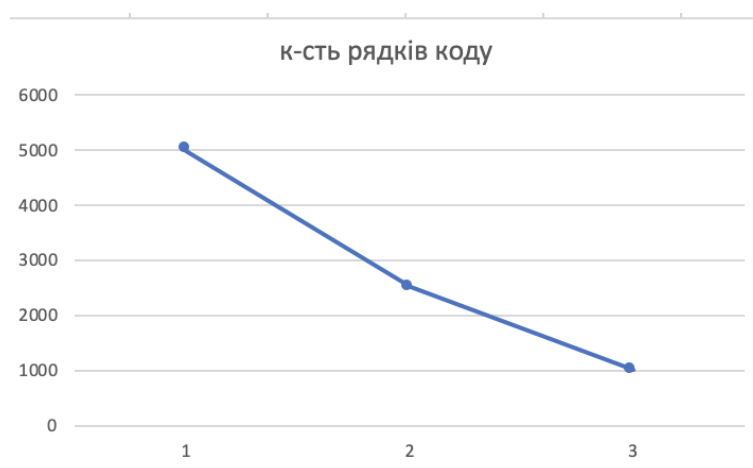


Рисунок 4.5 – X4, потенційний об'єм програмного коду

#### 4.4 Аналіз експертного оцінювання параметрів

Після детального обговорення й аналізу кожний експерт оцінює ступінь важливості кожного параметру для конкретно поставленої цілі – розробка програмного продукту, який дає найбільш точні результати при знаходженні параметрів моделей адаптивного прогнозування і обчислення прогнозних значень.

Значимість кожного параметра визначається методом попарного порівняння. Оцінку проводить експертна комісія із 5 людей. Визначення коефіцієнтів значимості передбачає:

- визначення рівня значимості параметра шляхом присвоєння різних рангів;
- перевірку придатності експертних оцінок для подальшого використання;
- визначення оцінки попарного пріоритету параметрів;
- обробку результатів та визначення коефіцієнту значимості.

Результати експертного ранжування наведені у таблиці 4.3.

Таблиця 4.3 – Результати ранжування параметрів

Параметр	Ранг параметра за оцінкою експерта					Сума рангів	Відхилення $\Delta_i$	$\Delta_i^2$
	1	2	3	4	5			
X1	1	2	1	1	1	6	-6,5	42,25

## Продовження таблиці 4.3

X2	3	3	2	4	2	14	1,5	2,25
X3	4	4	4	3	4	19	6,5	42,25
X4	2	1	3	2	3	11	-1,5	2,25
Разом	10	10	10	10	10	50	0	89

Для перевірки степені достовірності експертних оцінок, визначимо наступні параметри:

а) сума рангів кожного з параметрів і загальна сума рангів:

$$R_i = \sum_{j=1}^N r_{ij} R_{ij} = \frac{Nn(n+1)}{2} = 50 \quad (4.1)$$

де  $N$  – число експертів,  $n$  – кількість параметрів;

б) середня сума рангів:

$$T = \frac{1}{n} R_{ij} = 12,5 \quad (4.2)$$

в) відхилення суми рангів кожного параметра від середньої суми рангів:

$$\Delta_i = R_i - T \quad (4.3)$$

Сума відхилень по всім параметрам дорівнює 0;

г) загальна сума квадратів відхилення:

$$S = \sum_{i=1}^N \Delta_i^2 = 89 \quad (4.4)$$

Порахуємо коефіцієнт узгодженості:

$$W = \frac{12S}{N^2(n^3-n)} = \frac{12 \cdot 89}{5^2(4^3-4)} = 0,712 > W_k (= 0,67) \quad (4.5)$$

Ранжування можна вважати достовірним, тому що знайдений коефіцієнт узгодженості перевищує нормативний, котрий дорівнює 0,67.

Скориставшись результатами ранжирування, проведемо попарне порівняння всіх параметрів і результати занесемо у таблицю 4.4.

Таблиця 4.4 – Попарне порівняння параметрів

Параметри	Експерти					Кінцева оцінка	Числове значення
	1	2	3	4	5		
X1 і X2	<	<	<	<	<	<	0,5
X1 і X3	<	<	<	<	<	<	0,5
X1 і X4	<	>	<	<	<	<	0,5
X2 і X3	<	<	<	>	<	<	0,5
X2 і X4	>	>	<	>	<	>	1,5
X3 і X4	>	>	>	>	>	>	1,5

Числове значення, що визначає ступінь переваги  $i$ -го параметра над  $j$ -тим,  $a_{ij}$  визначається по формулі:

$$\alpha_{ij} = \{1,5 \ x_i > x_j; \ 1 \ x_i = x_j; \ 0,5 \ x_i < x_j\}. \quad (4.6)$$

З отриманих числових оцінок переваги складемо матрицю  $A = \| a_{ij} \|$ .

Для кожного параметра зробимо розрахунок вагомості  $K_{\delta i}$  за наступними формулами:

$$K_{\delta i} = \frac{b_i}{\sum_{i=1}^n b_i} \quad (4.7)$$

$$b_i = \sum_{i=1}^N a_{ij} \quad (4.8)$$

Відносні оцінки розраховуються декілька разів доти, поки наступні значення не будуть незначно відрізнятись від попередніх (менше 2%). На дру-гому і наступних кроках відносні оцінки розраховуються за наступними формулами:

$$K'_{\text{вi}} = \frac{b'_i}{\sum_{i=1}^n b_i} \quad (4.9)$$

$$b'_i = \sum_{i=1}^N a_{ij} b_j \quad (4.10)$$

Як видно з таблиці 4.5, різниця значень коефіцієнтів вагомості не перевищує 2%, тому більшої кількості ітерацій не потрібно.

Таблиця 4.5 – Розрахунок вагомості параметрів

Параметри	Параметри				Перша ітер.		Друга ітер.		Третя ітер.		Четверта ітер.	
	X1	X2	X3	X4	$b_i$	$K_{\text{вi}}$	$b'_i$	$K'_{\text{вi}}$	$b'_i$	$K'_{\text{вi}}$	$b'_i$	$K'_{\text{вi}}$
X1	1	0,5	0,5	0,5	2,5	0,179	8,25	0,167	28,875	0,160	104,313	0,158
X2	0,5	1	0,5	1,5	3,5	0,25	13,25	0,268	46,125	0,257	171,563	0,260
X3	0,5	0,5	1	1,5	3,5	0,25	13,25	0,268	46,125	0,257	171,563	0,260
X4	0,5	1,5	1,5	1	4,5	0,321	14,75	0,298	58,625	0,326	211,438	0,321
Всього:					14	1	49,5	1	179,75	1	658,877	1

#### 4.5 Аналіз рівня якості варіантів реалізації функцій

Визначаємо рівень якості кожного варіанту виконання основних функцій окремо.

Абсолютні значення параметрів X2 (Об'єм пам'яті), X3 (час попередньої обробки даних) та X4 (потенційний об'єм програмного коду) відповідають

технічним вимогам умов функціонування даного ПП.

Абсолютне значення параметра  $XI$  (швидкість роботи мови програмування) обрано не найгіршим.

Коефіцієнт технічного рівня для кожного варіанта реалізації ПП розраховується так (таблиця 4.6):

$$K_{TP} = \sum_{i=1}^n * K_{B_i} B_i \quad (4.11)$$

де  $n$  – кількість параметрів;

$K_{B_i}$  – коефіцієнт вагомості  $i$ -го параметра;

$B_i$  – оцінка  $i$ -го параметра в балах.

Таблиця 4.6 – Розрахунок показників рівня якості варіантів реалізації основних функцій ПП

Основні функції	Варіант реалізації функції	Абсолютне значення параметра	Бальна оцінка параметра	Коефіцієнт вагомості параметра	Коефіцієнт рівня якості
F1 (X2, X3)	А	700	5	0,257	1,285
	Б	500	7	0,257	1,799
F2(X1)	Б	120	3	0,158	0,474
F3(X4)	Б	1500	8	0,321	2,568

За даними з таблиці 4.6 визначаємо рівень якості кожного з варіантів:

$$K_K = K_{TY}[F_{1k}] + K_{TY}[F_{2k}] + \dots + K_{TY}[F_{zk}],$$

Знаходимо загальний рівень якості кожного варіанту:

$$K_{KI} = 1,285 + 0,474 + 2,568 = 4,327,$$

$$K_{K2} = 1,799 + 0,474 + 2,568 = 4,841.$$

Судячи з результатів обрахунку загальних рівнів якості варіантів, можна зробити висновок, що 2 варіант кращий, бо для нього більше значення має коефіцієнт технічного рівня.

#### 4.6 Економічний аналіз варіантів розробки ПП

Для визначення вартості розробки ПП спочатку проведемо розрахунок трудомісткості.

Всі варіанти включають в себе два окремих завдання:

1. Розробка проекту програмного продукту;
2. Розробка програмної оболонки;

Завдання 1 за ступенем новизни відноситься до групи А, завдання 2 – до групи Б. За складністю алгоритми, які використовуються в завданні 1 належать до групи 1; а в завданні 2 – до групи 3.

Для реалізації завдання 1 використовується довідкова інформація, а завдання 2 використовує інформацію у вигляді даних.

Проведемо розрахунок норм часу на розробку та програмування для кожного з завдань.

Загальна трудомісткість обчислюється як:

$$T_0 = T_P \cdot K_{\Pi} \cdot K_{СК} \cdot K_M \cdot K_{СТ} \cdot K_{СТ.М}, \quad (4.12)$$

де  $T_P$  – трудомісткість розробки ПП;

$K_{\Pi}$  – поправочний коефіцієнт;

$K_{СК}$  – коефіцієнт на складність вхідної інформації;

$K_M$  – коефіцієнт рівня мови програмування;

$K_{СТ}$  – коефіцієнт використання стандартних модулів і прикладних програм;

$K_{СТ.М}$  – коефіцієнт стандартного математичного забезпечення

Для першого завдання, виходячи із норм часу для завдань розрахункового характеру степеню новизни А та групи складності алгоритму 1, трудомісткість дорівнює:  $T_P = 23$  людино-днів. Поправочний коефіцієнт, який враховує вид нормативно-довідкової інформації для першого завдання:  $K_{\Pi} = 1.7$ . Поправочний коефіцієнт, який враховує складність контролю вхідної та вихідної інформації для всіх семи завдань рівний 1:  $K_{СК} = 1$ . Оскільки при розробці першого завдання використовуються стандартні модулі, врахуємо це за допомогою коефіцієнта  $K_{СТ} = 0.8$ . Тоді загальна трудомісткість програмування першого завдання дорівнює:

$$T_1 = 23 \cdot 1,5 \cdot 0,8 = 27,6 \text{ людино-днів.}$$

Проведемо аналогічні розрахунки для подальших завдань.

Для другого завдання (використовується алгоритм третьої групи складності, степінь новизни Б), тобто  $T_P = 28$  людино-днів,  $K_{\Pi} = 0.95$ ,  $K_{СК} = 1$ ,  $K_{СТ} = 0,7$ :

$$T_2 = 28 \cdot 0,95 \cdot 0,7 = 18.6 \text{ людино-днів.}$$

Складаємо трудомісткість відповідних завдань для кожного з обраних варіантів реалізації програми, щоб отримати їх трудомісткість:

$$T_I = (27,6 + 18,6 + 12,3 + 18,6) \cdot 8 = 616,8 \text{ людино-годин.}$$

$$T_{II} = (27,6 + 18,6 + 16,2 + 18,6) \cdot 8 = 648 \text{ людино-годин.}$$

Найбільш високу трудомісткість має варіант II. В розробці беруть участь програміст та маркетолог з окладом 32000 грн та 23500 грн відповідно. Визначимо середню зарплату за годину за формулою:

$$CЧ = \frac{M}{T_m \cdot t} \text{ грн.}, \quad (4.13)$$

де  $M$  – місячний оклад працівників;

– кількість робочих днів тиждень;

– кількість робочих годин в день.

$$CЧ = \frac{32000 + 23500}{2 \cdot 21 \cdot 8} = 165,18 \text{ грн.} \quad (4.14)$$

Тоді, розрахуємо заробітну плату за формулою:

$$CЗП = C_ч \cdot T_i \cdot КД, \quad (4.15)$$

де  $C_ч$  – величина погодинної оплати праці програміста;

– трудомісткість відповідного завдання;

$КД$  – норматив, який враховує додаткову заробітну плату.

Зарплата розробників становить:

$$I. C_{ЗП} = 165,18 \cdot 616,8 \cdot 1.2 = 122\,259,6 \text{ грн.}$$

$$II. C_{ЗП} = 165,18 \cdot 648 \cdot 1.2 = 128\,443,9 \text{ грн.}$$

Відрахування на єдиний соціальний внесок становить 22%:

$$I. C_{ВІД} = C_{ЗП} \cdot 0,22 = 122\,259,6 \cdot 0,22 = 26\,897,1 \text{ грн.}$$

$$II. C_{ВІД} = C_{ЗП} \cdot 0,22 = 128\,443,9 \cdot 0,22 = 28\,257,7 \text{ грн.}$$

Тепер визначимо витрати на оплату однієї машино-години. ( $C_M$ )

Так як одна ЕОМ обслуговується одним інженером з окладом 32000 грн.,

з коефіцієнтом зайнятості 0,2 то для однієї машини отримаємо:

$$C_{\Gamma} = 12 \cdot M \cdot K_3 = 12 \cdot 32000 \cdot 0,2 = 76\,800 \text{ грн.}$$

З урахуванням додаткової заробітної плати:

$$C_{3П} = C_{\Gamma} \cdot (1 + K_3) = 76800 \cdot (1 + 0,2) = 92\,160 \text{ грн.}$$

Відрахування на соціальний внесок:

$$C_{ВІД} = C_{3П} \cdot 0,22 = 92160 \cdot 0,22 = 20\,275,2 \text{ грн.}$$

Амортизаційні відрахування розраховуємо при амортизації 25% та вартості ЕОМ – 16000 грн.

$$C_A = K_{TM} \cdot K_A \cdot Ц_{ПР} = 1,4 \cdot 0,25 \cdot 16000 = 5\,600 \text{ грн.,}$$

де  $K_{TM}$  – коефіцієнт, який враховує витрати на транспортування та монтаж приладу у користувача;

$K_A$  – річна норма амортизації;

$Ц_{ПР}$  – договірна ціна приладу.

Витрати на ремонт та профілактику розраховуємо як:

$$C_P = K_{TM} \cdot Ц_{ПР} \cdot K_P = 1,4 \cdot 16000 \cdot 0,08 = 1\,792 \text{ грн.,}$$

де  $K_P$  – відсоток витрат на поточні ремонти.

Ефективний годинний фонд часу ПК за рік розраховуємо за формулою:

$$\begin{aligned} T_{ЕФ} &= (D_K - D_B - D_C - D_P) \cdot t_3 \cdot K_B = \\ &= (365 - 104 - 12 - 18) \cdot 8 \cdot 0,75 = 1386 \text{ години,} \end{aligned}$$

де  $D_K$  – календарна кількість днів у році;

$D_B, D_C$  – відповідно кількість вихідних та святкових днів;

$D_P$  – кількість днів планових ремонтів устаткування;

$t$  – кількість робочих годин в день;

$K_B$  – коефіцієнт використання приладу у часі протягом зміни.

Витрати на оплату електроенергії розраховуємо за формулою:

$$C_{\text{ЕЛ}} = T_{\text{ЕФ}} \cdot N_C \cdot K_3 \cdot C_{\text{ЕН}} = 1386 \cdot 0,2 \cdot 0,4 \cdot 4,79 = 531,12 \text{ грн.},$$

де  $N_C$  – середньо-споживча потужність приладу;

$K_3$  – коефіцієнтом зайнятості приладу;

$C_{\text{ЕН}}$  – тариф за 1 кВт-годин електроенергії.

Накладні витрати розраховуємо за формулою:

$$C_H = C_{\text{ПР}} \cdot 0,67 = 16000 \cdot 0,67 = 10\,720 \text{ грн.}$$

Тоді, річні експлуатаційні витрати будуть:

$$C_{\text{ЕКС}} = C_{\text{ЗП}} + C_{\text{ВІД}} + C_A + C_P + C_{\text{ЕЛ}} + C_H, \quad (4.16)$$

$$C_{\text{ЕКС}} = 92\,160 + 20\,275,2 + 5\,600 + 1\,792 + 531,12 + 10\,720 = 131\,078,32 \text{ грн.}$$

Собівартість однієї машино-години ЕОМ дорівнюватиме:

$$C_{\text{М-Г}} = C_{\text{ЕКС}} / T_{\text{ЕФ}} = 131\,078,32 / 1386 = 94,57 \text{ грн/год.}$$

Оскільки в даному випадку всі роботи, які пов'язані з розробкою програмного продукту ведуться на ЕОМ, витрати на оплату машинного часу, в залежності від обраного варіанта реалізації, складає:

$$C_M = C_{M-\Gamma} \cdot T, \quad (4.17)$$

$$\text{I. } C_M = 94,57 \cdot 616,8 = 58\,330,8 \text{ грн.}$$

$$\text{II. } C_M = 94,57 \cdot 648 = 61\,281,4 \text{ грн.}$$

Накладні витрати складають 67% від заробітної плати:

$$C_H = C_{3\Pi} \cdot 0,67, \quad (4.18)$$

$$\text{I. } C_H = 122\,259,6 \cdot 0,67 = 81\,913,9 \text{ грн.}$$

$$\text{II. } C_H = 128\,443,9 \cdot 0,67 = 86\,057,4 \text{ грн.}$$

Отже, вартість розробки ПП за варіантами становить:

$$C_{\text{ПП}} = C_{3\Pi} + C_{\text{ВІД}} + C_M + C_H, \quad (4.19)$$

$$\text{I. } C_{\text{ПП}} = 122\,259,6 + 26\,897,1 + 58\,330,8 + 81\,913,9 = 289\,401,4 \text{ грн.}$$

$$\text{II. } C_{\text{ПП}} = 128\,443,9 + 28\,257,7 + 61\,281,4 + 86\,057,4 = 304\,040,4 \text{ грн.}$$

#### 4.7 Вибір кращого варіанту ПП техніко-економічного рівня

Розрахуємо коефіцієнт техніко-економічного рівня за формулою:

$$K_{\text{ТЕР}j} = K_{Kj} / C_{\Phi j}, \quad (4.20)$$

$$K_{\text{ТЕР}} = 4,327 / 289\,401,4 = 1,495 \cdot 10^{-5},$$

$$K_{\text{TEP}} = 4,841 / 304\,040,4 = 1,592 \cdot 10^{-5},$$

Як бачимо, найбільш ефективним є варіант реалізації програми з коефіцієнтом техніко-економічного рівня  $K_{\text{TEP}} = 1,592 \cdot 10^{-5}$ .

Після виконання функціонально-вартісного аналізу програмного комплексу що розроблюється, можна зробити висновок, що з альтернатив, що залишились після першого відбору двох варіантів виконання програмного комплексу оптимальним є розглянутий варіант реалізації програмного продукту. У нього виявився показник техніко-економічного рівня якості  $K_{\text{TEP}} = 1,592 \cdot 10^{-5}$ .

Цей варіант реалізації програмного продукту має такі параметри:

- Інструмент SEO – Moz Pro: SEO Software;
- Використання мови програмування PHP;
- Розробка в середовищі PHPStorm.

Використовуючи цей програмний комплекс, можна отримати дуже точну інформацію про стан пошукової оптимізації веб-сайтів та створювати розширення для систем керування контентом, що використовують мову PHP, в зручному середовищі розробки.

#### **4.8 Висновки до розділу**

Отже, в результаті функціонально-вартісного аналізу був підібраний програмний комплекс для проведення пошукової оптимізації на WordPress веб-сайтах. Як показав аналіз, незважаючи на те, що один варіант програмного продукту був дещо дорожчий за інший, його функціональні переваги були варті додаткових витрат.

## ВИСНОВКИ

В роботі були досліджені та впроваджені методи пошукової оптимізації та проаналізовано їхній вплив на роботу веб-сайтів.

Перш за все, було проведено комплексне дослідження різноманітних методів пошукової оптимізації та їх впливу на продуктивність, доступність та зручність веб-сайту. Це включало огляд основних напрямлень пошукової оптимізації та оцінку доцільності застосування їх методів в даній роботі. Далі було проведено повну перевірку стану сторінок веб-сайту кафедри, що дало розуміння, які проблеми в роботі з пошуковими системами зустрічаються найчастіше та що потрібно виправити задля її усунення. Наступним кроком було застосування засобів оптимізації для вдосконалення веб-сайту. Це включало в себе оптимізацію розмітки, покращення швидкості завантаження сторінок, забезпечення доступності та мобільної адаптивності. Дотримуючись найкращих практик, таких як правильне використання тегів заголовків та aria-атрибутів, усунення змішаного контенту та неправильних посилань, створення структурованої розмітки даних та XML-карт, веб-сайт став більш зручним для пошукових систем. Нарешті був проведений аналіз впливу проведених заходів на показники пошукової оптимізації веб-сайту.

Як результат, веб-сайт кафедри був значною мірою оптимізований для роботи з пошуковими системами. Важливою частиною цієї роботи є опис та рішення типових проблем, які виникають на сторінках сайтів. Представлені в ній підходи можна застосовувати як керівництво по покращенню оптимізації веб-сайтів.

На жаль, високі показники оптимізації веб-сайту тимчасові, якщо за ними не слідкувати та не підтримувати. Як вже було згадано раніше, пошукові системи постійно еволюціонують, вдосконалюючи пошук та впроваджуючи нові способи взаємодії з веб-простором. Якісна оптимізація веб-сайту це постійний процес, який повинен починатися та закінчуватись з життєвим циклом веб-сайту. Для полегшення цього процесу важливо, щоб не тільки експерти з пошукової оптимізації розумілися в її принципах, але й розробники

веб-сайтів, штатні маркетологи, веб-дизайнери, та й взагалі всі, хто використовує веб-сайти як платформу для надання послуг чи товарів. Власники веб-сайтів повинні розуміти, що виділення додаткових ресурсів на пошукову оптимізацію є хорошою інвестицією в майбутнє платформи.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. The Art Of SEO: Mastering Search Engine Optimization by Eric Enge, Stephan Spencer and Jessie Stricchiola, Third Edition, 2015 – Режим доступу до ресурсу:  
<https://s3.amazonaws.com/appcursosdegraca/apostilas-en/information-technology/it-tools/the-art-of-seo-by-eric-and-jessie.pdf>
2. Search Engine Optimization For Dummies by Peter Kent, 2004 – Режим доступу до ресурсу:  
<https://doc.lagout.org/Others/Search%20Engine%20Optimization%20for%20ODummies%20%282004%29.pdf>
3. How to get to the top of Google by Tim Cameron-Kitchen and Exposure Ninja, 2018 – Режим доступу до ресурсу:  
<https://www.forcesrecruitment.co.uk/uploads/images/adverts/How%20to%20get%20to%20the%20top%20of%20Google%202018%20edition%20-%20FORMATTED%20A5%20printbook2.pdf>
4. How Many Websites Are There in the World? [Електронний ресурс] – Режим доступу до ресурсу:  
<https://sitefy.com/how-many-websites-are-there/#How-Many-Websites-Are-There>
5. Optimize Cumulative Layout Shift [Електронний ресурс] – Режим доступу до ресурсу:  
[https://web.dev/optimize-clr/?utm\\_source=lighthouse&utm\\_medium=devtools#images-without-dimensions](https://web.dev/optimize-clr/?utm_source=lighthouse&utm_medium=devtools#images-without-dimensions)
6. Enable Text Compression [Електронний ресурс] – Режим доступу до ресурсу:  
[https://developer.chrome.com/docs/lighthouse/performance/uses-text-compression/?utm\\_source=lighthouse&utm\\_medium=devtools](https://developer.chrome.com/docs/lighthouse/performance/uses-text-compression/?utm_source=lighthouse&utm_medium=devtools)

7. How to Enable GZIP Compression for Faster Web Pages [Електронний ресурс] – Режим доступу до ресурсу:  
<https://blog.hubspot.com/website/gzip-compression>
8. Links must have discernible text [Електронний ресурс] – Режим доступу до ресурсу:  
<https://dequeuniversity.com/rules/axe/4.6/link-name>
9. How to Add WordPress Meta Description, Meta Title, and Focus Keyphrase [Електронний ресурс] – Режим доступу до ресурсу:  
<https://www.hostinger.com/tutorials/wordpress-meta-description>
10. Frames must have an accessible name [Електронний ресурс] – Режим доступу до ресурсу:  
<https://dequeuniversity.com/rules/axe/4.6/frame-title>
11. Schema.org [Електронний ресурс] – Режим доступу до ресурсу:  
<https://schema.org/>
12. Атака «людина посередині» [Електронний ресурс] – Режим доступу до ресурсу:  
[https://uk.wikipedia.org/wiki/%D0%90%D1%82%D0%B0%D0%BA%D0%B0\\_%C2%AB%D0%BB%D1%8E%D0%B4%D0%B8%D0%BD%D0%B0\\_%D0%BF%D0%BE%D1%81%D0%B5%D1%80%D0%B5%D0%B4%D0%B8%D0%BD%D1%96%C2%BB](https://uk.wikipedia.org/wiki/%D0%90%D1%82%D0%B0%D0%BA%D0%B0_%C2%AB%D0%BB%D1%8E%D0%B4%D0%B8%D0%BD%D0%B0_%D0%BF%D0%BE%D1%81%D0%B5%D1%80%D0%B5%D0%B4%D0%B8%D0%BD%D1%96%C2%BB)
13. The Difference Between Brotli and Gzip Compression Algorithms To Speed Up Your Site [Електронний ресурс] – Режим доступу до ресурсу:  
<https://eu.siteground.com/blog/brotli-vs-gzip-compression/>
14. Accessible Tap Targets [Електронний ресурс] – Режим доступу до ресурсу:  
<https://web.dev/accessible-tap-targets/>
15. Sitemaps XML format [Електронний ресурс] – Режим доступу до ресурсу:  
<https://www.sitemaps.org/protocol.html>

16. Мікророзмітка сайту [Електронний ресурс] – Режим доступу до ресурсу:  
<https://holostenko.ua/uk/seo/mikrorazmetka-schema-org>
17. The ultimate guide to robots.txt [Електронний ресурс] – Режим доступу до ресурсу:  
<https://yoast.com/ultimate-guide-robots-txt/>
18. HTTP response status codes [Електронний ресурс] – Режим доступу до ресурсу:  
[https://developer.mozilla.org/en-US/docs/Web/HTTP/Status#redirection\\_messages](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status#redirection_messages)
19. Оцінка продуктивності Lighthouse [Електронний ресурс] – Режим доступу до ресурсу:  
<https://developer.chrome.com/en/docs/lighthouse/performance/performance-scoring/>
20. On-Page SEO: What It Is and How to Do It [Електронний ресурс] – Режим доступу до ресурсу:  
<https://www.semrush.com/blog/on-page-seo/>