УДК 004.738.1

Шубенкова Ірина Анатоліївна
кандидат фізико-математичних наук,
доцент кафедри математичних методів системного аналізу
Інституту прикладного системного аналізу
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Шубенкова Ирина Анатольевна
кандидат физико-математических наук,
доцент кафедры математических методов системного анализа
Института прикладного системного анализа
Национальный технический университет Украины
«Киевский политехнический институт имени Игоря Сикорского»
Shubenkova Iryna
Candidate of Physico-Mathematical Sciences, Associate Professor of
Department of the Mathematical Methods of System Analysis of
Institute for Applied System Analysis
National Technical University of Ukraine
«Igor Sikorsky Kyiv Polytechnic Institute»

Кухарєв Сергій Олександрович
асистент кафедри математичних методів системного аналізу
Інституту прикладного системного аналізу
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Кухарев Сергей Александрович
ассистент кафедры математических методов системного анализа
Института прикладного системного анализа
Национальный технический университет Украины
«Киевский политехнический институт имени Игоря Сикорского»
Kukharyev Sergyi
Assistant of Department of the Mathematical Methods of System Analysis of
Institute for Applied System Analysis
National Technical University of Ukraine
«Igor Sikorsky Kyiv Polytechnic Institute»

Поповська Анна Василівна
магістрант кафедри математичних методів системного аналізу
Інституту прикладного системного аналізу
Національного технічного університету України
«Київський політехнічний інститут імені Ігоря Сікорського»
Поповская Анна Васильевна
магистрант кафедры математических методов системного анализа
Института прикладного системного анализа
Национального технического университета Украины
«Киевский политехнический институт имени Игоря Сикорского»
Popovska Anna
Master Degree Student of Department of the
Mathematical Methods of System Analysis of
Institute for Applied System Analysis of National Technical University of Ukraine
«Igor Sikorsky Kyiv Polytechnic Institute»

МОДУЛЬ ОПТИМІЗАЦІЇ МАКЕТА В СТРУКТУРІ ТЕКСТОВОГО ОНЛАЙН РЕДАКТОРА

МОДУЛЬ ОПТИМИЗАЦИИ МАКЕТА В СТРУКТУРЕ ТЕКСТОВОГО ОНЛАЙН РЕДАКТОРА

THE LAYOUT OPTIMIZATION MODULE IN THE STRUCTURE OF ONLINE TEXT EDITOR

Анотація. *HTML макет, згенерований текстовими онлайн редакторами за результатами форматування тексту користувачами, є зазвичай перевантаженим надлишковими елементами й атрибутами та має недоліки в побудові логічної структури DOM-дерева. Дана стаття присвячена дослідженню аспектів HTML макету, що можуть бути оптимізовані, виробленню рекомендацій до алгоритмів оптимізації та їх програмній реалізації. У статті викладено в скороченому вигляді основні аспекти сучасного стану даної проблеми, що може стати корисним для студентів, аспірантів та фахівців профілю інформаційних технологій в плані розширення знань в напрямку вивчення оптимізаційних алгоритмів та удосконалення роботи у сфері веб-розробок.*

Ключові слова: *HTML макет, оптимізація HTML, SEO, розмітка, верстка, текстовий онлайн редактор.*

Аннотация. *HTML макет, сгенерированный текстовыми онлайн редакторами по результатам форматирования текста пользователями, является обычно перегруженным избыточными элементами и атрибутами и имеет недостатки в построении логической структуры DOM-дерева. Данная статья посвящена исследованию аспектов HTML макета, которые могут быть оптимизированы, выработке рекомендаций к алгоритмам оптимизации и их программной реализации. В статье изложены в сокращенном виде основные аспекты современного положения по данной проблематике, что может стать полезным для студентов, аспирантов и специалистов профиля информационных технологий в плане расширения знаний в направлении изучения оптимизационных алгоритмов и усовершенствования работы в сфере веб-разработок.*

Ключевые слова: *HTML макет, оптимизация HTML, SEO, разметка, верстка, текстовый онлайн редактор.*

Summary. *The HTML layout generated by online text editors upon users' text formatting results is usually overloaded with redundant elements and attributes, and has drawbacks in organizing logical structure of the DOM tree. This article is devoted to the investigating of aspects of the HTML layout that can be optimized, the development of recommendations for optimization algorithms and their program realization. The article outlines the main aspects of the current state of the problem, which may be useful for students, postgraduates and specialists in the field of information technologies in terms of expanding knowledge in the field of studying optimization algorithms and improving performance in the web development area.*

Key words: *HTML layout, HTML optimization, SEO, layout, online text editor.*

HTML макети, створені в результаті роботи текстових онлайн редакторів, відрізняються від макетів веб-ресурсів, розроблених фахівцями. Вони, як правило, не містять помилок, зокрема відсутніх закриваючих тегів та ін., та відповідають (не завжди) рекомендаціям сучасних специфікацій. Однак, HTML код таких макетів є дуже часто невиправдано перебільшеним за рахунок надмірного застосування повторюваних та / або недоцільних елементів, атрибутів, а також не має належної структурованості.

Оптимізація HTML макету в роботі текстових онлайн редакторів має особливе значення, оскільки текст коригується та форматується користувачем «на льоту», постійно змінюючи при цьому HTML макет, за допомогою якого відображається редагований текст. При цьому, як правило, при збільшенні форматування HTML макет тільки збільшується за рахунок нових коректур користувача, а попередньо створені елементи HTML макету не редагуються або не видаляються, навіть коли вони більше не мають вплив на відображення представлення тексту для користувача.

Проведений аналіз макетів, згенерованих такими найсучаснішими та найпрогресивнішими веб-сервісами, як Word Online, Dropbox Paper, Google Docs, Zoho Writer, свідчать про те, що всі макети веб-документів для кожного з редакторів є специфічними, вони мають різні закладені алгоритми побудови та організації DOM-дерева. І, одночасно, всі вони мають різні аспекти, що можуть бути оптимізовані, зокрема:

– подвійне «огорнення» інлайнових елементів тегом з різними CSS властивостями, що призводить до надлишкової кількості елементів, та, в деяких випадках, задвоювання CSS властивостей;
– дублювання CSS властивостей у батьківських та дочірніх елементів;
– наявність суміжних інлайнових елементів з однаковими стилями відображення тексту;
– наявність порожніх інлайнових елементів, що не мають вплив на відображення веб-документа для користувача;
– виокремлення пробілів та, відповідно, кожного з окремих слів чи символів в окремі інлайнові елементи, що мають однакове форматування;
– застосування декількох елементів — блочних та інлайнових для відображення тексту у випадках, коли аналогічне представлення може бути реалізоване шляхом застосування найбільш доцільного, спеціально розробленого для цього елемента, наприклад ненумеровані та нумеровані списки. В деяких випадках, кожний елемент списку з тегом <li>, огортається в окремий контейнер для нього — та ще один блочний елемент ;
– застосування індивідуальних «стильових» тегів для форматування тексту, таких як <b> (для форматування тексту жирним виділенням), <i> (для форматування тексту курсивом) та інші. В сучасних специфікаціях HTML — мови розмітки веб документів та рекомендаціях W3C (World Wide Web Consortium — Консорціуму Всесвітньої мережі), дана технологія не рекомендується до використання, а перевага надається застосуванню CSS властивостей для надання веб-сторінці належного зовнішнього вигляду;
– застосування депрекованих (застарілих) тегів, наприклад, <s> — для форматування тексту закресленим, що також не рекомендується до застосування специфікаціями HTML.

Дане дослідження було б неповним без аналізу наявних інструментів з оптимізації HTML. Розглянутий функціонал таких інструментів, як Tidy, Jevix, HTML Cleaner, HTML-покращувачів (HTML-beautifiers) передбачає, в основному, «технічні» функції стискання, такі як видалення пробільних символів або, навпаки, коректне розставлення відступів для кращої читабельності коду, додавання відсутніх закриваючих тегів, виправлення інших помилок. Такий функціонал звісно має значний оптимізаційний ефект, але не працює з логічною перебудовою та внутрішньою структурою DOM-дерева. Найбільш потужним та багатофункціональним інструментом із досліджених є бібліотека Tidy. Однак, навіть її головним завданням є саме виправлення наявних та очевидних помилок, валідація та забезпечення коректності коду, а не логічний аналіз та оптимізаційна перебудова HTML макету.

На сьогодні вже наявні деякі теоретико-практичні роботи по даній темі. В основному, вони стосуються SEO (Search Engine Optimization), де серед інших рекомендацій авторами вироблені рекомендації до оптимального HTML макету [1]. Так, Керівництво по пошуковій оптимізації від Google [2], однією з рекомендацій відзначає необхідність використання оптимальної структури сайту, при якій контент буде впорядковано і користувачеві буде легко в ньому орієнтуватися. Правило простоти дизайну, відзначене Беном Хеніком [3], включає в себе такі вимоги як, найменша кількість елементів розмітки, найкоротші запити, скорочення кількості нефункціональних елементів та інші. Однією з вартих уваги та реалізованих під час програмування Модуля оптимізації HTML макету (далі — «Модуль») є рекомендація, відзначена Джеремі Кейсом (Jeremy Keith), щодо уникнення застосування депрекованих (застарілих) елементів [4]. Дані роботи, безперечно є важливим підґрунтям даного дослідження, втім вони не є повними та не завершені практичною реалізацією оптимізаційних механізмів засобами програмування.

Проведені дослідження стали підґрунтям для вироблення рекомендацій до алгоритму оптимізації макету, а саме:
– Максимальне скорочення кількості елементів в структурі DOM-дерева;
– Максимальне скорочення кількості атрибутів в структурі кожного з елементів DOM-дерева;
– Збереження форматування та вигляду редагованого й форматованого тексту, HTML макет якого оптимізується, у повному обсязі та у незміненому вигляді;
– Усунення дублювань елементів, що не впливають на відображення редагованого та форматованого тексту для користувача;
– Усунення дублювань атрибутів елементів, що не впливають на відображення редагованого та форматованого тексту для користувача;
– Трансформація елементів зі стилістичними тегами на інлайнові елементи — span з присвоєнням CSS-властивостей у атрибуті «style»;
– Уніфікація найменувань CSS-властивостей у атрибуті «style», що по-різному відображаються у HTML макеті, однак мають однакове відображення для кінцевого користувача текстового онлайн-редактора;
– Скорочення гілок DOM-дерева, тобто передача CSS-властивостей останнім нащадкам (листочкам) DOM-дерева та видалення проміжних елементів, що є інлайновими та/або не впливають на відображення тексту для користувача;
– Збереження CSS-властивостей блочних елементів DOM-дерева у відповідного елемента без передачі їх останньому нащадку у відповідній гілці DOM-дерева;
– Видалення пустих інлайнових елементів;

– Видалення технічних символів, що додаються до HTML макету браузерами, таких як, наприклад нерозривні пробіли — « »;
– Заміна депрекованих (застарілих) елементів на нові аналоги;
– Заміна CSS-властивостей елементів на класи, що значно скорочує розмір HTML макету;
– Об'єднання суміжних елементів, що мають однакові теги та однакові CSS-властивості в атрибуті «style»;
– Можливість налаштування Модуля оптимізації HTML макету у його конфігураційному елементі, в тому числі шляхом обрання:
  – елементів, що мають одинарні теги, та не підлягають видаленню;
  – інших, як правило, блочних елементів, та не підлягають видаленню;
  – CSS-властивостей, що не підлягають передачі нащадкам та повинні залишитись у відповідного батьківського елемента в гілці DOM-дерева;
  – обрання префіксів для поіменування класів елементів.

Вироблені в рамках дослідження рекомендації становитимуть логічну основу для розробки архітектури та здійснення програмної реалізації Модуля оптимізації макету.

Модуль оптимізації макету є прикладною програмою, що виконує конкретну прикладну задачу — коректування та реорганізацію структури DOM-дерева у відповідності до закладеної логіки.

При розробці Модуля оптимізації макету перевагу надано найбільш вдалим та сучасним інструментам. Обрана мова програмування JavaScript є однією з найбільш потужних інструментів веб-програмування, що має багато переваг. Оскільки більшість текстових онлайн редакторів мають архітектуру клієнт-сервер, однією з таких переваг є можливість програмної реалізації закладених алгоритмів як на стороні клієнта так і на стороні сервера.

Використання бібліотеки jQuery, головним фокусом якої є взаємодія JavaScript та HTML, також обумовлене можливістю легкої трансформації програмного коду з front-end до back-end.

Програмна платформа Node js є незалежною від операційної системи та обслуговується з допомогою обраної вище мови програмування. npm (аббр. node package manager) — це стандартний менеджер пакетів, що автоматично встановлюється разом з Node. js. В реєстрі npm на час написання даної статті налічується 650 тисяч пакетів. Для інтеграції Модуля оптимізації макету до текстових онлайн редакторів використовується Gulp. Gulp — це утиліта з потокової збірки проектів, це таск-менеджер для автоматичного виконання часто повторюваних завдань (наприклад, мініфікації, тестування, об'єднання файлів), також написаний на мові програмування JavaScript.

Також одним з найсучасніших інструментів, що забезпечують найвищу зручність організації роботи з програмним додатком є Docker. Docker — це програмне забезпечення для автоматизації розгортання і управління додатками в середовищі віртуалізації на рівні операційної системи. Дозволяє «упакувати» додаток з усім його оточенням і залежностями в контейнер, який може бути перенесений на будь-яку Linux-систему з підтримкою cgroups в ядрі, а також надає середовище з управління контейнерами.

Завдяки застосуванню зазначених технологій Модуль оптимізації макета є надзвичайно маневровим елементом, що може бути інтегровано не тільки до текстових онлайн редакторів, але і після його адаптації до будь-якого веб-ресурсу, що потребуватиме відповідний функціонал.

Модуль розроблено на базі об'єктно орієнтованого програмування у функціональному стилі. І власне сам Модуль являє собою клас, що називається optimizeHTML. Клас optimizeHTML включає в себе:
– об'єкт, який містить поля для налаштування конфігураційних даних;
– набір функцій, що виконуються над вхідними даними — HTML макетом.

Елемент для конфігурації даних — config, має поля що можуть редагуватись адміністратором текстового онлайн-редактора та має наступну структуру:

1. Одинарні теги — self_closing_tags, зокрема такі, як
, <hr> та інші теги, які впливають на стиль оформлення тексту і не містять власного текстового вмісту. В процесі обробки Модулем, вони будуть залишені без змін в обробленому HTML макеті.

2. Теги, що не підлягають об'єднанню — unjoinable_tags, навіть якщо всі їх атрибути є однаковими та/або вони не містять власного текстового вмісту, оскільки дані теги, як правило є блочними елементами DOM-дерева, і відповідно впливають на його зовнішній вигляд. Наприклад, сюди включені теги та <p>, які впроваджують абзаци для текстового наповнення, що міститься в них, або просто перенесення на новий рядок. При цьому, Модуль не передбачає заміни пустого блочного елементу <p> на тег перенесення на новий рядок
, оскільки саме <p> може містити специфічні стилі, які особливим в кожному випадку чином оформлюватимуть відступи та перенесення на наступний рядок.

3. Властивості CSS — unjoinable_styles, наявність яких в атрибуті певного елементу повинна мати результатом залишення такого елементу в незміненому вигляді, навіть якщо суміжні елементи мають однакові з ним властивості CSS. До даного поля, в тестовій версії Модуля перенесено такі CSS-властивості, що мають значення для оформлення та форматування блочних елементів, зокрема margin, padding, width та інші.

4. Префікси для найменування класів — classes_prefix. Для зручності організації та розуміння розробниками вихідного HTML коду, префікси для найменування класів, що призначатимуться елементам можна змінювати будь-яким чином.

Корегуючи конфігураційні параметри, можна досягати різних результатів залежно від поставленої задачі.

Модуль оптимізації включає в себе наступну послідовність 4х основних функцій:

- transferToCSS(). Дана функція приймає вхідні дані як рядок, та за допомогою регулярних виразів змінює стильові теги, такі як <b>, <i>, <u> та інші на інлайнові елементи <span>, з відповідними CSS-властивостями. Таким чином, здійснюється уніфікація CSS-властивостей для забезпечення можливості подальшого аналізу та оптимізації отриманих елементів, зокрема об'єднання різних стильових тегів в один <span> з декількома CSS-властивостями, та подальше об'єднання суміжних інлайнових елементів з однаковими набором таких властивостей. А також, завдяки цій функції Модуль оптимізації макета виконує завдання приведення розмітки у відповідність до рекомендацій W3C. Крім того, функція викликає інші більш технічні функції, результатом роботи яких є видалення надлишкових символів, уніфікація різних позначень однієї CSS властивості, видалення дублювань порожніх елементів, що не мають будь-яких стилів та інші.
- transferStylesFromParents(). Це основна функція Модуля, яка забезпечує збирання всіх CSS-властивостей у гілці DOM-дерева для кожного з кінцевих елементів, та «передає» їх останньому нащадку. Виключення становлять функції «блочних» елементів, які повинні бути збереженні у відповідного батьківського елемента, що буде залишено в структурі DOM-дерева. Здійснюється проходження DOM-дерева по-елементно з виконанням наступних операцій:
  1) для кожного елемента виконується ряд перевірок, а саме: чи це текстова нода, чи парний або непарний тег, чи елемент порожній, чи елемент повинен бути збережений в структурі DOM-дерева незалежно від наявності у нього тих чи інших CSS властивостей, чи має елемент, нащадків, чи має він CSS властивості, що мають бути збережені на відповідному рівні DOM-дерева;
  2) збираються в окремі масиви всі батьківські елементи та всі батьківські CSS-властивості, що повинні бути передані останньому нащадку. На цьому етапі видаляються дублювання.
  3) для визначення місця в структурі результуючого елемента аналогічні дії 1) і 2) провадяться для попереднього елемента. Порівнюються шляхи до поточного і попереднього елемента від кореневого елемента, визначається останній спільний батьківський елемент (якщо такий є) та за результатами зазначених перевірок поточний елемент додається у відповідне місце результуючого макету — або до попереднього елемента, або до останнього спільного батьківського елемента, або до кореневого елемента макета.
- alterStylesByClasses(). Результатом роботи даної функції є заміна стилів елементів — CSS-властивостей на класи. Даний запис значно скорочує

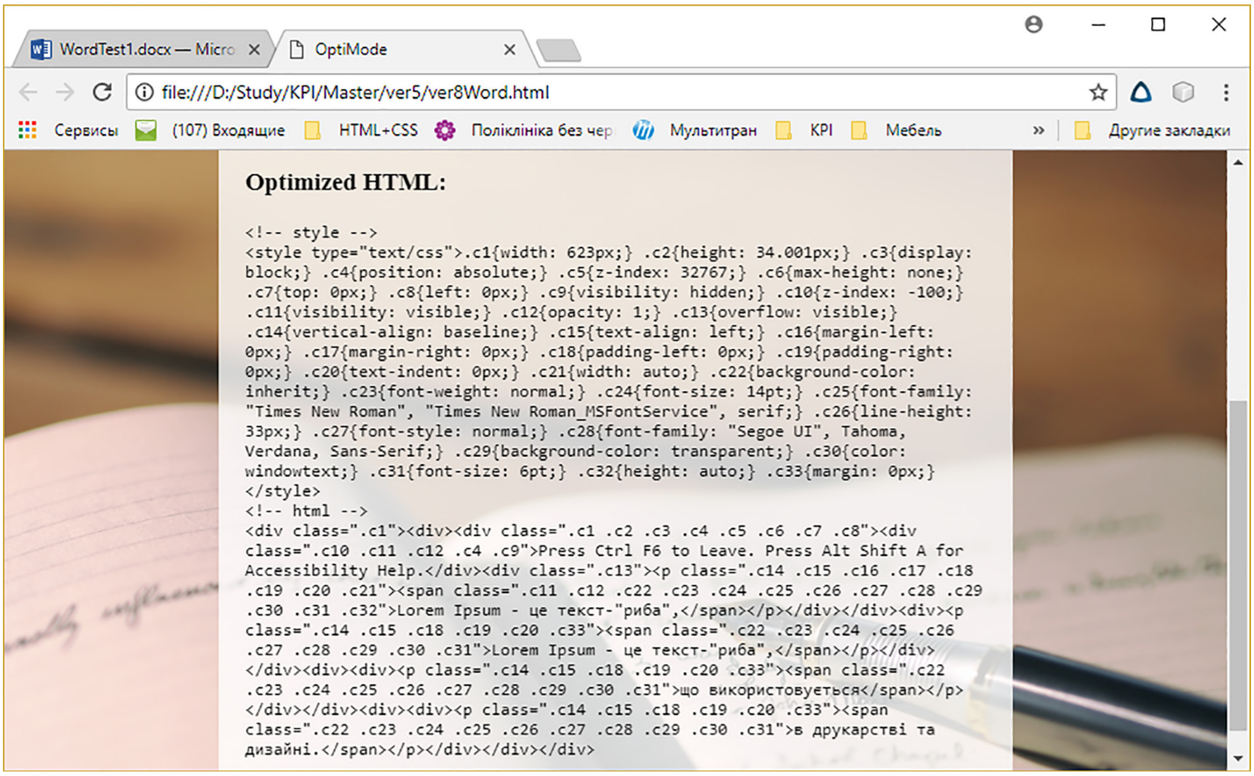

Рис. 1. Скріншот результату роботи Модуля оптимізації макета

Рис. 2. Порівняння параметрів вхідного та оптимізованого макетів Word Online

HTML код та відповідно дозволяє заощадити розмір вихідного макету.

– joinSameElements(). Результатом роботи даної функції є об'єднання суміжних елементів, що мають однакові теги та класи. Функція залишає без змін непарні теги, а також елементи, що визначені як такі, що не підлягають об'єднанню у конфігураційному елементі Модуля. Власне за рахунок цієї функції здійснюється суттєве зменшення кількості елементів DOM-дерева та відповідно зменшується розмір пам'яті, необхідний для його збереження та завантаження.

«Під-операції», викладені вище, також для зручності винесені в окремі функції, по-перше, для більш лаконічної організації розроблюваного програмного коду, по-друге, для заощадження розміру коду, оскільки вони викликаються по декілька разів і для поточного і для попереднього елементів.

HTML код сприймається Модулем як текстовий рядок. В першій функції скрипти виконуються над рядком, в основній функції рядок трансформується (парситься) у набір елементів DOM-дерева. На виході після опрацювання отримуємо HTML код як рядок.

Проведені успішні тестування роботи Модуля оптимізації макету. В якості вхідних даних використовувались реальні згенеровані текстовими онлайн редакторами макети веб-документів, створених користувачами. Для тестування були обрані найбільш потужні веб-ресурси сучасності, такі як Word Online, Dropbox Paper, Google Docs. Тестування підтвердило ефективність та правильність функціонування розробленого додатку. На рисунку 1 представлено Скріншот з результатом роботи Модуля оптимізації макета. Результат має дві частини: стильову з класами, яким корелюють відповідні CSS-властивості та власне вихідний HTML код.

Порівняння параметрів на рисунках 2–4 вхідного та оптимізованого HTML макетів очевидно свідчить про значний оптимізаційний ефект.

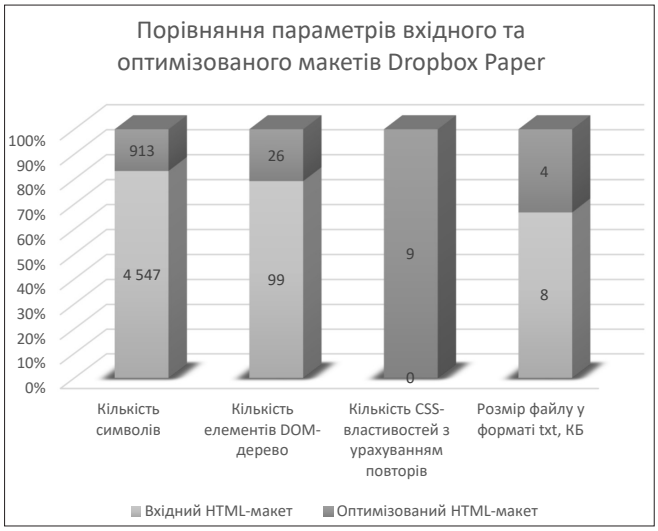

Рис. 3. Порівняння параметрів вхідного та оптимізованого макетів Dropbox Paper

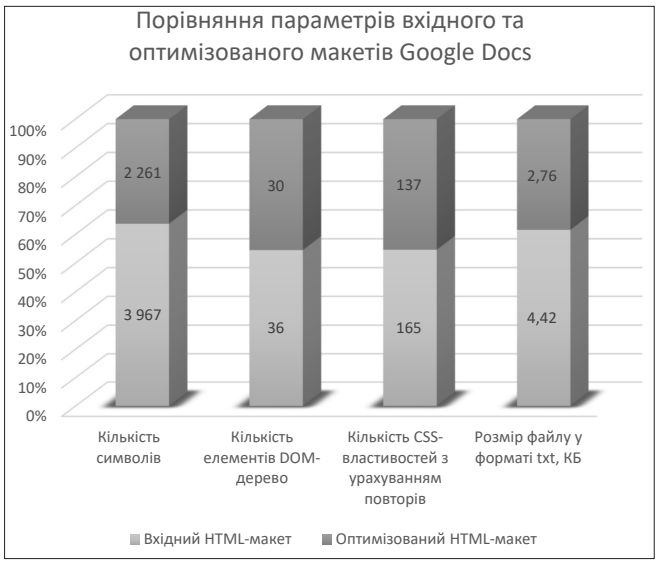

Рис. 4. Порівняння параметрів вхідного та оптимізованого макетів Google Docs

Завдяки програмній реалізації алгоритмів в Модулі оптимізації макету розмір оптимізованих HTML документів зменшуються на розмір, що становить від 37 % до понад 400 % від початкового.

Програмний продукт є зрозумілим у використанні, легко інтегрується до текстових онлайн-редакторів, коректно відпрацьовує подані в якості вхідних даних тести. Модуля оптимізації макету становить міцну технологічну платформу для його подальшого розвитку та адаптації з метою оптимізації макетів будь-яких веб-ресурсів, представлених у Всесвітній мережі.

Література

1. Гроховский Л., Сливинский М., Чекушин А. та Ставский С. SEO: руководство по внутренним факторам. — М.: Центр исследований и образования «ТопЭксперт.РФ», 2011. — 133 с.

2. Google Руководство по поисковой оптимизации для начинающих [Електронний ресурс]. — Режим доступу: https://support.google.com/webmasters/answer/7451184?hl=ru.

3. Хеник Б. HTML и CSS. Путь к совершенству. — СПб.: Питер, 2011. — 336 с.: ил. — (Серия. «Бестселлеры O'Reilly»).

4. Keith J. HTML5 for Web Designers. — New York. — A Book Apart. Second Ed. Feb 17, 2016. — 92 p.