

РОЗРОБКА СИСТЕМИ АНАЛІЗУ ТЕХНІК ШВИДКОГО ЗАВАНТАЖЕННЯ ВЕБ-СТОРІНОК

Соколов Д.В., науковий керівник Міловідов Ю.О., старший викладач.

Актуальність теми. У сучасному веб-просторі швидкість завантаження веб-сторінок є критичним фактором, що безпосередньо впливає на користувацький досвід (UX), рейтинг у пошукових системах (SEO), утримання відвідувачів і рівень конверсії. За статистикою Google, затримка навіть на 1 секунду може знизити конверсію до 20%, а сторінки, які завантажуються довше 3 секунд, втрачають понад половину мобільних користувачів.

Сьогодні веб-розробники мають доступ до численних технік оптимізації продуктивності: lazy-loading, мініфікація, кешування, використання CDN, заміна важких форматів зображень на WebP, відкладене завантаження скриптів тощо. Проте більшість з них впроваджуються інтуїтивно або без чіткого розуміння їх ефективності в конкретному контексті. Це призводить до неочевидних проблем: дублювання запитів, блокуючі ресурси, надмірне завантаження шрифтів, недостатнє кешування — що суттєво знижує швидкодію.

Окрім цього, зростає потреба у вбудованих рішеннях для автоматичного аналізу веб-продуктивності в рамках CI/CD-процесів. Більшість компаній переходять на DevOps-підхід, де продуктивність є таким же важливим параметром, як і функціональність.

Метою роботи є проєктування програмного веб сайту для аналізу структури та завантаження веб-сторінок з метою виявлення, класифікації та оцінки технік оптимізації продуктивності.

Об'єкт дослідження: веб-сторінки, які застосовують різні техніки прискорення завантаження.

Предмет дослідження: методи аналізу ефективності завантаження сторінок та рекомендації щодо їх оптимізації.

Завдання дослідження:

- Провести огляд технік прискорення (minification, lazy loading, caching, prefetching, CDN).
- Розробити модуль сканування DOM-структури, HTTP-запитів, ресурсів.
- Впровадити логіку оцінювання ефективності використаних технік (на основі метрик Core Web Vitals: LCP, FID, CLS).
- Реалізувати інтерфейс для перегляду результатів та отримання рекомендацій.
- Забезпечити інтеграцію з Lighthouse, Puppeteer або аналогічними інструментами.

Практичне застосування. Розроблена система аналізу дозволить веб-ресурсам динамічно оптимізувати продуктивність завантаження, надаючи автоматичні рекомендації залежно від структури сторінки та виявлених вузьких місць. Наприклад, при виявленні затримок у рендерингу сторінки вона може рекомендувати застосування критичного CSS, перенесення JavaScript в кінець документа або впровадження lazy-loading для зображень.

Також система здатна автоматично визначати відсутність кешування статичних ресурсів, надлишкове завантаження шрифтів або дублювання запитів, і формувати оптимізаційні поради з урахуванням метрик Core Web Vitals (LCP, FID, CLS).

Очікувані результати:

- Прототип системи з інтеграцією Lighthouse API.
- Візуалізація технік, що впливають на швидкість.

- Генерація звітів та шаблонів рекомендацій.
- Підвищення ефективності оптимізацій завдяки точковому аналізу.

Опис системи. Розроблена система аналізу технік швидкого завантаження веб-сторінок являє собою програмний модуль, який автоматично сканує сайт, аналізує структуру DOM, мережеві запити, використані ресурси та сучасні практики оптимізації. На основі вбудованої бази знань та інтеграції з інструментами на кшталт Google Lighthouse, Puppeteer або PageSpeed Insights API, система формує звіт із рекомендаціями щодо поліпшення продуктивності.

Система включає:

- Модуль збору даних, який завантажує сторінку у безголовому браузері (headless Chrome) і фіксує ключові метрики: час до першого рендеру, LCP, FID, CLS, обсяг і тип ресурсів, блокуючі скрипти тощо.
- Аналітичне ядро, яке порівнює зібрані дані з best practices (Google Web.dev, W3C, GTMetrix), визначає неефективності та пропонує зміни (наприклад: «Замість PNG використати WebP», «Об'єднати CSS-файли», «Увімкнути кешування»).

Інтерфейс користувача (CLI/Web), який дозволяє:

- вказати адресу сторінки для перевірки;
- переглянути інтерактивний звіт з порадами;
- експортувати рекомендації у форматі PDF, Markdown або JSON.

Механізм формування рекомендацій, який класифікує знайдені проблеми за пріоритетністю (високий/середній/низький вплив на продуктивність), щоб користувач міг впроваджувати зміни поступово, починаючи з критичних.

Можливість інтеграції з CI/CD, що дозволяє вбудувати перевірку продуктивності прямо в процес розгортання — наприклад, автоматичне відхилення pull request'ів, які погіршують Core Web Vitals.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Google Developers. *Web Fundamentals: Performance*. — <https://web.dev/performance/>
2. W3C. *Web Performance Working Group Specifications*. — <https://www.w3.org/webperf/>
3. Lighthouse documentation. *Performance audits*. — <https://github.com/GoogleChrome/lighthouse>
4. GTMetrix. Website Performance Testing Tool. — <https://gtmetrix.com>
5. Документація Docker Compose. <https://docs.docker.com/compose/>

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І
ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**



**ТЕОРЕТИЧНІ ТА ПРИКЛАДНІ
АСПЕКТИ РОЗРОБКИ
КОМП'ЮТЕРНИХ СИСТЕМ**

**ЗБІРНИК НАУКОВИХ ПРАЦЬ ЗА МАТЕРІАЛАМИ
VII ВСЕУКРАЇНСЬКОЇ НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ
СТУДЕНТІВ І АСПІРАНТІВ
*24 квітня 2025 року***

Київ 2025

УДК 004

Відповідальна за випуск: М.І. Лендел

Збірник наукових праць за матеріалами VII Всеукраїнської науково-практичної конференції студентів і аспірантів «ТЕОРЕТИЧНІ ТА ПРИКЛАДНІ АСПЕКТИ РОЗРОБКИ КОМП'ЮТЕРНИХ СИСТЕМ '2025», 24 квітня 2025 року, НУБіП України, Київ. – 452 с. (електронне видання)

Відповідальність за зміст публікацій несуть автори.

Передрук матеріалів, а також використання їх будь-якій формі допускається лише з дозволу авторів