

СПОСОБИ ПІДВИЩЕННЯ ПРОДУКТИВНОСТІ ВЕБСАЙТУ НА PHP

Козодой О.Д., Світенко С.М.

e-mail: olesia.kozodoi@nure.ua, sofia.svitenko@nure.ua

Харківський національний університет радіоелектроніки, каф. ПІ
м. Харків, Україна

This work is devoted to investigating and analyzing methods for enhancing PHP website performance through advanced technological approaches. It encompasses comprehensive strategies for optimization, including caching mechanisms like OPcache, database performance improvement techniques, modern framework utilization (Laravel, Symfony, CodeIgniter), and architectural solutions such as microservices. The research explores performance optimization methodologies, comparing pure PHP with framework-based development, and demonstrates how modern technologies can significantly improve web application efficiency, responsiveness, and development productivity.

Оптимізація веб-продуктивності (Web Performance Optimizarion) – це процес підвищення швидкості та ефективності веб-сайту, що сприяє покращенню користувацького досвіду, зменшенню показників відмов та збільшенню конверсії. В даній доповіді розглянемо комплекс технологічних підходів, спрямованих на підвищення швидкості та продуктивності PHP-додатків, включно з кешуванням, оптимізацією бази даних, вдосконаленням коду, застосуванням сучасних фреймворків та інноваційних архітектурних рішень.

Кешування це один з найбільш ефективних способів прискорення роботи вебсайтів. Він дозволяє зменшити навантаження на сервер та швидше обробляти запити користувачів. Сучасні рішення, такі як Redis та Memcached, доповнюються інтелектуальними системами розподіленого кешування з підтримкою машинного навчання. Це дозволяє не просто зберігати проміжні результати, але й прогнозувати найбільш вірогідні сценарії кешування. Кешування опкодів дає можливість збереження скомпільованого коду PHP, при цьому використання OPcache дозволяє скоротити час виконання скриптів на 50-60% (зі 120 мс до 45 мс) [1].

Ефективна робота з базами даних також значним чином впливає на швидку дію вебсайту, тому оптимізація взаємодії з базою даних є важливим кроком у покращенні продуктивності. Цього можна досягти шляхом використання індексів. Це дозволяє зменшити час пошуку даних у великих таблицях. Наприклад, додавання індексу до стовпця e-mail у таблиці користувачів може прискорити вибірку. Оптимізація SQL-запитів передбачає уникнення зайвих операцій: замість "SELECT * FROM users" рекомендується використовувати "SELECT id, name FROM users".

Використання підготовлених SQL-запитів або ж оптимізації їх структури може відчутно зменшити час виконання запитів [2].

Мікросервісна архітектура пропонує принципово новий підхід до розробки веб-додатків. На відміну від монолітних систем, вона дозволяє розділити додаток на незалежні компоненти, кожен з яких може бути розроблений, розгорнутий і масштабований окремо. Контейнеризація з використанням Docker та оркестрація Kubernetes значно спрощують управління такими розподіленими системами.

Популярним є використання фреймворків Laravel, Symfony та CodeIgniter, які мають вбудовані механізми оптимізації. У сучасному програмуванні часто постає питання вибору між використанням фреймворків або написанням коду на "чистому" PHP. Зокрема, Laravel використовує кешування маршрутів, що зменшує час завантаження, Symfony має оптимізований автозавантажувач класів, а CodeIgniter є мінімалістичним фреймворком, що споживає менше ресурсів [3].

Для того щоб з'ясувати, чи є реальні переваги використання фреймворку, ми проведемо дослідження на прикладі фреймворку Laravel. Основною метою тестування є довести, що фреймворки значно підвищують ефективність розробки порівняно з використанням «чистого» PHP.

Об'єктом тестування є розробка нескладного коду для виведення даних на веб-сторінці з уже існуючої бази даних, в якій зберігається інформація про персонал компанії. Отже, основною задачею є реалізація коду, який зможе звертатись до бази даних та виводити загальні дані працівників. Під час тестування буде визначено час, необхідний для написання одного й того ж функціоналу, а також порівняно кількість рядків коду та час виконання запитів до бази даних. Такі показники дозволять на початковому рівні зробити необхідні порівняння.

Спочатку використаємо «чисте» PHP. Використовуючи його, потрібно вручну налаштувати з'єднання з базою даних, створити запит до бази, обробити дані і вивести їх на сторінці, що займає час. Використання фреймворку Laravel значно спрощує розробку. Для написання простого коду, який виконує необхідні запити до бази даних, було витрачено 3,5 хвилини. Для реалізації цього функціоналу в Laravel було витрачено 2 хвилини. Фреймворк автоматично налаштував підключення до бази даних, обробив запит за допомогою Eloquent ORM і вивів дані через Blade шаблон. Отже, використання фреймворку Laravel скоротило час, необхідний для розробки, на 40%. Результати тестувань наведено до таблиці 1.

Крім того, Laravel має вбудовану маршрутизацію та автоматичне налаштування підключення до бази даних, що не вимагає написання додаткового коду. Вбудовані функції для валідації, аутентифікації та безпеки зменшують необхідність у написанні додаткових бібліотек або

складних рішень. Завдяки цьому, кількість рядків коду на Laravel виявилась приблизно в півтора рази меншою порівняно з «чистим» PHP.

Для вимірювання часу виконання запитів до бази даних було використано просту таблицю працівників з трьома полями: id, name, position, salary. Запит до бази даних на «чистому» PHP та за допомогою Laravel виконувалися за однакових умов (одна і та ж база даних та запит). Час виконання запиту на отримання даних на PHP – 12 мілісекунд, Laravel – 10 мілісекунд. Різниця в часі виконання запитів до бази даних незначна, що показує ефективність фреймворку, але й те, що чистий PHP може бути ефективним для невеликих проєктів (див. табл. 1).

Таблиця 1 – Результати тестування (таблицю виконано самостійно)

Апробовані технології	Кількість рядків коду	Час розробки, (хв)	Час виконання запиту, (мс)
PHP	26	3,5	12
Laravel	16	2	10

Отже, підбиваючи підсумки, зазначимо що підвищення продуктивності вебсайту на PHP є комплексним процесом, що включає кешування, оптимізацію бази даних, покращення коду, використання фреймворків та архітектурних рішень. Проведене дослідження розглядало один з зазначених методів, і за його результатами можемо стверджувати, що використання сучасних технологій та оптимізаційних підходів, є ключем до успіху у веб-розробці.

Список використаних джерел:

1. PHP Benchmarks: OPcache vs OPcache w/ Performance Tweaks : вебсайт. URL: <https://linuxblog.io/php-benchmarks-opcache-performance-tweaks/> (дата звернення: 02.03.2025).
2. Оптимізація SQL-запитів в PHP для покращення продуктивності : вебсайт. URL: <https://webcraftingcode.com/uk/rozrobka-bekenda-z-php/optymizatsiia-sql-zapytiv-v-php-dlia-pokrashchennia-produktyvnosti/> (дата звернення: 03.03.2025).
3. Порівняння показників продуктивності PHP-фреймворків під високим навантаженням : вебсайт. URL: <https://peerdh.com/uk/blogs/programming-insights/comparing-performance-metrics-of-php-frameworks-under-heavy-load-1> (дата звернення: 04.03.2025).
4. Laravel - The PHP Framework For Web Artisans : вебсайт. URL: <https://laravel.com/> (дата звернення: 04.03.2025).
5. Сокорчук, І. П. (2017). Генерація даних зі скриптових мов програмування на UNIX-сумісних платформах [Thesis, ДРУКАРНЯ МАДРИД]. Електронний архів відкритого доступу Харківського національного університету радіоелектроніки. <http://openarchive.nure.ua/handle/document/9483>.