УДК 004.738.5:004.42

А. О. Рикачевський
Я. В. Іванчук

ВИКОРИСТАННЯ NEXT.JS ДЛЯ СТВОРЕННЯ ВЕБ-ЗАСТОСУНКІВ

Вінницький національний технічний університет

Анотація

Запропоновано підхід до використання фреймворку Next.js для створення сучасних вебзастосунків, що передбачає поєднання серверного та клієнтського рендерингу, оптимізацію продуктивності та покращення пошукової індексації. Розглянуто особливості маршрутизації, статичної та динамічної генерації сторінок, а також механізми оптимізації ресурсів. Проведено аналіз впливу архітектурних рішень на швидкодію, масштабованість і зручність підтримки вебзастосунків.

Ключові слова: серверний рендеринг, статична генерація, маршрутизація, оптимізація продуктивності, вебзастосунок.

Abstract

An approach to the use of the Next.js framework for developing modern web applications is proposed, combining server-side and client-side rendering, performance optimization, and improved search engine indexing. The features of routing, static and dynamic page generation, as well as resource optimization mechanisms are examined. An analysis of the impact of architectural decisions on performance, scalability, and maintainability of web applications is conducted.

Keywords: server-side rendering, static generation, routing, performance optimization, web application.

Вступ

Сучасні вебтехнології відіграють важливу роль у розробленні інформаційних систем, систем електронної комерції, освітніх платформ та корпоративних сервісів. Зі зростанням вимог до швидкодії, масштабованості та пошукової оптимізації вебресурсів особливої актуальності набуває застосування сучасних JavaScript-фреймворків і метафреймворків, що забезпечують ефективну інтеграцію серверної та клієнтської логіки. Одним із таких інструментів є фреймворк Next.js, який поєднує можливості серверного рендерингу, статичної генерації сторінок та гібридних підходів до побудови вебзастосунків [1].

Метою роботи є аналіз особливостей використання фреймворку Next.js для створення вебзастосунків з урахуванням архітектурних підходів, механізмів оптимізації продуктивності та забезпечення масштабованості програмних рішень.

Результати дослідження

Для розроблення сучасних вебзастосунків доцільним є використання фреймворку Next.js, оскільки він забезпечує поєднання серверного рендерингу (SSR), статичної генерації сторінок (SSG) та клієнтського рендерингу (CSR). Такий підхід дозволяє оптимізувати час завантаження сторінок, підвищити показники Core Web Vitals та покращити індексацію вебресурсів пошуковими системами [2].

Порівняльний аналіз режимів рендерингу показав, що статична генерація сторінок є найбільш ефективною для контентно орієнтованих ресурсів із нечастим оновленням даних, тоді як серверний рендеринг доцільно застосовувати для динамічних вебзастосунків із персоналізованим вмістом [3]. Використання інкрементальної статичної регенерації (ISR) дає змогу поєднати переваги обох підходів, забезпечуючи автоматичне оновлення сторінок без необхідності повної перебудови проєкту.

У процесі дослідження впливу архітектурних рішень на продуктивність вебзастосунку встановлено, що оптимізація зображень за допомогою вбудованих механізмів Next.js, автоматичне розділення програмного коду (code splitting) та підтримка маршрутизації на основі файлової структури сприяють зменшенню навантаження на клієнтську частину застосунку та скороченню часу першого відображення контенту [4].

Продуктивність вебзастосунку можна подати у вигляді функції ключових параметрів [5]:

$$P = f(R_s, R_c, O_r, C_s), \quad (1)$$

де R_s – частка серверного рендерингу; R_c – частка клієнтського рендерингу; O_r – рівень оптимізації ресурсів; C_s – складність структури застосунку.

Отримані результати свідчать, що раціональне поєднання режимів рендерингу та оптимізаційних механізмів фреймворку Next.js забезпечує підвищення швидкодії, масштабованості та зручності супроводу вебзастосунків порівняно з традиційними підходами до побудови односторінкових застосунків.

Висновки

У результаті дослідження встановлено, що використання фреймворку Next.js для створення вебзастосунків сприяє підвищенню їх продуктивності, покращенню пошукової індексації та забезпеченню гнучкості архітектурних рішень. Поєднання серверного та клієнтського рендерингу, застосування статичної генерації сторінок і механізмів оптимізації ресурсів дозволяють скоротити час завантаження сторінок і підвищити ефективність використання обчислювальних ресурсів.

Запропонований підхід забезпечує масштабованість та зручність супроводу вебзастосунків, що відповідає сучасним вимогам до розроблення інформаційних систем різного рівня складності.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Vercel. Next.js Documentation [Електронний ресурс]. – Режим доступу: https://nextjs.org/docs (дата звернення: 08.03.2026).
2. Alex Banks, Eve Porcello. *Learning React: Modern Patterns for Developing React Apps*. – 2nd ed. – Sebastopol: O'Reilly Media, 2020. – 310 p.
3. Addy Osmani. *Learning JavaScript Design Patterns*. – Sebastopol: O'Reilly Media, 2017. – 254 p.
4. Google. Web Vitals [Електронний ресурс]. – Режим доступу: https://web.dev/vitals/ (дата звернення: 08.03.2026).
5. Mozilla. Server-side rendering (SSR) [Електронний ресурс] // MDN Web Docs. – Режим доступу: https://developer.mozilla.org/ (дата звернення: 08.03.2026).
6. React Team. React Documentation [Електронний ресурс]. – Режим доступу: https://react.dev/ (дата звернення: 08.03.2026).

Рикачевський Андрій Олександрович — студент групи 1КН-22б, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, Вінниця,

Іванчук Ярослав Володимирович — д-р техн. наук, професор, професор кафедри комп'ютерних наук, Вінницький національний технічний університет, м. Вінниця.

Rykachevskyi Andrii O. — Faculty of Intelligent Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: mariabelaya17@gmail.com.

Ivanchuk Yaroslav V. — Dr. Sc. (Eng.), Professor of the Department of Computer Science, Vinnytsia National Technical University, Vinnytsia.