

В. Ю. КОЗУБ

доктор філософії з комп'ютерних наук,
асистент кафедри математики та інформатики
ДЗ «Луганський національний університет імені Тараса Шевченка»
ORCID: 0000-0003-2710-7206

М. А. СЕМЕНОВ

кандидат педагогічних наук, доцент,
завідувач кафедри інформаційних технологій та систем
ДЗ «Луганський національний університет імені Тараса Шевченка»
ORCID: 0000-0003-4989-8109

С. І. ОСАДЧУК

викладач кафедри комп'ютерних систем і технологій
Приватний вищий навчальний заклад «Буковинський університет»
ORCID: 0009-0000-0203-2581

АНАЛІЗ ВПЛИВУ WEBASSEMBLY НА ПРОДУКТИВНІСТЬ СКЛАДНИХ ВЕБЗАСТОСУНКІВ

Мета статті. Мета дослідження полягає у комплексному аналізі впливу технології WebAssembly на продуктивність складних вебзастосунків, визначенні ключових факторів підвищення ефективності обчислень та оцінці перспектив практичного застосування у сучасній веброзробці для оптимізації користувацького досвіду.

Наукова новизна. Наукова новизна роботи полягає в систематизації методів оцінки впливу WebAssembly на продуктивність складних вебзастосунків, розробці комплексного підходу до аналізу ефективності бінарних інструкцій у веб середовищі. Вперше проведено порівняльний аналіз різних архітектурних рішень інтеграції WebAssembly з традиційними JavaScript-рішеннями. Запропоновано методологію оцінки продуктивності для різних типів обчислювальних задач у вебзастосунках з використанням метрик швидкодії та оптимізації пам'яті.

Результати. Результати дослідження демонструють значне підвищення продуктивності складних вебзастосунків при використанні WebAssembly порівняно з традиційними JavaScript-рішеннями. Встановлено, що WebAssembly забезпечує прискорення виконання обчислювально-інтенсивних операцій у 2–10 разів залежно від типу задач. Виявлено оптимальні сценарії застосування технології для математичних обчислень, обробки зображень, криптографічних операцій та мультимедійних застосунків. Проаналізовано вплив різних компіляторів та оптимізації на підсумкову продуктивність. Визначено критичні аспекти інтероперабельності між WebAssembly та JavaScript, що впливають на загальну ефективність системи. Розроблено рекомендації щодо архітектурних рішень для максимізації переваг WebAssembly у складних вебзастосунках.

Висновки. WebAssembly демонструє високий потенціал для оптимізації продуктивності складних вебзастосунків, особливо у сферах з інтенсивними обчисленнями. Технологія забезпечує значне прискорення виконання критично важливих операцій при збереженні безпеки та кросплатформності. Оптиміальне застосування WebAssembly вимагає ретельного аналізу архітектури застосунку та специфіки обчислювальних задач для досягнення максимального ефекту.

Ключові слова: WebAssembly, продуктивність, веб-застосунки, веб-технології, оптимізація, ядра.

V. YU. KOZUB

PhD in Computer Science,
Assistant at the Department of Information Technology and Systems
Luhansk Taras Shevchenko National University
ORCID: 0000-0003-2710-7206

M. A. SEMENOV

PhD in Pedagogy, Associate Professor,
Head of the Department of Information Technologies Educational and Scientific
Luhansk Taras Shevchenko National University
ORCID: 0000-0003-4989-8109

S. I. OSADCHUK

Lecturer at the Department of Computer Systems and Technologies

Private Higher Educational Institution "Bukovinian University"

ORCID: 0009-0000-0203-2581

ANALYSIS OF THE IMPACT OF WEBASSEMBLY ON THE PRODUCTIVITY OF COMPLEX WEB APPLICATIONS

The purpose of the article. The purpose of the study is to comprehensively analyze the impact of WebAssembly technology on the performance of complex web applications, identify key factors for increasing computing efficiency, and assess the prospects for practical application in modern web development to optimize the user experience.

Scientific novelty. The scientific novelty of the work lies in the systematization of methods for assessing the impact of WebAssembly on the performance of complex web applications, the development of a comprehensive approach to analyzing the efficiency of binary instructions in the web environment. For the first time, a comparative analysis of various architectural solutions for integrating WebAssembly with traditional JavaScript solutions has been carried out. A methodology for assessing performance for various types of computational tasks in web applications using performance metrics and memory optimization has been proposed.

Results. The results of the study demonstrate a significant increase in the performance of complex web applications when using WebAssembly compared to traditional JavaScript solutions. It was found that WebAssembly provides an acceleration of computationally intensive operations by 2–10 times, depending on the type of tasks. Optimal scenarios for using the technology for mathematical calculations, image processing, cryptographic operations, and multimedia applications were identified. The impact of different compilers and optimizations on the final performance was analyzed. Critical aspects of interoperability between WebAssembly and JavaScript that affect the overall efficiency of the system were identified. Recommendations for architectural solutions were developed to maximize the benefits of WebAssembly in complex web applications.

Conclusions. WebAssembly shows great potential for optimizing the performance of complex web applications, especially in compute-intensive areas. The technology provides significant acceleration of critical operations while maintaining security and cross-platform compatibility. Optimal use of WebAssembly requires careful analysis of the application architecture and the specifics of the computing tasks to achieve maximum effect.

Key words: WebAssembly, performance, web applications, web technologies, optimization, kernels.

Постановка проблеми

Сучасні вебзастосунки стають дедалі складнішими та вимагають виконання обчислювально-інтенсивних операцій безпосередньо у браузері користувача. Традиційні підходи з використанням JavaScript часто не забезпечують достатньої продуктивності для складних задач обробки даних, математичних обчислень, мультимедійного контенту та ігрових додатків. Обмеження інтерпретованої природи JavaScript створюють бар'єри для розробки високопродуктивних вебзастосунків, що можуть конкурувати з нативними рішеннями. Поява WebAssembly як бінарного формату інструкцій для вебплатформи відкриває нові можливості для значного підвищення продуктивності, проте потребує ґрунтовного дослідження ефективності та оптимальних сценаріїв застосування у контексті складних вебсистем.

Аналіз останніх досліджень і публікацій

Аналіз сучасної літератури демонструє активний розвиток досліджень у сфері WebAssembly та його впливу на продуктивність вебзастосунків. Sipek M., Muharemagic D., Mihaljevic B., Radovan A. [1, с. 1695] досліджували застосування WebAssembly у контексті наступного покоління вебзастосунків з використанням TruffleWasm, що демонструє перспективи інтеграції з сучасними технологічними стеками. Harnes H., Morrison D. [2, с. 1] представили систематизований огляд методів аналізу WebAssembly, що є фундаментальним для розуміння підходів до оцінки продуктивності. Akieva Z. M., Murzin D. A., Eltaev S. I. [3, с. 1] сфокусувалися на практичному застосуванні WebAssembly для покращення продуктивності інтерактивних вебзастосунків. Robati Shirzad M., Lam P. [4, с. 2] розробили інноваційний підхід WasmWalker для аналізу програм WebAssembly на основі шляхових представлень коду. Wang W. [5, с. 1] провів комплексну оцінку готовності вебзастосунків до повноцінного використання WebAssembly. Burkhart N., Liao W., Guzide O. [6, с. 1] надали фундаментальний огляд архітектури WebAssembly. Rao N. S. [7, с. 1973] дослідив революційний вплив WebAssembly на розробку користувацьких інтерфейсів через призму продуктивності та міжмовної інтеграції. Long Y., Su Y., Jiang Z. [8, с. 1] розробили спеціалізований інструмент WACP для профілювання продуктивності взаємодії WebAssembly-Python. Waseem M., Das T., Ahmad A., Liang P., Mikkonen T. [9, с. 1] провели емпіричне дослідження проблем та їх причин у WebAssembly-застосунках. Cabrera-Arteaga J., Fitzgerald N., Monperrus M., Baudry B. [10, с. 1] представили WASM-MUTATE для швидкої та ефективної бінарної диверсифікації WebAssembly.

Формулювання мети дослідження

Метою дослідження є комплексний аналіз впливу WebAssembly на продуктивність складних вебзастосунків через систематизацію методів оцінки ефективності, визначення оптимальних архітектурних рішень та розробку практичних рекомендацій для максимізації переваг технології у реальних проєктах.

Викладення основного матеріалу дослідження

Дослідження продуктивності WebAssembly у складних вебзастосунках базується на комплексному підході до аналізу різних аспектів технології. Sipek M., Muharemagic D., Mihaljevic B., Radovan A. [1, с. 1698] демонструють, що інтеграція WebAssembly з TruffleWasm забезпечує значне підвищення продуктивності для обчислювально-інтенсивних задач, особливо у контексті полімовних застосунків.

Для системного розуміння факторів, що впливають на продуктивність WebAssembly у складних вебзастосунках, необхідно проаналізувати ключові параметри та їх взаємозв'язки. Дані параметри представлено в таблиці 1.

Таблиця 1

Порівняльний аналіз параметрів продуктивності WebAssembly

Параметр	JavaScript	WebAssembly	Приріст продуктивності	Тип застосунку
Швидкість виконання обчислень	Базова	2–10х швидше	200–1000 %	Математичні
Обробка зображень	Повільна	3–8х швидше	300–800 %	Мультимедійні
Криптографічні операції	Обмежена	5–15х швидше	500–1500 %	Безпека
Витрати пам'яті	Високі	Оптимізовані	30–60 % менше	Ігрові
Час завантаження	Швидкий	Повільніший	10–30 % більше	Всі типи

Джерело: розроблено на основі [1, 2, 3, 6]

Аналіз даних таблиці 1 демонструє суттєві переваги WebAssembly у продуктивності обчислень при незначному збільшенні часу завантаження, що підтверджує доцільність застосування технології для складних вебзастосунків. Harnes H., Morrison D. [2, с. 15] підкреслюють важливість методів статичного та динамічного аналізу для оцінки продуктивності WebAssembly. Їх дослідження показує, що комбінування різних підходів до аналізу дозволяє точніше оцінити реальну ефективність застосунків.

Для візуалізації архітектурного впливу WebAssembly на продуктивність складних вебзастосунків розроблено структурну модель, представлену на рисунку 1.

Структурна модель на рисунку 1 ілюструє ієрархічну взаємодію компонентів системи, де WebAssembly відповідає за обчислювально-інтенсивні операції, а JavaScript керує логікою користувацького інтерфейсу, що забезпечує оптимальний розподіл навантаження. Akieva Z. M., Murzin D. A., Eltaev S. I. [3, с. 3] підтверджують, що застосування WebAssembly для інтерактивних вебзастосунків дозволяє досягти значного покращення відгуку системи. Robati Shirzad M., Lam P. [4, с. 5] розробили інноваційний підхід до аналізу WebAssembly програм, який дозволяє ефективніше оптимізувати код для конкретних задач. Wang W. [5, с. 2] досліджував питання готовності сучасних вебзастосунків до впровадження WebAssembly та виявив, що основні перешкоди пов'язані з інтеграційними аспектами та інструментарієм розробки.

Burkhart N., Liao W., Guzide O. [6, с. 3] надали детальний огляд архітектури WebAssembly, що є важливим для розуміння принципів оптимізації продуктивності. Rao N. S. [7, с. 1978] акцентує на революційному потенціалі WebAssembly для розробки користувацьких інтерфейсів, особливо у контексті кросплатформенної інтеграції. Long Y., Su Y., Jiang Z. [8, с. 2] розробили спеціалізований інструмент WACP для профілювання продуктивності, який демонструє практичні підходи до оцінки ефективності WebAssembly у реальних умовах.

Waseem M., Das T., Ahmad A., Liang P., Mikkonen T. [9, с. 4] провели емпіричне дослідження проблем WebAssembly-застосунків, що виявило основні виклики: складність налагодження, обмежений інструментарій та проблеми з керуванням пам'яттю. Ці результати критично важливі для розробки стратегій оптимізації продуктивності.

Cabrera-Arteaga J., Fitzgerald N., Monperrus M., Baudry B. [10, с. 3] представили WASM-MUTATE, який забезпечує швидку бінарну диверсифікацію WebAssembly, що має значення для оптимізації безпеки без втрати продуктивності.

Практичне застосування WebAssembly у складних вебзастосунках демонструє найбільшу ефективність у сценаріях з інтенсивними обчисленнями: обробка зображень та відео, математичне моделювання, криптографічні операції, ігрові движки та симуляції. Ключовими факторами успішної інтеграції є правильний розподіл функціональності між JavaScript та WebAssembly, оптимізація передачі даних між модулями та вибір відповідних компіляторів.

Висновки

Проведене дослідження підтверджує значний потенціал WebAssembly для оптимізації продуктивності складних вебзастосунків. Технологія забезпечує прискорення виконання обчислювально-інтенсивних операцій у 2–15 разів порівняно з традиційними JavaScript-рішеннями залежно від типу задач. Найбільша ефективність досягається у математичних обчисленнях, обробці мультимедіа та криптографічних операціях. Ключовими факторами успішного впровадження WebAssembly є оптимальний розподіл функціональності між компонентами системи, мінімізація накладних витрат на міжмодульну взаємодію та правильний вибір архітектурних рішень.

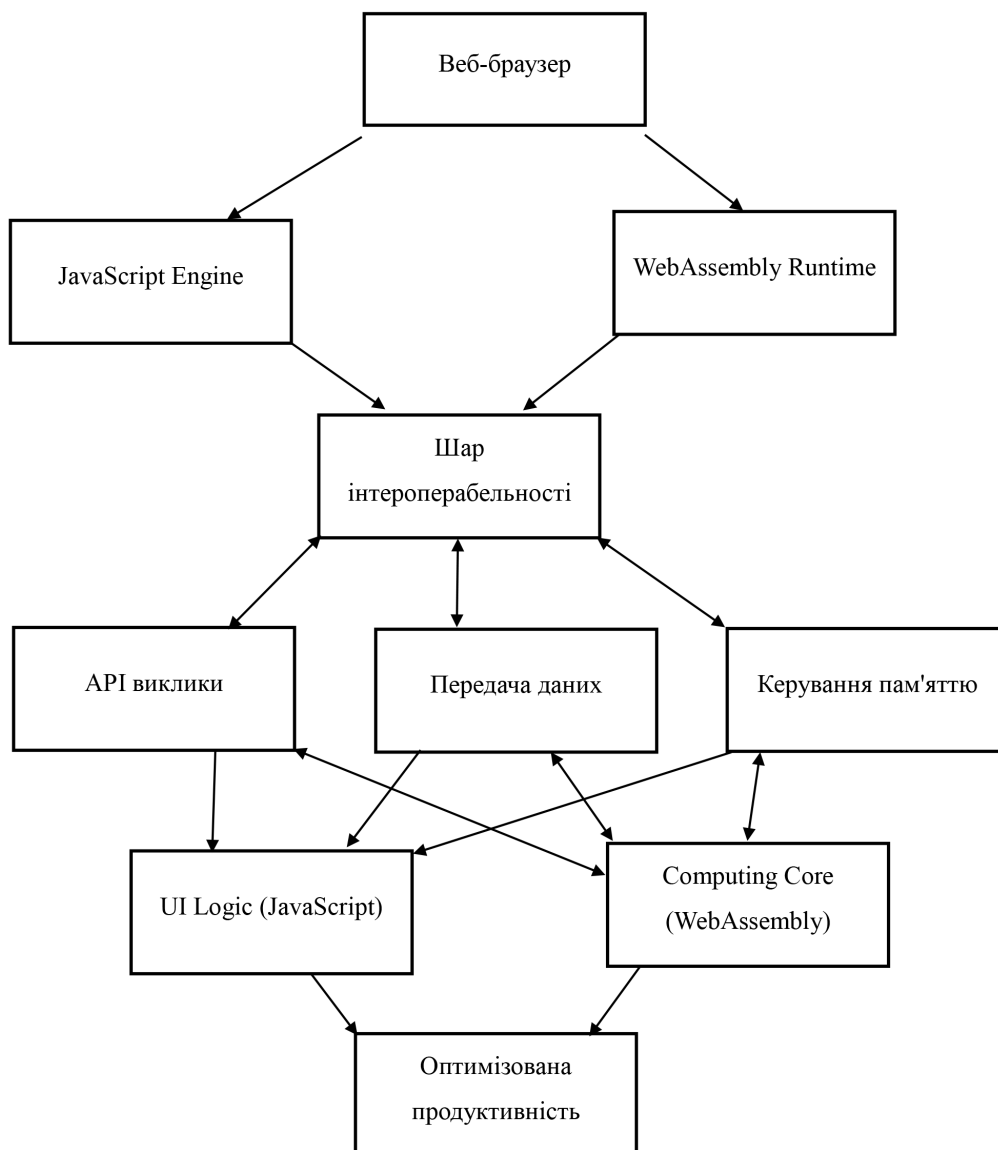


Рис. 1. Структурна модель впливу WebAssembly на продуктивність веб-застосунків

Джерело: авторська розробка

Виявлено, що WebAssembly найефективніше працює у гібридних архітектурах, де JavaScript відповідає за логіку користувацького інтерфейсу, а WebAssembly – за обчислювальне ядро.

Рекомендації для практичного застосування включають ретельний аналіз профілю навантаження застосунку, використання спеціалізованих інструментів профілювання продуктивності та поетапний підхід до міграції критично важливих компонентів. WebAssembly демонструє високу перспективність для майбутнього розвитку високопродуктивних вебзастосунків, що можуть конкурувати з нативними рішеннями за ефективністю при збереженні переваг вебплатформа.

Список використаної літератури

1. Sipek M., Muharemagic D., Mihaljevic B., Radovan A. Next-generation Web Applications with WebAssembly and TruffleWasm. *Proceedings of the 44th International Convention on Information, Communication and Electronic Technology (MIPRO)*. 2021. P. 1695–1700. DOI: <https://doi.org/10.23919/MIPRO52101.2021.9596883> URL: <https://ieeexplore.ieee.org/document/9596883>
2. Harnes H., Morrison D. SoK: Analysis techniques for WebAssembly. *Future Internet*. 2024. Vol. 16, iss. 3. P. 1–27. DOI: <https://doi.org/10.3390/fi16030084> URL: <https://www.mdpi.com/1999-5903/16/3/84>

3. Akieva Z. M., Murzin D. A., Eltaev S. I. Using WebAssembly to Improve the Performance of Interactive Web Applications. *Economy: Yesterday, Today, Tomorrow*. 2024. DOI: 10.36871/ek.up.p.r.2024.12.07.021 URL: <https://is.gd/sKPlhB>
4. Robati Shirzad M., Lam P. WasmWalker: Path-based Code Representations for Improved WebAssembly Program Analysis. arXiv:2410.08517. DOI: <https://doi.org/10.48550/arXiv.2410.08517> URL: <https://arxiv.org/abs/2410.08517>
5. Wang W. Empowering Web Applications with WebAssembly: Are We There Yet?. *Proceedings of the 36th IEEE/ACM International Conference on Automated Software Engineering*. 2021. DOI: <https://doi.org/10.1109/ASE51524.2021.9678831> URL: <https://is.gd/fUqdDA>
6. Burkhart N., Liao W., Guzide O. An Overview of WebAssembly. *Proceedings of the West Virginia Academy of Science*. 2020. Vol. 92, iss. 1. DOI: <https://doi.org/10.55632/pwvas.v92i1.682> URL: <https://is.gd/R9Omvk>
7. Rao N. S. WebAssembly: Revolutionizing Web User Interface Development through Performance and Cross-Language Integration. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*. 2024. Vol. 10, iss. 6. P. 1973–1981. DOI: <https://doi.org/10.32628/CSEIT241061235> URL: <https://is.gd/PNg9lo>
8. Long Y., Su Y., Jiang Z. WACP: A Performance Profiling Tool for WebAssembly-Python Interoperability. *Proceedings of the 15th Asia-Pacific Symposium on Internetware*. 2024. DOI: <https://doi.org/10.1145/3671016.3671408> URL: <https://dl.acm.org/doi/10.1145/3671016.3671408>
9. Waseem M., Das T., Ahmad A., Liang P., Mikkonen T. Issues and Their Causes in WebAssembly Applications: An Empirical Study. *Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering*. 2024. DOI: <https://doi.org/10.48550/arXiv.2311.00646> URL: <https://arxiv.org/abs/2311.00646>
10. Cabrera-Arteaga J., Fitzgerald N., Monperrus M., Baudry B. WASM-MUTATE: Fast and Effective Binary Diversification for WebAssembly. *Computers & Security*. 2024. Vol. 139. DOI: <https://doi.org/10.1016/j.cose.2024.103731> URL: <https://is.gd/zmwKsG>
11. Козуб Г. О., Сурма Ю. Ю., Артеменко О. І. Роль вебкомпонентів у побудові сучасних інтерфейсів: переваги та обмеження. *Вісник Херсонського національного технічного університету*. Том 2 № 2(93), 2025. С. 175–180. <https://doi.org/10.35546/kntu2078-4481.2025.2.2.21>
12. Вередін М. О., Козуб Г. О. Особливості UX/UI-дизайну при розробці вебсайтів та мобільних застосунків. збірник наукових праць з матеріалами IX Міжнародної наукової конференції, м.Ужгород, 4липня, 2025р. Міжнародний центр наукових досліджень. Вінниця: ТОВ «УКРЛОГОС Груп, 2025. С. 122-126. <https://doi.org/10.62731/mcnd-04.07.2025.005>

References

1. Sipek, M., Muharemagic, D., Mihaljevic, B., & Radovan, A. (2021) Next-generation web applications with WebAssembly and TruffleWasm. *Proceedings of the 44th International Convention on Information, Communication and Electronic Technology (MIPRO)*, 1695–1700. DOI: <https://doi.org/10.23919/MIPRO52101.2021.9596883> Retrieved from <https://ieeexplore.ieee.org/document/9596883> [in English].
2. Harnes, H., & Morrison, D. (2024) SoK: Analysis techniques for WebAssembly. *Future Internet*, 16(3), 1–27. DOI: <https://doi.org/10.3390/fi16030084> Retrieved from <https://www.mdpi.com/1999-5903/16/3/84> [in English].
3. Akieva, Z. M., Murzin, D. A., & Eltaev, S. I. (2024) Using WebAssembly to improve the performance of interactive web applications. *Economy: Yesterday, Today, Tomorrow*. DOI: 10.36871/ek.up.p.r.2024.12.07.021 Retrieved from <https://is.gd/sKPlhB> [in English].
4. Robati Shirzad, M., & Lam, P. (2024) WasmWalker: Path-based code representations for improved WebAssembly program analysis. arXiv. DOI: <https://doi.org/10.48550/arXiv.2410.08517> Retrieved from <https://arxiv.org/abs/2410.08517> [in English].
5. Wang, W. (2021) Empowering web applications with WebAssembly: Are we there yet? *Proceedings of the 36th IEEE/ACM International Conference on Automated Software Engineering*. DOI: <https://doi.org/10.1109/ASE51524.2021.9678831> Retrieved from <https://ieeexplore.ieee.org/document/9678831> [in English].
6. Burkhart, N., Liao, W., & Guzide, O. (2020) An overview of WebAssembly. *Proceedings of the West Virginia Academy of Science*, 92(1). DOI: <https://doi.org/10.55632/pwvas.v92i1.682> Retrieved from <https://pwvas.org/index.php/pwvas/article/view/682> [in English].
7. Rao, N. S. (2024) WebAssembly: Revolutionizing web user interface development through performance and cross-language integration. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 10(6), 1973–1981. DOI: <https://doi.org/10.32628/CSEIT241061235> Retrieved from <https://is.gd/PNg9lo> [in English].

8. Long, Y., Su, Y., & Jiang, Z. (2024) WACP: A performance profiling tool for WebAssembly-Python interoperability. *Proceedings of the 15th Asia-Pacific Symposium on Internetware*. DOI: <https://doi.org/10.1145/3671016.3671408> Retrieved from <https://dl.acm.org/doi/10.1145/3671016.3671408> [in English].
9. Waseem, M., Das, T., Ahmad, A., Liang, P., & Mikkonen, T. (2024) Issues and their causes in WebAssembly applications: An empirical study. *Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering*. DOI: <https://doi.org/10.48550/arXiv.2311.00646> Retrieved from <https://arxiv.org/abs/2311.00646> [in English].
10. Cabrera-Arteaga, J., Fitzgerald, N., Monperrus, M., & Baudry, B. (2024) WASM-MUTATE: Fast and effective binary diversification for WebAssembly. *Computers & Security*, 139. DOI: <https://doi.org/10.1016/j.cose.2024.103731> Retrieved from <https://is.gd/zmwKsG> [in English].
11. Kozub, G.O., Surma, Yu. Yu., Artemenko, O.I. (2025) Rol'vebkomponentiv u pobudovi suchasnykh interfeysiv: perevahy ta obmezheniya [The role of web components in building modern interfaces: advantages and limitations]. *Bulletin of the Kherson National Technical University*, Volume 2 No. 2(93), P. 175–180. <https://doi.org/10.35546/kntu2078-4481.2025.2.2.21> [in Ukrainian].
12. Veredin, M.O., Kozub, G.O. (2025) Osoblyvosti UX/UI-dyzaynu pryrozrobsi veb saytiv ta mobil'nykh zastosunkiv [Features of UX/UI design in the development of websites and mobile applications]. Collection of scientific papers with materials of the IX International Scientific Conference, Uzhgorod, July 4, 2025. International Center for Scientific Research. Vinnytsia: LLC «UKRLOGOS Group, P. 122-126. <https://doi.org/10.62731/mcnd-04.07.2025.005> [in Ukrainian].

Дата першого надходження рукопису до видання: 26.09.2025
Дата прийнятого до друку рукопису після рецензування: 21.10.2025
Дата публікації: 28.11.2025