

<https://doi.org/10.31891/2307-5732-2025-359-103>
УДК 004.738.5:004.4'22

НЕДІЛЬКО ОЛЬГА

Луцький національний технічний університет
<https://orcid.org/0000-0002-7416-8899>
e-mail: o.nedilko@lutsk-ntu.com.ua

САЧУК ВІКТОРІЯ

Луцький національний технічний університет
<https://orcid.org/0009-0009-9532-7586>
e-mail: sulim.v2608@lntu.edu.ua

ПОРІВНЯННЯ JAVASCRIPT-ФРЕЙМВОРКІВ REACT, ANGULAR І VUE.JS У ДОСЛІДЖЕННІ ПРОДУКТИВНОСТІ ТА МАСШТАБОВАНOSTІ ВЕБ-ДОДАТКІВ

JavaScript-фреймворки стали невід'ємною частиною сучасної веб-розробки. Серед найпопулярніших – React, Angular і Vue.js. У статті визначено, як ці фреймворки впливають на продуктивність і масштабованість додатків, здійснено порівняння їх сильних та слабких сторін.

Ключові слова: JavaScript, фреймворк, веб-розробка, React, Angular, Vue.js.

NEDILKO OLHA

SACHUK VICTORIA

Luts'k National Technical University

COMPARISON OF JAVASCRIPT FRAMEWORKS REACT, ANGULAR AND VUE.JS IN THE STUDY OF PERFORMANCE AND SCALABILITY OF WEB APPLICATIONS

The article focuses on the comparative study of three of the most widely used JavaScript frameworks—React, Angular, and Vue.js—in the context of web application performance and scalability. In recent years, JavaScript frameworks have become the foundation of modern web development, providing developers with efficient tools for building dynamic and interactive user interfaces. Each framework offers its own architectural approach, ecosystem, and performance characteristics, which significantly influence the overall efficiency and maintainability of a web project.

React, developed by Facebook, is a library oriented toward creating reusable UI components. Its virtual DOM implementation ensures high rendering performance, while its unidirectional data flow simplifies debugging and enhances predictability. However, scaling large applications often requires integrating additional libraries for routing, state management, and data handling. Angular, supported by Google, represents a full-featured framework with a well-defined structure, two-way data binding, and dependency injection mechanisms. These features make Angular suitable for enterprise-level solutions and large teams, though its steep learning curve and complex syntax can present challenges for newcomers.

Vue.js, in turn, combines the advantages of React's component-based design and Angular's declarative templates, offering an easy learning path and high adaptability. Its lightweight core and excellent documentation make Vue a popular choice for small to medium projects or rapid prototyping.

The analysis concludes that the selection of a JavaScript framework should be based on the project's complexity, team expertise, and scalability goals. React is best suited for flexible, performance-oriented solutions; Angular excels in large structured applications; and Vue.js offers an optimal balance between simplicity and functionality. Understanding these distinctions allows developers to optimize web application development and ensure efficient long-term maintenance.

Keywords: JavaScript, framework, web development, React, Angular, Vue.js.

Стаття надійшла до редакції / Received 15.10.2025

Прийнята до друку / Accepted 15.11.2025

Постановка проблеми у загальному вигляді

та її зв'язок із важливими науковими чи практичними завданнями

У сучасній розробці веб-додатків вибір відповідного JavaScript-фреймворку є критичним рішенням, яке впливає на продуктивність, масштабованість і стабільність кінцевого продукту. React, Angular і Vue.js є найпопулярнішими фреймворками, кожен із яких пропонує унікальні підходи до створення додатків, але їх порівняння в контексті продуктивності та масштабованості залишається актуальним завданням. Постановка проблеми полягає в необхідності об'єктивної оцінки цих технологій для вибору оптимального інструменту в умовах конкретних проєктних вимог.

Розв'язання цієї проблеми має важливе значення як з наукової, так і з практичної точки зору:

1) Проведення емпіричних досліджень, які дозволяють кількісно оцінити продуктивність (швидкість завантаження, обробки даних, рендерингу) та масштабованість (можливість адаптації під більші навантаження) веб-додатків, створених на кожному з фреймворків.

2) Розробка критеріїв та методологій порівняння фреймворків з урахуванням реальних сценаріїв використання.

3) Надання рекомендацій розробникам, які дозволять оптимально обрати фреймворк залежно від проєктних вимог.

4) Підвищення ефективності процесів розробки завдяки зменшенню часу на вибір технології та уникненню ризиків невідповідності інструменту.

5) Зниження витрат на подальшу підтримку та модернізацію веб-додатків завдяки використанню фреймворку, що забезпечує найкращу продуктивність у конкретних умовах.

Наприклад, React часто використовується для створення компонентно-орієнтованих додатків, Angular

пропонує комплексний інструментарій, а Vue.js є гнучким і легким для впровадження. Проте, відсутність ґрунтового порівняння їх ефективності в однакових умовах створює прогалину, яку ми прагнемо заповнити.

Аналіз досліджень та публікацій

На сьогодні існує значна кількість досліджень і публікацій, присвячених порівнянню JavaScript-фреймворків Angular, React та Vue.js у контексті їх продуктивності, масштабованості та практичного застосування. У роботах дослідників часто акцентується увага на архітектурних особливостях кожного фреймворку, їхньому впливові на швидкодію веб-додатків і зручність роботи з великими даними.

Вчені Адамова В. та Кузнецов Д. у своїй статті дослідили значення та переваги таких фреймворків, як: React, Angular, Vue.js у контексті створення Web сторінок. Ними продемонстровано дані, які засвідчують переваги фреймворку React серед інших [1, с.78].

Продовжують вивчати JavaScript рішення для розробки веб-додатків вчені Малохвій Е., Бугай В., Молчанов Г. та Черних О. Для визначення дієвості фреймворків React та Vue, вчені провели аналіз таких інструментів, як Google Trends, Stack Overflow Developer Survey, GitHub Stars, The State of JavaScript. На їхній основі визначено, що React вирізняється високою популярністю та підтримкою великої корпорації, що підвищує його цінність для ринку праці. Vue натомість легше освоїти, забезпечує швидкий старт у розробці, пропонує багатий набір вбудованих інструментів і вирізняється меншою вагою. Крім того, Vue інтегрує передові підходи від інших успішних фреймворків, дозволяючи розробникам поєднувати найкращі практики в одному рішенні [2, с. 58].

У дослідженні Н. Ічанської увагу зосереджено на сучасних технологіях розробки веб-додатків та їх застосуванні для оцінки рівня знань. Розглянуто актуальні інструменти створення веб-додатків і виконано порівняльний аналіз найпоширеніших фреймворків у сфері веб-розробки, якими стали JavaScript фреймворки Angular, React та Vue [3, с. 101]. У публікаціях підкреслюється комплексний підхід до створення веб-додатків. Автор відзначає переваги вбудованих інструментів, таких як маршрутизація, управління станом і двостороннє зв'язування даних, які забезпечують високу масштабованість.

Більшість сучасних досліджень зосереджуються на емпіричному порівнянні продуктивності фреймворків у різних умовах. Автори аналізують час завантаження, швидкість рендерингу, управління станом і використання ресурсів у тестових середовищах. Також популярними є огляди практичних кейсів, у яких описується використання кожного фреймворку для створення конкретних додатків, зокрема електронної комерції, корпоративних платформ чи SaaS-рішень.

Разом із тим існує недостатність досліджень, які б детально аналізували довгострокові аспекти, такі як витрати на підтримку, адаптацію до нових технологій або зручність у роботі з мікросервісною архітектурою. Це відкриває перспективи для подальших розвідок у даному напрямі, особливо з огляду на швидкий розвиток JavaScript-фреймворків і їхню інтеграцію з інноваційними підходами в розробці веб-додатків.

Формулювання цілей статті

Метою роботи є: проведення порівняльного аналізу JavaScript-фреймворків React, Angular і Vue.js з точки зору продуктивності та масштабованості веб-додатків для обґрунтування їхнього вибору у залежності від специфічних потреб проєктів.

Виклад основного матеріалу

JavaScript-фреймворки React, Angular і Vue.js є одними з найпопулярніших у веб-розробці. Кожен із них має свої особливості, переваги та недоліки, що впливають на вибір у контексті продуктивності та масштабованості веб-додатків. Розглянемо їхні основні характеристики, переваги та недоліки.

React є популярною JavaScript-бібліотекою для створення інтерфейсів користувача, розроблена компанією Facebook. Її основними перевагами є висока продуктивність, гнучкість і компонентний підхід. React використовує Virtual DOM, що дозволяє ефективно оновлювати інтерфейс, порівнюючи зміни у віртуальній копії DOM і оновлюючи лише ті елементи, які зазнали змін. Це забезпечує швидкий рендеринг навіть у складних додатках із великою кількістю компонентів.

Компонентна архітектура React дозволяє розробникам розбивати інтерфейс на незалежні компоненти, що сприяє повторному використанню коду, спрощенню підтримки й масштабуванню додатків. Крім того, React забезпечує унідирекційний потік даних, що робить управління станом передбачуваним і зручним. JSX, який використовується в React, дозволяє поєднувати JavaScript із HTML у коді, спрощуючи написання компонентів.

React має безліч переваг. Завдяки Virtual DOM продуктивність додатків залишається на високому рівні, навіть якщо інтерфейс дуже складний. Гнучкість React дозволяє інтегрувати його з іншими бібліотеками або фреймворками, створюючи унікальні рішення для конкретних задач. Широка екосистема бібліотек і інструментів робить React ще більш потужним інструментом для веб-розробників. Масштабованість забезпечується завдяки компонентній архітектурі та додатковим інструментам, як-от Redux або Context API, що дозволяють ефективно управляти станом у великих проєктах. Окремою перевагою є можливість створення мобільних додатків за допомогою React Native, що використовує ті самі концепції, що й веб-додатки [4].

Попри переваги, React має й недоліки. Оскільки це лише бібліотека, для створення повноцінного додатка потрібні сторонні інструменти для управління станом, маршрутизації чи роботи з API, що збільшує час на налаштування. Також новачкам може бути складно освоїти React через необхідність розуміння додаткових концепцій, таких як JSX, Virtual DOM, хуки тощо. Активний розвиток бібліотеки іноді ускладнює підтримку старих проєктів через часті зміни в API. У великих проєктах, навіть із використанням Redux, управління станом може бути викликом.

Названий фреймворк є потужним інструментом для створення сучасних веб-додатків із високими вимогами до продуктивності та гнучкості. Це чудовий вибір як для невеликих проєктів, так і для масштабованих додатків, якщо правильно організувати структуру коду та вибрати необхідні додаткові бібліотеки.

Наступним фреймворком для вивчення став Angular, що є потужним інструментом для створення масштабованих веб-додатків, особливо в корпоративному середовищі, завдяки своїй компонентно-орієнтованій архітектурі та використанню TypeScript. Його основними особливостями є модульна структура, яка спрощує командну розробку, двостороннє зв'язування даних для синхронізації моделі та інтерфейсу, а також декларативне програмування, що базується на розширеннях HTML. Angular пропонує широкий набір інструментів «із коробки», зокрема для маршрутизації, обробки форм та HTTP-запитів, а також інтеграцію з RxJS для реактивного програмування. Завдяки Ahead-of-Time (AOT) компіляції Angular оптимізує продуктивність, дозволяючи компілювати код до JavaScript на етапі розробки, що зменшує час завантаження додатка. Крім того, підтримка Lazy Loading забезпечує динамічне завантаження модулів, зменшуючи початковий обсяг даних для завантаження [5].

Angular особливо підходить для розробки великих корпоративних проєктів завдяки масштабованості та чіткій модульній структурі, яка дозволяє ефективно організовувати роботу великих команд. Однак, незважаючи на свої переваги, Angular має й певні недоліки. Складність архітектури, використання TypeScript і велика кількість концепцій, таких як директиви, сервіси чи ін'єкція залежностей, створюють круту криву навчання. Це може уповільнити старт проєктів, особливо для новачків. Крім того, двостороннє зв'язування даних і складна система перевірки змін іноді призводять до високих витрат ресурсів у додатках із великою кількістю компонентів та інтерактивних подій. Початковий розмір Angular-додатків, через інтеграцію численних вбудованих функцій, часто більший, ніж у React або Vue.js, що може впливати на час завантаження [6].

Оновлення між основними версіями Angular іноді є трудомістким процесом, особливо для масштабних проєктів, що вимагає ретельного планування. У цілому Angular є оптимальним вибором для великих і складних проєктів, які потребують потужних інструментів для управління масштабованістю та продуктивністю. Проте його використання в малих або середніх проєктах може бути недоцільним через складність освоєння та надлишковість функціоналу.

Останнім фреймворком для аналізу є Vue.js. Цей прогресивний JavaScript-фреймворк вирізняється своєю простотою, гнучкістю та легкістю інтеграції в проєкти різного масштабу. У контексті дослідження продуктивності та масштабованості веб-додатків Vue.js пропонує низку важливих особливостей, переваг і недоліків, які впливають на його застосування. Однією з ключових особливостей Vue.js є його простота у використанні: зрозумілий синтаксис, заснований на HTML, CSS та JavaScript, забезпечує низький поріг входу для новачків і водночас залишається потужним інструментом для досвідчених розробників. Vue.js базується на компонентно-орієнтованій архітектурі, яка дозволяє створювати модульні додатки зі зручною організацією та повторним використанням коду. Він підтримує як одностороннє, так і двостороннє зв'язування даних, що дає розробникам можливість вибирати оптимальний підхід для кожного проєкту. Крім того, реактивна система Vue.js автоматично оновлює DOM при зміні стану додатка, підвищуючи продуктивність і спрощуючи управління даними.

Однією з найбільших переваг Vue.js є його висока продуктивність. Завдяки невеликому розміру ядра (близько 20 КБ) Vue.js забезпечує швидке завантаження додатків, що особливо важливо для мобільних користувачів або при слабкому інтернет-з'єднанні. Ефективний рендеринг і оновлення DOM завдяки використанню віртуального DOM також сприяють швидкій роботі додатків. Vue.js добре підходить як для невеликих, так і для середніх проєктів, забезпечуючи мінімальне використання ресурсів для базових операцій. У контексті масштабованості Vue.js також демонструє свої переваги, оскільки дозволяє організувати код у модульній структурі, розподіляючи логіку між компонентами. Це забезпечує зрозумілу архітектуру та спрощує розвиток великих проєктів. Інструменти Vue CLI, Vue Router та Vuex допомагають ефективно управляти станом та маршрутизацією навіть у великих додатках [7].

Таблиця 1

Порівняння JavaScript-фреймворків			
Критерій/фреймворк	React	Angular	Vue.js
Продуктивність	Висока завдяки Virtual DOM	Оптимізована через AOT і Zone.js	Висока завдяки Virtual DOM
Масштабованість	Гнучка, потребує додаткових інструментів	Відмінна для великих проєктів	Добра для середніх і великих проєктів
Простота навчання	Середня, потрібно розуміти екосистему	Складна для новачків	Проста, навіть для новачків
Гнучкість	Висока	Обмежена суворюю структурою	Дуже висока
Спільнота	Велика	Велика	Менша, але активна

Водночас Vue.js має деякі недоліки, які слід враховувати при виборі фреймворку. Хоча його гнучкість є перевагою, вона також може створювати проблеми при роботі над масштабними проєктами, де відсутність чітких стандартів може призводити до неузгодженості коду. Крім того, Vue.js має менше поширення, ніж React чи Angular, що може обмежувати доступ до розширених бібліотек або інструментів. Для дуже великих корпоративних проєктів Vue.js може бути менш оптимальним через меншу інтеграцію з рішеннями на кшталт TypeScript у порівнянні з Angular. Незважаючи на це, Vue.js залишається чудовим вибором для проєктів, які потребують швидкого старту, високої продуктивності та гнучкості, особливо для невеликих і середніх команд.

Узагальнені дані таблиці щодо порівняння досліджуваних фреймворків дає можливість усвідомити їхнє значення для різних видів проєктів та цілей. Таким чином, кожен із фреймворків має свої сильні сторони, що дозволяє використовувати їх у відповідних сценаріях для досягнення оптимальних результатів.

Висновки з даного дослідження

і перспективи подальших розвідок у даному напрямі

Проведений аналіз JavaScript-фреймворків Angular, React та Vue.js у контексті продуктивності та масштабованості веб-додатків дозволяє зробити кілька ключових висновків. Angular демонструє високу масштабованість і є оптимальним вибором для великих корпоративних проєктів завдяки вбудованим рішенням і модульній архітектурі. Однак його складність і високий поріг входу обмежують використання для невеликих команд або проєктів із обмеженими ресурсами. React забезпечує чудову продуктивність і гнучкість завдяки використанню віртуального DOM і компонентного підходу. Цей фреймворк підходить для широкого спектра проєктів, але для забезпечення функціоналу, такого як маршрутизація або управління станом, потребує додаткових бібліотек і налаштувань. Vue.js виділяється своєю простотою, легкістю інтеграції та високою продуктивністю, що робить його ідеальним для швидкого старту невеликих і середніх проєктів. Проте для великих проєктів його застосування може бути менш оптимальним через обмежену екосистему та відсутність чітких стандартів.

Таким чином, вибір фреймворку повинен базуватися на специфічних вимогах проєкту, таких як розмір команди, масштаб проєкту, обсяг інтерактивності та довгострокові перспективи розвитку. Подальші дослідження в цьому напрямі відкривають нові перспективи. Зокрема, важливо поглиблено дослідити реальну продуктивність фреймворків у складних умовах із великим навантаженням і обсягами даних. Необхідно також проаналізувати довгострокові витрати на підтримку додатків, створених на різних фреймворках, що дозволить оцінити їх економічну ефективність. Окрім цього, актуальним є вивчення нових функцій і технологічних змін у наступних версіях фреймворків, таких як React Server Components чи розширена інтеграція TypeScript у Vue.js.

Література

8. Адамова В. С., Кузнецов Д. І. Порівняльний аналіз сучасних javascript фреймворків. Комп'ютерні інтелектуальні системи та мережі / В. С. Адамова, Д. І. Кузнецов Д. І. // Матеріали XIII Всеукраїнської науково-практичної WEB конференції аспірантів, студентів та молодих вчених (24-26 березня 2020 р.). – Кривий Ріг: Криворізький національний університет. – 2020. – С. 77–79.
9. Малохвій Е. Е., Бугай В. С., Молчанов Г. І., Черних О. П. Аналітичний огляд та порівняння сучасних Javascript рішень для розробки веб-додатків / Е. Е. Малохвій, В. С. Бугай, Г. І. Молчанов, О. П. Черних // Системи управління, навігації та зв'язку. – 2021. – Випуск 4(66). – С. 55–58.
10. Ічанська Н.В. Засоби та інструменти для розробки веб-додатку з тестування рівня знань студентів / Н.В. Ічанська // Механіка та математичні методи. – 2024. – Т. 6. Вип. 1. – С. 95–106.
11. React.js [Електронний ресурс] – Режим доступу до ресурсу: <https://reactjs.org/>.
12. Williamson K. Learning AngularJS. // Williamson K. – O'Reilly Media, 2015 – 212 p.
13. Angular [Електронний ресурс] – Режим доступу до ресурсу: <https://angular.io/>.
14. Vue.js [Електронний ресурс] – Режим доступу до ресурсу: <https://vuejs.org/>.

References

1. Adamova V. S., Kuznietsov D. I. Porivnialnyi analiz suchasnykh javascript freimvorkiv. Kompiuterni intelektualni systemy ta merezhi / V. S. Adamova, D. I. Kuznietsov D. I. // Materialy XIII Vseukrainskoi naukovo praktychnoi WEB konferentsii aspirantiv, studentiv ta molodykh vchenykh (24-26 bereznia 2020 r.). – Kryvyi Rih: Kryvorizkyi natsionalnyi universytet. – 2020. – S. 77–79.
2. Malokhvii E. E., Buhai V. S., Molchanov H. I., Chernykh O. P. Analitichnyi ohliad ta porivniannia suchasnykh Javascript rishen dla rozrobky veb-dodatkov / E. E. Malokhvii, V. S. Buhai, H. I. Molchanov, O. P. Chernykh // Systemy upravlinnia, navihatsii ta zviazku. – 2021. – Vypusk 4(66). – S. 55–58.
3. Ichanska N.V. Zasoby ta instrumenty dla rozrobky veb-dodatku z testuvannia rivnia znan studentiv / N.V. Ichanska // Mekhanika ta matematychni metody. – 2024. – T. 6. Vyp. 1. – S. 95–106.
4. React.js [Elektronnyi resurs]. – Rezhym dostupu : <https://reactjs.org/>.
5. Williamson K. Learning AngularJS. // Williamson K. – O'Reilly Media, 2015 – 212 p.
6. Angular [Elektronnyi resurs]. – Rezhym dostupu : <https://angular.io/>.
7. Vue.js [Elektronnyi resurs]. – Rezhym dostupu : <https://vuejs.org/>.