

УДК 004.4

Maksym Zhmutskyi ¹, Yuliya Bykova ²

¹ student of group CST-124m NU “Zaporizhzhia Polytechnic”

² senior teacher NU “Zaporizhzhia Polytechnic”

COMPARISON OF MODERN WEB FRAMEWORKS: PERFORMANCE, POPULARITY AND ARCHITECTURAL FEATURES

Modern web frameworks like React [1], Angular [2], and Vue.js [3] simplify building complex web applications by improving performance, scalability, and maintainability. React, from Facebook, uses a virtual DOM [4] for efficient updates, making it ideal for dynamic, large-scale apps. Angular, by Google, employs a real DOM and built-in two-way data binding, simplifying model-view synchronization but potentially slowing performance in larger apps. Vue.js combines features of both, offering a lightweight, flexible balance between simplicity and efficiency. This study bridges the gap in comparisons by analyzing performance, usability, architecture, and popularity.

Performance was assessed via Lighthouse and Google PageSpeed Insights [5], measuring metrics like time to interactive (TTI). React achieved the fastest response time (1.4 seconds) with its virtual DOM, Angular was slower (2.1 seconds) due to its real DOM, and Vue.js, blending both approaches, had a midrange response time of 1.7 seconds (Figure 1).

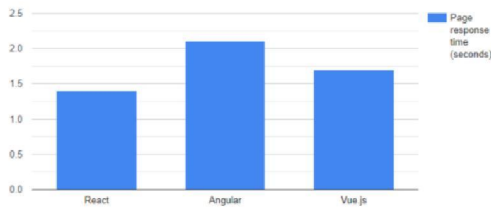


Figure 1 – Comparison of framework performance

Ease of use was evaluated by examining project setup complexity and the code needed for common tasks. React offers flexibility with library choices but requires extra configuration for state management and routing. Angular simplifies development with built-in tools but adds complexity due to its comprehensive structure. Vue.js, being lightweight, minimizes setup effort and boilerplate code. Starter package sizes reflect these differences: React's package is smaller than Angular's but slightly larger than Vue.js (Figure 2).

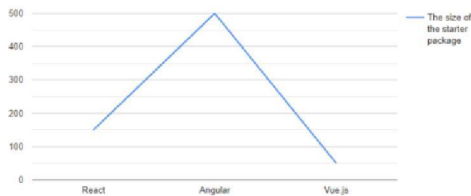


Figure 2 – Comparison of framework starter package sizes in kilobytes

Popularity was measured using GitHub stars and the Stack Overflow Developer Survey [6–7]. Vue.js leads with 210,000 stars, followed by React at 207,000, and Angular at 89,000. The survey shows React is used by 42% of developers, Angular by 23 %, and Vue.js by 18 %, highlighting React’s dominant position in the developer community (Figure 3).

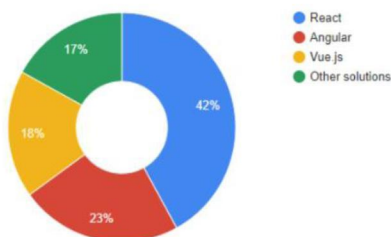


Figure 3 – Framework popularity distribution among developers

The frameworks' architectures were analyzed in detail. React's unidirectional data flow ensures predictability and easy debugging, with its component-based structure promoting code reuse. Angular's modular architecture and two-way data binding simplify forms but complicate scaling. Vue.js, as a hybrid, supports both unidirectional and bidirectional flows, offering flexibility in application design.

React stands out for versatility and performance (1.4 seconds average response time), thanks to its virtual DOM and scalable architecture. With 207,000 GitHub stars and 42 % developer usage, it is ideal for dynamic, large-scale applications requiring flexibility and maintainability. Angular, though slower (2.1 seconds) and complex, is well-suited for enterprise projects due to its integrated tools, modularity, and reliability, minimizing third-party risks but demanding more training for smaller teams. Vue.js offers a balanced solution with strong performance (1.7 seconds) and ease of use, making it attractive for medium-sized projects, startups, and resource-limited teams due to its clear documentation and flexible data flow options.

Future research will explore these frameworks under high-load conditions, focusing on WebAssembly integration for enhanced computation, long-term maintainability, and industry-specific scalability (e.g., e-commerce, healthcare). Evaluating their support for microservices, SSR, and PWAs is vital for modern development. Trends like AI-driven tools and compatibility with DevOps workflows (e.g., Docker, Kubernetes) will also be studied, helping developers make informed decisions for evolving needs.

REFERENCES

1. React [Electronic resource]. – Access mode : <https://react.dev/>
2. Angular [Electronic resource]. – Access mode : <https://angular.dev/>
3. Vue.js [Electronic resource]. – Access mode : <https://vuejs.org/>
4. Document Object Model [Electronic resource]. – Access mode : https://developer.mozilla.org/ru/docs/Web/API/Document_Object_Model/

5. PageSpeed Insights [Electronic resource]. – Access mode : <https://developers.google.com/speed/docs/insights/v5/get-started/>
6. GitHub [Electronic resource]. – Access mode : <https://github.com/>
7. Stack Overflow [Electronic resource]. – Access mode : <https://stackoverflow.com/>