MODERN TRENDS IN WEB DEVELOPMENT: HOW TECHNOLOGIES AND APPROACHES TO WEBSITE CREATION ARE CHANGING

Melnytskyi Ihor,
Software developer
Odessa, Ukraine

E-mail: melnitskiy.i.i@gmail.com

Abstract. Web development has changed significantly over the past 10 years due to advances in technology, changing user expectations, and the need for increased productivity. This article examines key trends such as the shift to serverless architecture, the use of frameworks to simplify development, static site generation, and the rise of web performance. It analyzes the technologies and approaches that have changed the process of creating websites, as well as their impact on the industry. Particular attention is paid to how these changes have affected the developer workflow, what new opportunities they have opened up, and what challenges they have posed for web development professionals. The article also provides a graph illustrating the growth of new technologies such as WebAssembly compared to traditional approaches.

Key words: web development trends, modern web development, JavaScript frameworks, WebAssembly, JAMstack, web performance optimization.

Introduction

Over the past decade, website development has evolved from a relatively simple process of coding and markup into a complex multidisciplinary task that requires taking into account many factors: performance, security, accessibility, and adaptability. The development of technologies such as JavaScript frameworks, serverless architecture, and tools for generating static sites has significantly simplified development, but at the same time has increased the requirements for the knowledge and skills of developers. Modern web applications must not only be functional, but also fast, secure, and convenient for users. This has led to the emergence of new approaches to development that have become the standard in the industry. In this article, we will consider how these changes have affected the process of creating websites, which technologies have become key, and what trends determine the future of web development. We will also analyze how new technologies such as WebAssembly compare to traditional approaches and present a graph illustrating their growth.

1. Key Technologies That Have Changed Web Development in the Last 10 Years

1.1 JavaScript Frameworks and Libraries

The emergence and popularity of frameworks such as React, Angular, and Vue.js have dramatically changed the way we build web applications. These tools have enabled developers to create dynamic, interactive interfaces with minimal effort. The component-based approach used in these frameworks has simplified code reuse and improved project maintainability. For example, React, with its virtual DOM and unidirectional data flow, has become the standard for building complex user interfaces. Angular, on the other hand, offers a full-fledged framework with built-in solutions for routing, state management, and dependency injection. Vue.js, in turn, has attracted attention for its simplicity and flexibility, which has made it popular among developers looking for a lightweight solution for their projects. These frameworks not only simplified development, but also set new standards for creating modern web applications.

1.2 Node.js and Full-Stack JavaScript

Node.js was a revolutionary technology that allowed JavaScript to be used on the server side. This led to the emergence of full-stack development, where one programming language is used for both the client and server parts of the application. This simplified the development process and lowered the entry barrier for new developers. Node.js also contributed to the development of the npm (Node Package Manager) ecosystem, which became the largest repository of libraries and tools for JavaScript. Thanks to this, developers got access to a huge number of ready-made solutions, which significantly accelerated the development process. In addition, Node.js became the basis for many modern technologies, such as server-side rendering (SSR) and serverless architecture, which we will discuss below.

1.3 WebAssembly

WebAssembly (Wasm) opened up new possibilities for web development, allowing high-performance computing right in the browser. This is especially useful for data-intensive applications such as games, video editors, and CAD systems. WebAssembly allows code written in languages such as C, C++, and Rust to be compiled into a binary format that can be executed in the browser with near-native performance. This makes it ideal for tasks that were previously impossible or highly inefficient in the browser. For example, projects such as Figma and AutoCAD use WebAssembly to provide high performance for their web versions. The technology continues to evolve and is likely to become even more important in the future, especially as interest in computationally intensive web applications grows.

2. Architectures That Simplify Website Development

2.1 Microservices Architecture

Microservices have replaced monolithic architectures by allowing applications to be broken down into small, independent components. This has made it easier to scale, test, and maintain large projects. Microservices architecture allows development teams to work on individual parts of an application independently, which speeds up the development process and the introduction of new features. For example, large companies like Netflix and Amazon use microservices to manage their complex systems. However, this architecture also requires careful planning and management, as it can lead to increased complexity in configuring and monitoring the system. However, for large projects, the benefits of microservices outweigh their drawbacks, making this architecture a popular choice for modern web applications.

2.2 Serverless Architecture

Serverless architecture, introduced by platforms like AWS Lambda, Google Cloud Functions, and Azure Functions, has freed developers from the need to manage servers. This has allowed them to focus on writing code, reducing development time and operational costs. Serverless architecture is based on the concept of functions as a service (FaaS), where code is executed only when needed and automatically scales depending on the load. This makes it ideal for applications with variable load, such as APIs, data processing, and task automation. For example, startups and small companies often use serverless to quickly launch their products without having to invest in infrastructure. However, serverless also has its limitations, such as cold start of functions and difficulty in debugging, which requires new skills and approaches from developers.

2.3 JAMstack

JAMstack (JavaScript, APIs, Markup) is an architecture based on generating static sites and using APIs for dynamic functions. It provides high performance, security, and scalability, which makes it popular for modern web projects. JAMstack allows developers to create sites that load instantly, as all pages are pre-generated and served as static files. APIs are used for dynamic functions, such as contact forms or search, which allows maintaining high performance. For example, companies like Smashing Magazine and Nike have switched to JAMstack for their websites, which has allowed them to significantly improve loading speeds and user experience. This architecture also simplifies the deployment process, as static files can be easily hosted on a CDN (Content Delivery Network), which ensures global availability and high-speed content delivery.

3. Simplify development with frameworks

Frameworks like Next.js, Gatsby, and Nuxt.js have made the process of building web applications much easier. They provide out-of-the-box solutions for routing, performance optimization, and SEO, allowing developers to focus on business logic. For example, Next.js supports both server-side rendering (SSR) and static generation (SSG), making it a versatile tool for building high-performance applications. SSR allows you to render pages on the server, which improves SEO and load times, while SSG allows you to generate static pages that can be deployed to a CDN for maximum performance. Gatsby, on the other hand, specializes in generating static sites and offers powerful data tools like GraphQL. Nuxt.js, inspired by Next.js, brings similar capabilities to the Vue.js ecosystem. These frameworks not only simplify development, but also set new standards for creating modern web applications, making them more productive and user-friendly.

4. Static Site Generation

Static sites are experiencing a renaissance thanks to tools like Hugo, Jekyll, and Gatsby. These generators allow you to create fast, secure, and easily scalable sites. Static sites are especially popular for blogs, documentation, and marketing pages where loading speed and ease of deployment are important. For example, Hugo, written in Go, is known for its fast page generation speed, making it an ideal choice for large projects with a lot of content. Jekyll, integrated with GitHub Pages, has become a popular tool for creating blogs and personal sites due to its simplicity and flexibility. Gatsby, as mentioned, offers powerful tools for working with data and integrating with various sources, such as CMS and APIs. Not only do these static site generators simplify the development process, but they also provide high performance and security, making them a popular choice for modern web projects.

5. Web Performance Growth

Performance has become one of the key success factors for web projects. Modern tools like Lighthouse, Web Vitals, and PageSpeed Insights help developers optimize their applications. Technologies like lazy loading, caching, and resource compression have become the standard for improving user experience. For example, lazy loading allows images and other resources to be loaded only when they are needed, which reduces the initial page load time. Caching, on the other hand, allows frequently used resources to be stored on the user's device, which reduces the number of requests to the server and speeds up page loading on repeat visits. Resource compression, such as minifying CSS and JavaScript, also plays an important role in improving performance. These technologies, along with modern standards like HTTP/2 and HTTP/3, allow you to create fast and responsive web applications, which is becoming critical in the face of growing user expectations.

6. New Technologies Usage Growth Chart

The chart below compares the growth of new technologies such as WebAssembly, serverless architecture, and JAMstack compared to traditional approaches over the past 10 years. The data is based on research and surveys conducted among developers and companies.

As you can see from the chart, WebAssembly usage has been growing rapidly since 2017, when the technology was officially supported by major browsers. Serverless architecture has also shown steady growth, especially since 2018, when major cloud providers began actively promoting their solutions. JAMstack, in turn, has become a popular choice for developers since 2019, due to its simplicity and high performance. Traditional approaches such as monolithic architectures and manual server management are gradually losing popularity, giving way to more modern and efficient solutions.

Conclusion

Current trends in web development demonstrate that the industry continues to evolve, offering new approaches and technologies for creating websites. Simplification of development with frameworks, the transition to serverless architecture and an emphasis on performance have become key factors determining the future of web development. These changes not only simplify the process of creating websites, but also improve the quality of the final product, which meets the growing expectations of users. In the future, we can expect further development of technologies such as WebAssembly, serverless and JAMstack, as well as the emergence of new tools and approaches that will continue to change the landscape of web development.

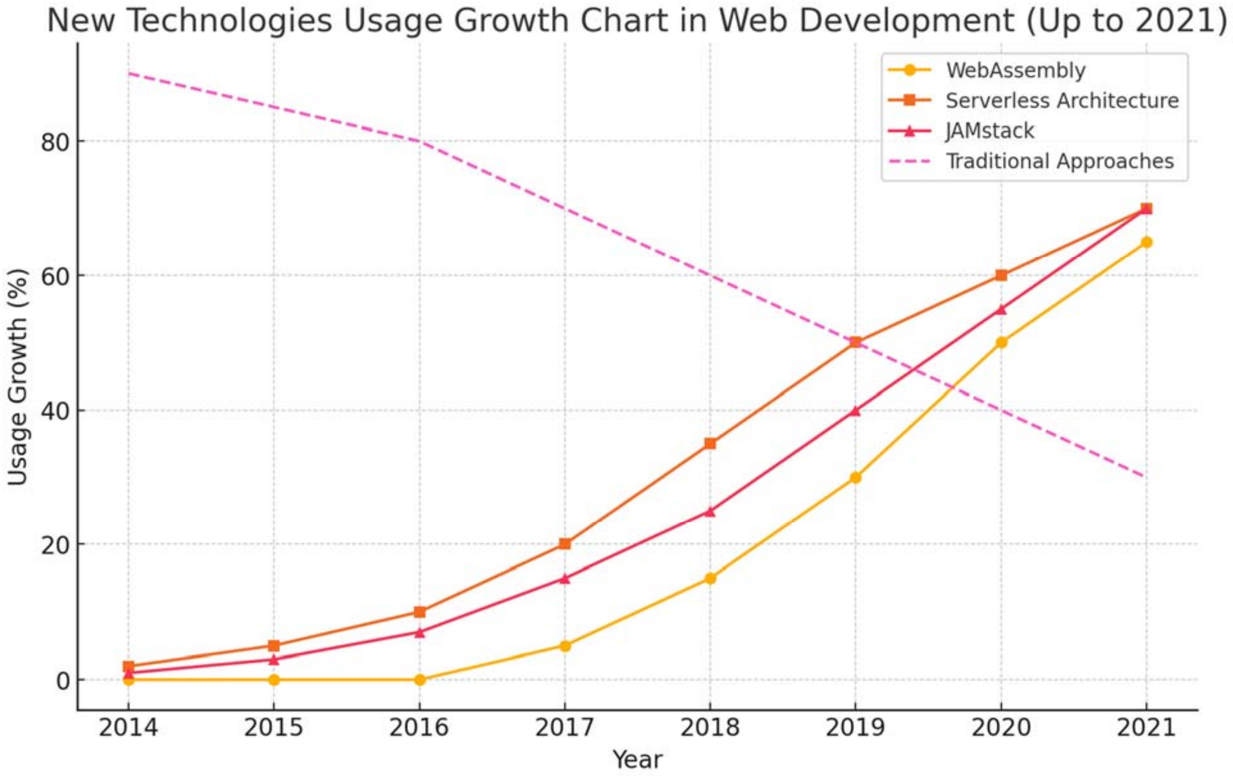

Fig. 1 New Technologies Usage Growth Chart in Web Development

References:

1. Richardson, L. (2020). "Microservices Patterns".
2. Fowler, M. (2019). "Serverless Architectures".
3. Hevery, M. (2021). "The Rise of JavaScript Frameworks".
4. Google Developers. (2021). "Web Vitals: Essential Metrics for a Healthy Site".
5. Smashing Magazine. (2021). "The JAMstack Revolution: Modern Web Development".
6. WebAssembly.org. (2021). "WebAssembly: A New Era for Web Performance".