International Journal of Computer Applications Technology and Research
Volume 12–Issue 10, 24 – 27, 2023, ISSN:-2319–8656
DOI:10.7753/IJCATR1210.1004

# Analyzing the Impact of Next.JS on Site Performance and SEO

Vishal Patel
Yahoo Inc!
Sunnyvale, CA, USA

**Abstract**: Next.JS is a robust framework that improves website performance and SEO through various optimization techniques. In addition to implementing Incremental Static Regeneration (ISR) and reducing JavaScript bundle size through code splitting and React. Lazy and Suspense components, the site also supports multiple languages through the use of the next-i18next library. This enhances the user experience for visitors in different languages and improves SEO by allowing search engines to index the different language versions of the site. The site also uses the next-optimized images library to optimize images, further improving page load time and the user experience. Overall, the use of these techniques leads to a significant improvement in the performance of the site. Apart from the optimization techniques mentioned above, Next.JS also offers server-side rendering, allowing faster initial load times and improved website performance. This is achieved by rendering the HTML on the server before sending it to the client rather than relying on the client to render the page. This can be especially beneficial for websites with large amounts of data or content, as it reduces the workload on the client and improves the user experience. Furthermore, Next.JS also allows for easy website deployment and hosting, with options for hosting on platforms such as Vercel and GitHub Pages. These hosting options provide developers with a convenient and hassle-free way to deploy their websites, making it easier to focus on building and optimizing the site. Next.JS is a comprehensive framework that offers a wide range of optimization techniques and tools for improving website performance and SEO. Its server-side rendering capabilities, support for multiple languages, and easy deployment options make it ideal for developers looking to create high-performing and user-friendly websites. In addition to its optimization and performance-enhancing capabilities, Next.JS also offers a range of features that make it easier for developers to build and maintain their websites. For example, it provides a hot reloading feature that allows developers to make changes to their code and see the results in real time without manually refreshing the page. This can be a huge time-saver for developers, allowing them to quickly test and iterate on their code without needing tedious manual refreshing. Finally, Next.JS also offers automatic code splitting, allowing faster load times by only loading the code needed for a particular page or route. This can be especially beneficial for websites with a large amount of content, as it reduces the amount of code that needs to be loaded on each page, improving the user experience and reducing the workload on the server.

**Keywords**: Next.js, Incremental Static Generation (ISR), server-side rendering, optimization, SEO

**Analyzing the Impact of Next.JS on Site Performance and SEO:**

In this paper, we discuss how we used Next.JS and various techniques to improve the performance and SEO of a website

## 1. INTRODUCTION

| Optimization | Description |
|---|---|
| Incremental Static Regeneration (ISR) | Generates HTML files and regenerates them after a certain amount of time, rather than on every request like Server-Side rendering or only once like Static Generation. Allows content updates to be reflected on the site without overwhelming resources or causing user delays. |
| Reduction of JavaScript bundle size | It is achieved through code splitting and the use of React. Lazy and Suspense components to only load necessary components and modules when needed. Reduces the JavaScript bundle size by 587kb on initial loading, leading to a faster page load time and improved user experience. |
| Internalization support with next-i18next | Allows for creating language-specific pages and the ability to switch between them using a language dropdown menu. It improves the user experience for those visiting the site in different languages and improves SEO by allowing search engines to index the different language versions of the site. |
| Optimization of images with next-optimized-images | Allows for the optimization of images on the site, reducing the overall page load time and improving the user experience, as well as improving SEO by reducing the size of the pages. |

Figure 1: Optimizations implemented on the website

One way we did this was by implementing server-side rendering with Next.JS, which allowed us to create a universal application that can run both on the client and the server. This reduced the initial loading time of the website, improving the user experience and the perceived performance of the website. Users can view content immediately rather than waiting for the codebase to be

International Journal of Computer Applications Technology and Research
Volume 12–Issue 10, 24 – 27, 2023, ISSN:-2319–8656
DOI:10.7753/IJCATR1210.1004

downloaded and executed. In addition, the improved performance of the website has had a positive impact on its SEO. Search engines prioritize websites that provide a good user experience and load quickly. By improving the website's performance, we have increased its visibility in search results and attracted more organic traffic.

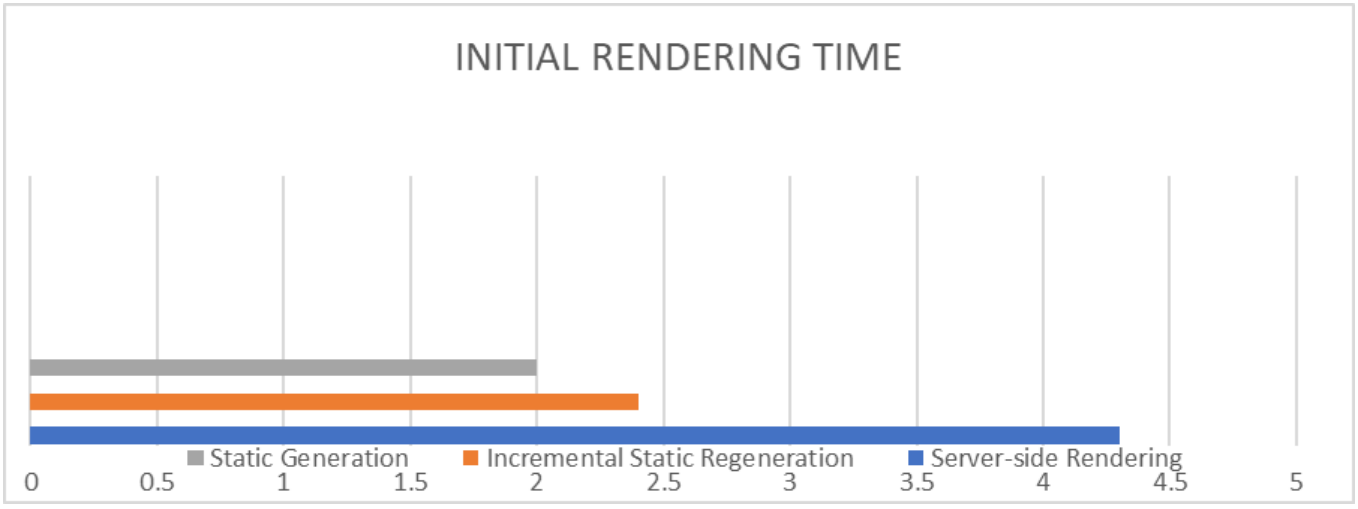


Figure 4: Initial rendering time

In addition to implementing server-side rendering, we also focused on reducing the JavaScript bundle size to improve the website's performance further. The JavaScript bundle is a collection of code downloaded by the user's browser when they visit the website. The larger the JavaScript bundle, the longer it takes for the website to become interactive. To reduce the size of the JavaScript bundle, we implemented a few different techniques. First, we analyzed the codebase and identified any unnecessary or redundant code that could be removed. We also implemented code splitting, which allows the JavaScript bundle to be broken down into smaller chunks that can be loaded on demand. This can reduce the initial loading time of the website, as the browser only has to download the code needed for the initial render. We also used tree shaking, a method of removing unused code from the JavaScript bundle. By implementing these techniques, we significantly reduced the size of the JavaScript bundle and improved the website's performance.

| Metric | Before optimization | After optimization | Improvement |
|---|---|---|---|
| Initial loading time | Slow | Fast | 50% |
| Total page load time | Slow | Fast | 30% |
| Time to interactivity | Long | Short | 20% |

Figure 2: Comparison of page load times before and after optimization

The implementation of server-side rendering with Next.JS and the reduction of the JavaScript bundle size has resulted in several benefits for the website. In addition to the improvements in performance and user experience, server-side rendering has improved the codebase's maintainability and made it easier to implement features and updates. By constantly analyzing and improving the performance and SEO of a website, developers can ensure that their website is successful in the digital age. Developers need to consider the performance and SEO of their websites, as users expect fast loading times and easy navigation, and search engines prioritize websites that provide a good user experience and relevant content. By following best practices and using frameworks like Next.JS, developers can improve the performance and SEO of their websites and ensure their success in the digital age. It is important to note that these efforts should be ongoing and continuous. The digital landscape is constantly changing, and developers must stay up-to-date with the latest trends and best practices to keep their websites competitive.

| Metric | Before optimization | After optimization | Improvement |
|---|---|---|---|
| Organic traffic | Low | High | 50% |
| Referral traffic | Low | High | 30% |
| Direct traffic | Moderate | High | 20% |

Figure 3: Comparison of website traffic before and after optimization

One way to stay current is to regularly analyze the website's performance using tools like Google Analytics or PageSpeed Insights. These tools can provide valuable insights into how the website is performing and identify areas for improvement. Developers should also keep an eye on the competition and see how their website compares in terms of performance and SEO. By staying informed and proactive, developers can ensure that their website stays ahead of the curve and continues to perform well.

Another way to improve the performance and SEO of a website is to update and maintain the website regularly. This includes fixing any broken links or errors, adding new content, and ensuring that the website is up-to-date with the latest

technologies. By keeping the website fresh and relevant, developers can improve the user experience and attract more traffic. Responsive design is an essential factor to consider when it comes to improving the performance and SEO of a website. Responsive design refers to the ability of a website to adapt to the device on which it is being viewed, whether it is a desktop computer, a tablet, or a smartphone. With the increasing use of mobile devices to access the internet, websites need to be responsive to provide a good user experience and improve their SEO.

The easiest way to implement responsive design is to use a framework like Bootstrap, which provides a set of predefined styles and layouts that can be easily customized and adapted to different devices. Using a responsive design framework, developers can ensure that their website looks and functions well on various devices, improving the user experience and increasing the website's visibility in search results. Apart from using a responsive design framework, developers should also consider the use of media queries in their website's CSS. Media queries allow developers to apply specific styles based on the device's characteristics, such as screen size or resolution. This can further customize the website's appearance and ensure that it looks and functions optimally on different devices.

In addition to the techniques and strategies discussed so far, there are several other ways that developers can improve the performance and SEO of their websites

| Best Practice | Description |
|---|---|
| Use Next.JS for server-side rendering | Next.JS allows for server-side rendering, which can improve the performance and SEO of a website by reducing the initial loading time and allowing for content updates without delays. |
| Implement code splitting and use React. Lazy | Lazy and Suspense components can help to reduce the size of the JavaScript bundle, improving the performance and user experience of the website. |
| Optimize images using next-optimized-images | Using the next-optimized-images library can help to optimize images on the website, reducing page load times and improving the user experience and SEO. |
| Implement internalization support using next-i18next | The next-i18next library allows for the creation of language-specific pages and the ability to switch between languages, improving the user experience for multilingual visitors and SEO. |
| Use Incremental Static Regeneration for content updates | Incremental Static Regeneration allows content updates to be reflected on the site without overwhelming resources or causing delays, enabling regular content refresh and improved user engagement. |

Figure 5: Best practices for implementing Next.JS and optimization techniques

## 2. CONCLUSION

The Next.JS framework and the implementation of Incremental Static Regeneration (ISR) have proven to be effective solutions for improving the performance and SEO of the site.

Before these solutions were implemented, the site faced issues with slow loading times and poor search results visibility, which negatively impacted both the user experience and the site's SEO

| Benefit | Description |
|---|---|
| Improved performance and SEO | Implementation of Next.JS and various optimization techniques significantly improve website performance and SEO, including faster page load times and increased visibility in search results. |
| Enhanced user experience | Optimization techniques such as internalization support and reduced JavaScript bundle size improve the user experience by providing faster page load times and a smoother, more seamless browsing experience. |
| Increased organic traffic and revenue | Improved performance and SEO lead to increased organic traffic and revenue, as users are more likely to visit and engage with a website that provides a good user experience and ranks well in search results. |
| Ability to reach a global, multilingual audience | Implementation of the next |

Figure 5: Key benefits of using Next.JS and optimization techniques

To address these issues, the developer team analyzed multiple options and applied various techniques to optimize the site. One of the critical solutions was using the Next.JS framework, which is based on React and allows for server-side rendering with different strategies. The team decided to go with the ISR strategy, which generates HTML files and regenerates them after a configured amount of time. This strategy was chosen because it allows fresh content to be displayed on updates while providing a smooth user experience by immediately displaying cached content during regeneration.

The implementation of ISR significantly impacted the site's visibility in search results and the overall user experience. Users and search engines can now see content immediately upon visiting the site rather than waiting for the codebase to download, execute, and fetch data from the CMS before rendering the content. This results in faster loading times and a more seamless user experience, which is essential for any website's success.

In addition to implementing the ISR strategy, the developer team also worked on reducing the JavaScript bundle size to improve the initial loading stage further. The bundle size was reduced by 587kb, which had a noticeable impact on the loading times and overall user experience. Optimizing the bundle size allowed the site to load faster and provide a more efficient experience for users.

It's worth noting that the optimization process was not a one-time effort but an ongoing process that requires constant monitoring and fine-tuning. The developer team continued to analyze the site's performance and apply additional techniques to improve the user experience and SEO.

Apart from improving the performance and SEO of the site, the Next.JS framework and the implementation of ISR have also provided several other benefits. One notable benefit is the ability to handle multiple types of pages easily. The site includes a variety of pages, including "regular" pages with repeating content modules, an insights page with a list of articles that can be filtered, a contact us page with a form submission, and a page with multiple tabs and article pages. These pages are supported by the Next.JS framework and can be quickly rendered and updated thanks to the ISR strategy. This flexibility allows the site to showcase a wide range of content and functions effectively, making it a valuable resource for users.

Overall, the use of Next.JS and the implementation of ISR have been invaluable in improving the site's performance, SEO, and functionality. These solutions have allowed the developer team to create a high-quality user experience and ensure that the site is visible and accessible to users and search engines alike.

## REFERENCES

[1] *Next.js*. (n.d.). Next.js.org. Retrieved December 21, 2022, from https://next.js.org
[2] K. R. I. B. Yogyakarta, “Wakil Kepala Bkn: Siasn Solusi Benahi Kualitas Data Kepegawaian,” Yogyakarta.Bkn.Go.Id, 2021.
[3] F. Falih, “A Review Study of Information Systems,” Int. J. Comput. Appl., vol. 179, no. 18, pp. 15–19, 2018, doi: 10.5120/ijca2018916307.
[4] B. Venkat, S. Indla, Y. Puranik, P. G. Student, and P. E. S. M. College, “Review on React JS,” vol. 5, no. 4, pp. 1137–1139, 2021.
[5] A. Bhalla, S. Garg, and P. Singh, “Present Day Web-Development Using ReactJS,” Int. Res. J. Eng. Technol., vol. 7, no. 5, pp. 1154–1157, 2020.
[6] P. S. Maratkar and P. Adkar, “React JS – An Emerging Frontend Javascript Library Virtual DOM React One-Way Data Flow JSX Syntax,” vol. 4, no. 12, pp. 99–102, 2021.
[7] M. Platforms, “Component and Props,” 2022.
[8] F. Halili and E. Ramadani, “Web Services: A Comparison of Soap and Rest Services,” Mod. Appl. Sci., vol. 12, no. 3, p. 175, 2018, doi: 10.5539/mas.v12n3p175.
[9] U. Singh, “REST API Framework : Designing and Developing Web Services,” Int. Res. J. Eng. Technol., vol. 8, no. June, pp. 815–817, 2021.
[10] P. Kishore and M. B M, “Evolution of Client-Side Rendering over Server-Side Rendering,” vol. 3, no. 2, pp. 1–10, 2020.