

Comparative Study of React, Astro and Eleventy Technologies in Developing a Lottery Statistical Analysis Application

Daiana-Larisa Lucaciu 

Faculty of Electrical Engineering and Information Technology, University of Oradea, Oradea, Romania

Email: lucaciu.daianalarisa@student.uoradea.ro

How to cite this paper: Lucaciu, D.-L. (2026) Comparative Study of React, Astro and Eleventy Technologies in Developing a Lottery Statistical Analysis Application. *Open Journal of Applied Sciences*, 16, 758-769. <https://doi.org/10.4236/ojapps.2026.163046>

Received: February 16, 2026

Accepted: March 3, 2026

Published: March 6, 2026

Copyright © 2026 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0). <http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

This paper compares React, Astro and Eleventy technologies by developing a web application for analyzing Loto 6/49 and Joker draws. The application includes displaying results, statistics and numerical frequencies. The study evaluates the performance, SEO optimization, ease of development and scalability of each framework. The results show that Astro offers an optimal balance between performance and flexibility, React stands out for its interactivity, and Eleventy for its simplicity and speed. The conclusions emphasize the importance of choosing the technology according to the specific requirements of the application.

Keywords

React, Astro, Eleventy, Lotto Application, Statistical Analysis, Web Frameworks, 6/49 Draw, Static Site Generator

1. Introduction

In the context of the accelerated evolution of web technologies, developers are faced with a multitude of options for implementing modern web applications [1]. Choosing the right technology influences application performance, scalability, user experience, and search engine optimization (SEO). Each technology involves a series of trade-offs between flexibility, development time, code efficiency, ease of maintenance, and future expansion possibilities [2].

Web application development has evolved significantly from static pages to complex dynamic applications, integrated with external services, APIs, and cloud platforms [3]. In this dynamic, new development paradigms have emerged that aim

to optimize content delivery and interactivity, including component-based architectures, static and dynamic rendering, and hybrid approaches [4].

This paper proposes a comparative approach between three current technologies: React, Astro, and Eleventy. They are analyzed through a practical case study, represented by the development of an application for the visualization and analysis of Loto 6/49 and Joker draws. The application includes functionalities such as displaying recent draws, calculating the frequency of number occurrences, generating visual statistics and an adaptive interface for various devices [5]. By designing the functionalities identically in all three technologies, an objective framework for comparison is created.

The purpose of the research is to provide a comparative and objective evaluation of each technology, based on quantifiable and qualitative indicators: loading times, Lighthouse scores (Performance, SEO, Accessibility, Best Practices), code structure and modularity, level of SEO support and extensibility potential. Through this analysis, the paper aims to support software architecture decisions in web projects oriented towards statistical or dynamic content, providing a clear vision of the advantages and limitations of each analyzed technology.

2. React

React is an open-source JavaScript library developed by Facebook, used for building user interfaces, especially component-based ones. React's wide adoption is due to its flexibility, rich ecosystem, and declarative paradigm that allows for more efficient management of application states [6].

One of the fundamental features of React is Virtual DOM, a mechanism that optimizes updates to the real DOM through an intelligent reconciliation model. This ensures improved performance in interactive applications [7]. In addition, React allows for the construction of unidirectional applications, where the data flow is controlled and predictable.

From the perspective of developing a Lottery analysis application, React offers multiple advantages:

Reusable components: Allows structuring the application into reusable components for extractions, tables, charts, etc.

Mature ecosystem: The library is compatible with a large number of external libraries for graphics (e.g., Chart.js, Recharts), routing (React Router), formatting, and data manipulation.

Support for interactivity: It is ideal for dynamic functionalities, such as filtering extracts or animating graphs.

However, using React in its standard form (**Figure 1**), without Server-Side Rendering (SSR), brings some limitations, especially in the context of SEO [7]. Being a library intended for Single Page Application (SPA) applications, React renders content on the client side, which can affect indexing by search engines. In addition, the size of the JavaScript package and the dependence on its complete loading can negatively affect the initial speed of displaying content.

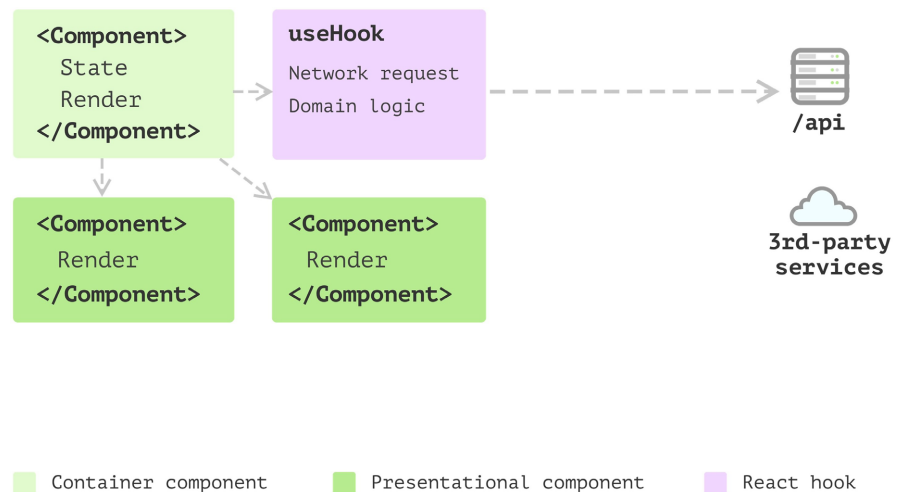


Figure 1. Modularizing react applications with established UI patterns.

In this study, the Loto application made with React was implemented using the classic version (SPA), with Node.js as the development server and custom build scripts. Modular JavaScript was used for better organization of the code. The interface includes a main page with the latest extracts, sections for detailed statistics and interactive graphical visualizations. The functionality is delivered entirely through the browser, and data retrieval was done through external APIs, processed locally. Performance tests and comparative analysis will be presented in the following chapters.

3. Astro

Astro is a recently released modern web framework designed to build fast and efficient websites using a hybrid approach between static site generation (SSG) and controlled interactivity of components. It optimizes performance through a model known as an “islands architecture,” which separates static content from dynamically loaded interactive components.

Astro’s main feature is that it generates HTML content at build time, ensuring the delivery of pre-rendered pages with reduced load times. Interactive components (Figure 2)—called “islands”—are only hydrated when the user interacts with them, which significantly reduces the amount of JavaScript loaded initially and optimizes the overall performance of the page [8].

Astro offers support for multiple integrations with popular front-end frameworks, including React, Vue, and Svelte, allowing developers to combine components built in these technologies with generated static pages [8]. This flexibility allows for easy adaptation to specific project requirements and development team preferences.

By its nature, Astro is highly optimized for SEO, as content is immediately available for search engine indexing, unlike SPA applications that generate content on the client. In addition, improved load time performance contributes to a better experience for end users, which can increase retention and satisfaction.

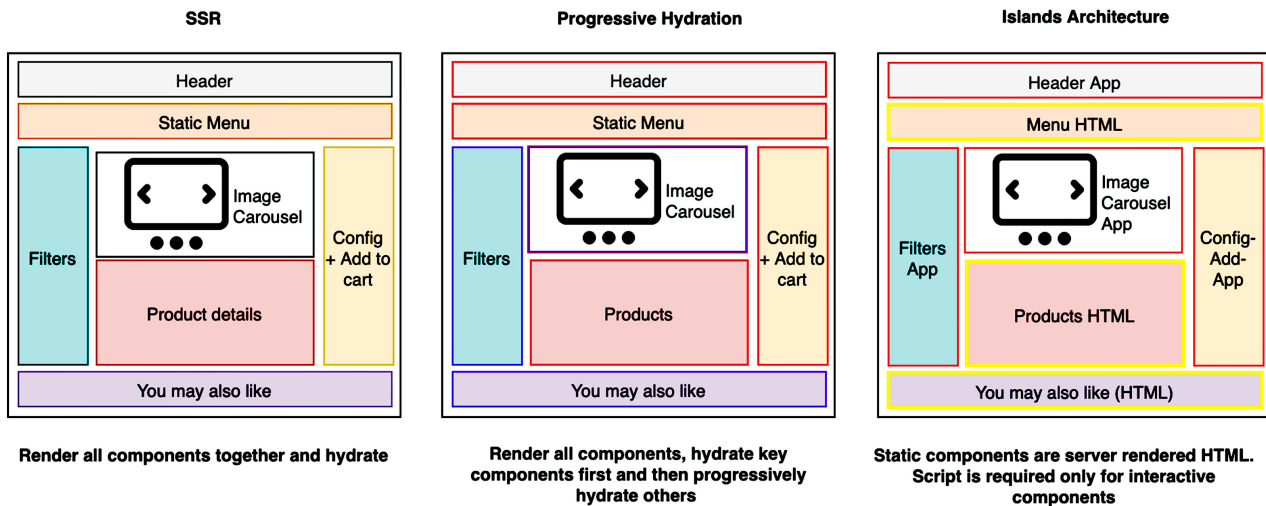


Figure 2. Architectures for Astro—“islands”.

Astro uses Node.js for the build process and supports static site generation with easy integration of external data via APIs, thus facilitating the development of dynamic web applications with a solid static foundation [9].

4. Eleventy

Eleventy is a Static Site Generator (SSG) built on the Node.js platform, which allows the transformation of source files into static HTML pages ready for publication (Figure 3). This approach makes it possible to quickly develop high-performance, scalable and easy-to-maintain websites, without requiring a dynamic backend at runtime [10].

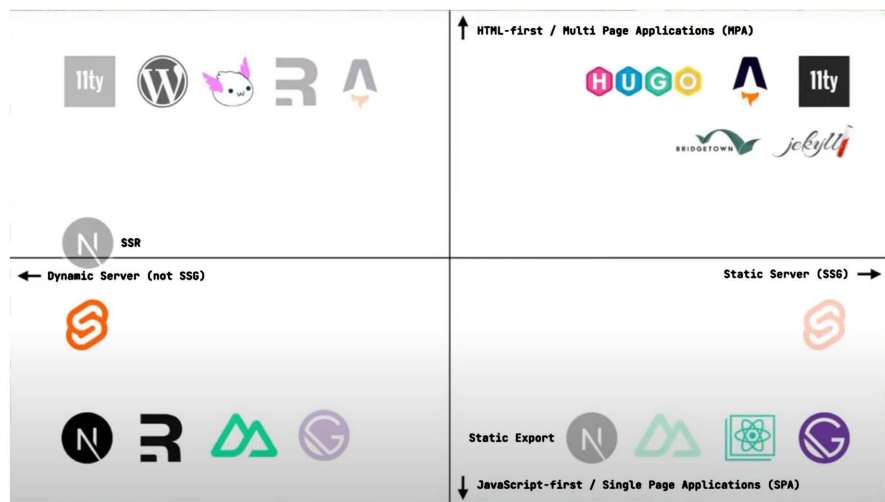


Figure 3. Static Sites—11ty (eleventy).

Static generators have established themselves in recent years as an effective solution for creating websites with stable or periodically updated content, offering significant advantages in terms of loading speed, security and SEO optimization.

Eleventy stands out in this landscape through its simplicity and flexibility, offering support for a wide variety of template languages, such as Markdown, Nunjucks, Liquid, Handlebars and JavaScript templates, which allows adaptation to various types of content and work styles.

The building process in Eleventy is completely local, all pages are generated in advance and served as static files [10]. This method eliminates the need for server-side processing on each request and reduces the reliance on JavaScript in the browser to display the initial content. As a result, sites built with Eleventy benefit from fast loading times and are perfectly indexable by search engines, which contributes to increased organic visibility.

Another major advantage is the structural flexibility offered by Eleventy. There is no rigidly imposed architecture, and developers can organize files and folders according to the specific needs of the project. This freedom makes Eleventy suitable for a wide range of applications, from simple blogs to complex sites with multiple types of content.

Although oriented towards static content, Eleventy also supports the integration of external data in the build process, which makes it capable of generating updated pages based on dynamic data retrieved at compile time [11]. Thus, it is possible to create semi-dynamic sites, where regular updates do not compromise speed and scalability.

On the other hand, Eleventy's major limitation is the lack of an advanced JavaScript runtime for dynamic interactivity in the page. For complex functionality, such as real-time filtering and sorting or other interactive components, it is necessary to manually add client-side scripts, which can complicate development for highly interactive applications.

5. Comparative Analysis

A. Evaluation methodology

In order to perform an objective analysis of the three technologies—React, Astro and Eleventy—the same statistical analysis web application was implemented for the Loto 6/49 and Joker draws. Each variant includes the same functionalities: displaying recent draws, calculating the frequency of number occurrences, generating statistical graphs, adaptive interface and retrieving data from external sources.

The evaluation was carried out through a combination of quantitative and qualitative indicators, using a set of standard tools: Google Lighthouse for performance scores, accessibility, best practices and SEO; PageSpeed Insights for loading times; as well as manual analysis of the code structure, modularization mode and development complexity.

For consistency, the tests were performed under the same conditions: the same browser, network connection and the same local test machine.

B. Application interface and user experience

All three implementations offer a similar interface at a visual and functional level: a main page with the latest extractions, a statistics section that includes fre-

quency charts and dynamic visualizations, as well as a responsive structure for mobile access.

React (Figure 4) offers a fluid and dynamic experience, with fast updates of state-based components. Interactivity is natural, but the initial load time is affected by the amount of JavaScript delivered to the client.

Astro (Figure 5) combines the advantages of server-side rendering with interactive components only where necessary (“islands architecture”). Thus, pages load very quickly, but retain interactivity where it is relevant.

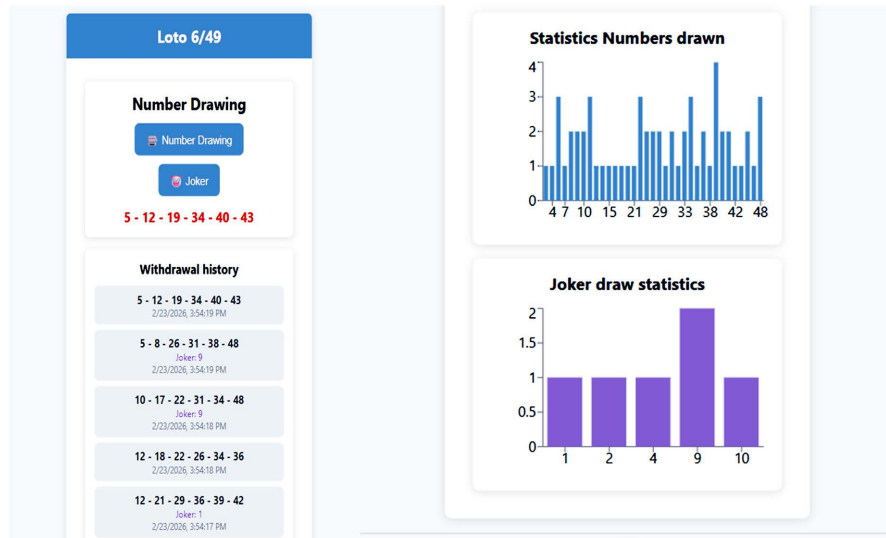


Figure 4. React interface.

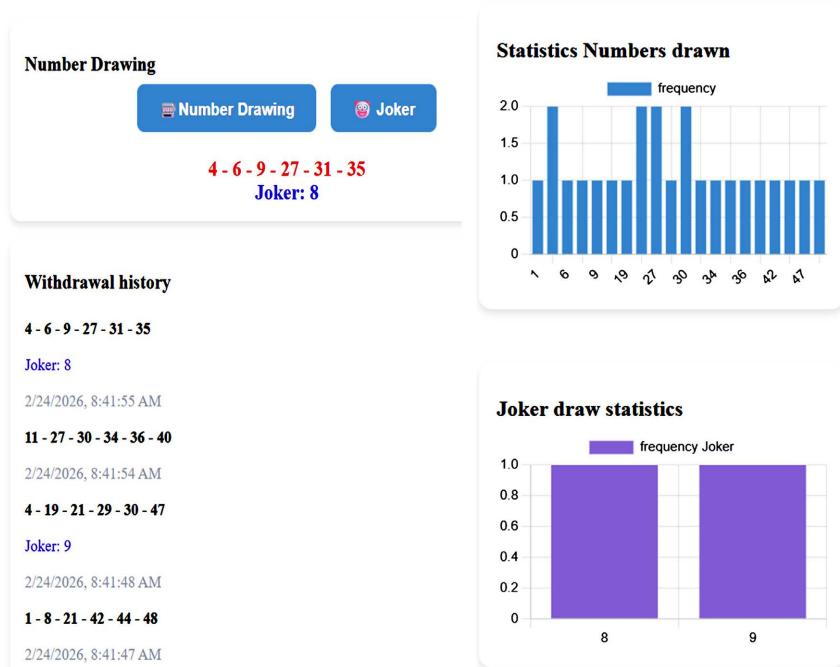


Figure 5. Astro interface.

Eleventy (Figure 6), being a static site generator, generates completely static and fast pages, but with a reduced level of interactivity, requiring additional JavaScript code to achieve the same functionality.

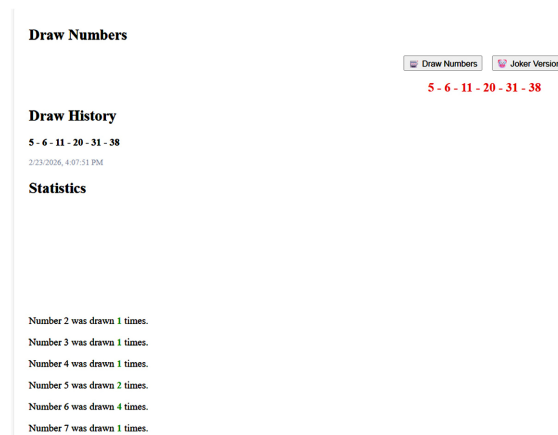


Figure 6. Eleventy interface.

C. Detailed comparative analysis

In the context of developing modern web applications, choosing the right technology is a complex process that involves careful evaluation of several technical and user experience criteria. For the statistical analysis application of Loto 6/49 and Joker draws, three representative technologies from the current ecosystem were compared: React, Astro and Eleventy. This comparison was made based on key indicators provided by the Google Lighthouse tool, which measures performance, accessibility, compliance with best practices and search engine optimization (SEO).

Although the application is a single-page application (SPA) built with React without server-side rendering (SSR), SEO is high because the application uses pre-rendering for key pages. During the build process, static HTML snapshots are generated with all essential content and meta tags, which Lighthouse can index. Additionally, appropriate meta descriptions, titles, and Open Graph tags are included for all pages, and dynamic rendering ensures that search engines can access the page content correctly. This approach allows a Lighthouse SEO (Figure 7) score of 100 despite the SPA architecture [12].

| Category | React (%) | Astro (%) | Eleventy (%) |
|----------------|-----------|-----------|--------------|
| Performance | 68 | 83 | 97 |
| Accessibility | 90 | 89 | 100 |
| Best Practices | 100 | 80 | 96 |
| SEO | 100 | 89 | 90 |

Figure 7. Comparative analysis.

D. Performance

Performance is a crucial indicator for any web application, influencing both the user experience and the conversion rate or loyalty. The scores obtained reflect how quickly the page can load and become interactive, in real-world usage conditions.

Eleventy achieves a remarkable score of 97%, demonstrating the static generator's ability to produce completely pre-rendered pages during the build phase. This eliminates the need to execute intensive JavaScript in the browser, which translates into extremely short load times and an optimal experience on devices with limited resources. This aspect is essential for mobile users or users with slow connections, where every millisecond counts.

Astro, with a score of 83%, adopts a hybrid strategy through its "islands architecture". This involves generating static pages, but loading interactive components only where they are absolutely necessary. This approach optimizes the initial display time and allows for maintaining dynamic functionalities where the user needs them, without sacrificing overall speed.

React, with its score of 68%, reflects the inherent constraints of a classic Single Page Application (SPA), where a significant amount of JavaScript must be loaded and executed at the client level. This negatively impacts load times and can lead to a slower user experience, especially on mobile devices or unstable connections. To mitigate these issues, React offers solutions such as Server-Side Rendering (SSR) or Static Site Generation (SSG), but the current implementation uses the standard SPA.

E. Accessibility

Accessibility ensures that the application is usable by all users, including those with disabilities, which is not only an ethical requirement but also a legal obligation in many jurisdictions.

Eleventy stands out with a perfect score of 100%, indicating rigorous adherence to accessibility standards, including the correct use of semantic HTML elements and the avoidance of practices that can negatively affect assisted navigation.

React achieves a very good score of 90%, showing that while it is a flexible and powerful platform, implementing accessibility depends largely on how the components are developed. This allows users to create highly accessible interfaces, but this requires additional attention in the design and testing of the components.

Astro achieves 89%, a good score but which suggests that some aspects of accessibility can be improved, possibly due to the complexity of managing the mix of static and interactive content. For example, asynchronous loading of interactive components may require special measures to maintain a consistent experience for users with assistive technologies.

F. Best practices

Following best practices involves the correct use of web technologies, enhanced security, optimal resource management, and compliance with modern standards, all of which contribute to the stability and ease of maintenance of the application in the long term.

React scores a maximum of 100%, thanks to its mature ecosystem and numerous tools and guides that facilitate development according to best practices. Its extensive support and active community provide a robust framework for secure and high-performance implementation.

Eleventy scores 96%, confirming that, although it is a relatively simple static site generator, it provides a solid and secure platform for building fast and well-architected sites.

Astro, with 80%, indicates potential for improvement in this area, probably due to the new hybrid paradigm and the lower maturity of the ecosystem. Security and resource management optimizations can be further developed.

G. SEO Optimization

Search engine optimization is essential for the visibility and accessibility of content in the online environment.

React achieves a perfect score of 100%, which may seem surprising for a SPA application, but this is possible thanks to additional strategies implemented, such as partial pre-rendering or the integration of dedicated SEO solutions, which allow for efficient indexing.

Eleventy, with 90%, naturally benefits from its static nature, which facilitates crawler access and fast indexing of content without additional dependencies.

Astro has a score of 89%, being very close to Eleventy, but the partial loading of components on the client can introduce some challenges related to indexing dynamic content.

H. Code complexity

The complexity of the associated code is an essential factor in evaluating web technologies, as it directly influences the maintainability, scalability of the application, and speed of development. In this study, complexity was assessed qualitatively, taking into account modularity, readability, the amount of boilerplate code, and the learning curve associated with each technology.

React enforces a component-based architecture, which favors a modular and reusable code structure. This increases maintainability and scalability, but generates a larger volume of boilerplate code, especially in state management and routing, which can extend the initial development time. The JSX syntax, which combines the structure of HTML with JavaScript, provides flexibility, but can increase cognitive growth for inexperienced developers. Also, frequent use of external libraries adds an additional level of complexity (Figure 8).

| Framework | Components/Pages | Routing | State | Code Size | Files |
|-----------|------------------|-----------------|---------------------|------------|-------|
| React | 12 | React Router v6 | Local + Context API | ~1,250 LOC | 18 |
| Astro | 4 | Static | N/A | ~300 LOC | 4 |
| Eleventy | 6 | Static | N/A | ~150 LOC | 6 |

Figure 8. Complexity of frameworks.

Astro adopts a hybrid approach, combining static generation with selective hy-

dration of interactive components (“island architecture”). This leads to a simpler code compared to React, static content and dynamic widgets coexist naturally. Astro allows the use of multiple frontend frameworks (React, Vue, Svelte), which can introduce diversity into the code, but also increased flexibility. Overall, the code complexity in Astro is moderate: lower than in React, but higher than in traditional static generators, due to its hybrid nature.

Eleventy, as a static website generator, stands out for its simple code structure. It uses simple templating languages (Nunjucks, Liquid) and minimal JavaScript, focusing on page generation during build. Thus, the complexity of the rules is very low, and the code base is small and easy to understand. However, the lack of native support for advanced interactivity requires the development of custom solutions or the integration of client-side frameworks, which can increase complexity in large projects.

6. Conclusions

This research aimed to provide a detailed comparative analysis of three modern web technologies—React, Astro, and Eleventy—by implementing a practical application for statistical analysis of Loto 6/49 and Joker draws. Through a rigorous evaluation of performance, code complexity, SEO support, and user experience, the distinctive features and potential limitations of each technology were highlighted in the context of a dynamic web project with specific visualization and interactivity requirements.

First, React, as a mature and widely adopted library, provides a robust model for developing complex interactive applications, thanks to its component-based architecture and unidirectional data flow. This structure facilitates modularity, code reuse, and clear state management, essential elements in applications with dynamic interfaces and frequent data updates. However, the analysis shows that React has a higher cost in terms of JavaScript bundle size, which negatively impacts initial load times and thus the user experience on mobile devices or low-bandwidth connections. Furthermore, client-side rendering alone, in the absence of Server-Side Rendering (SSR) mechanisms, can compromise SEO indexing, a critical aspect for applications with dynamic content accessible to the general public. Thus, React is recommended in scenarios where advanced interactivity is a priority, and additional optimizations for performance and SEO can be integrated.

On the other hand, Astro proposes an innovative paradigm, known as the “island architecture”, which effectively combines static content generation with limited hydration of interactive components, thus reducing JavaScript overhead on the client. This strategy leads to significantly shorter load times, also reflected in the competitive performance and SEO scores obtained in the tests. Astro’s flexibility to integrate multiple front-end frameworks (React, Vue, Svelte) allows developers to take advantage of their ecosystems without compromising overall site speed. However, this hybrid complexity can introduce additional challenges to the development and maintenance process, requiring a deep understanding of how

server-side versus client-side rendering is done. Astro is therefore recommended for projects that seek a balance between performance, interactivity, and SEO optimization, especially in the context of applications that require a fast and well-indexed web presence.

Eleventy, on the other hand, is a minimalist and efficient solution, based on completely static page generation during the build process. Its architectural simplicity is a major advantage in terms of code complexity, offering a low learning curve and easy maintenance. This translates into very fast loading times and optimal performance in Lighthouse tests, where Eleventy stood out with maximum scores for performance and SEO. However, the lack of native support for complex dynamic functionality requires additional development of client-side scripts or integration with other frameworks to achieve advanced levels of interactivity. Thus, Eleventy is an excellent choice for projects oriented towards predefined content, with high speed and SEO requirements, but with limited interactivity.

In conclusion, choosing the optimal technology for developing a web application must take into account the nature and complexity of the required functionalities, the performance and SEO goals, as well as the available development resources. React is recommended if the application requires advanced interactivity and granular control over the behavior of components, but must be carefully optimized so as not to compromise performance. Astro offers a balanced solution, suitable for applications that need both performance and moderate interactivity, while Eleventy is mainly aimed at static projects, where speed and SEO are a priority, and interactivity can be limited or added selectively.

This study emphasizes the importance of a rigorous and contextualized analysis before choosing web technologies for specific projects, highlighting that there is no universal solution, but an informed decision based on the particular requirements and constraints of the application. As future directions, it is recommended to investigate the integration of additional optimization mechanisms for React and Astro, as well as explore the potential of Eleventy in combination with client-side frameworks for extending interactivity.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

- [1] Smith, J.F. (2021) *Practical Entity Framework Core 6: Database Access for Enterprise Applications*. Apress.
- [2] Google Developers (2025) *Web.dev—Tools and Guidance for Modern Web Development*. <https://web.dev/>
- [3] Ahmed, R. (2013) *The Web Book—Build Static and Dynamic Websites: A Beginner's Step-by-Step Guide to Creating Static and Dynamic Websites*, 1st ed. Apress.
- [4] ReactJS.org (2025) *React—A JavaScript Library for Building User Interfaces*. <https://reactjs.org/>
- [5] Birlea, M., Muntean, D. and Pop, M. (2023) *A Comparative Study on Web Frame-*

-
- works for Front-End Development. *2023 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, Cluj-Napoca, 16-18 May 2024, 1-6.
- [6] Schwarzmüller, M. (2022) *React Key Concepts: Consolidate Your Knowledge of React's Core Features*. Packt Publishing.
 - [7] Fowler, M. (2025) *Modularizing React Apps for Scale*. <https://martinfowler.com/articles/modularizing-react-apps.html>
 - [8] Hygraph Blog (2025) *Astro JavaScript Framework Overview*. <https://hygraph.com/blog/astro-javascript>
 - [9] Astro.build (2025) *Astro—The Modern Static Site Builder*. <https://astro.build/>
 - [10] Leatherman, Z. (2025) *Eleventy—A Simpler Static Site Generator*. <https://www.11ty.dev/>
 - [11] The New Stack (2025) *Static Sites Do Scale: Eleventy vs. Next.js at 11ty Event*. <https://thenewstack.io/static-sites-do-scale-eleventy-vs-next-js-at-11ty-event/>
 - [12] Lighthouse (2025) *Lighthouse: An Open-Source, Automated Tool for Improving the Quality of Web Pages*. <https://developers.google.com/web/tools/lighthouse>

Data Availability Statement

All lottery results for Loto 6/49 and Joker are generated automatically by the application code, which simulates draws, records history, and computes number frequency statistics. The full source code is publicly available on GitHub at <https://github.com/daia-code/Loto>.

By accessing this repository, readers can run the code to reproduce the draws, generate updated statistics, and verify all analyses presented in this work.

No manual data entry is required, ensuring complete reproducibility.