Technical sciences

UDC 004.738.5:004.942

Movchan Bohdan
Student of the
Institute of Special Communication and Information Protection
National Technical University of Ukraine
«Igor Sikorsky Kyiv Polytechnic Institute»
ORCID: 0009-0007-5470-4934

Onishchenko Volodymyr
Lecturer of the
Institute of Special Communication and Information Protection
National Technical University of Ukraine
«Igor Sikorsky Kyiv Polytechnic Institute»
ORCID: 0009-0000-1355-9178

Shevchuk Olha
PhD, Lecturer of the
Institute of Special Communication and Information Protection
National Technical University of Ukraine
«Igor Sikorsky Kyiv Polytechnic Institute»
ORCID: 0000-0002-2866-439X

DOI: https://doi.org/10.25313/2520-2057-2026-6-12084

MODELING OF WEB APPLICATION SPEED BASED ON WEB PAGE PERFORMANCE METRICS

Summary. *The article examines the main factors affecting the performance of modern web applications and analyzes metrics for evaluating web page performance. The indicators TTFB, FCP, LCP, INP, and CLS are considered, which characterize page loading speed, interface stability, and the efficiency of user interaction with a web application. A mathematical model for evaluating web application performance is proposed, formalizing the relationship between the parameters of the client-side and server-side parts of the system and the performance indicators of a web page. Within the framework of the model, the main input parameters are identified, including the number of HTTP requests, the size of JavaScript and CSS resources, network connection speed, caching parameters, and interface rendering characteristics. For comprehensive performance evaluation, an integral performance indicator using weighting coefficients of performance metrics is introduced. Simulation-based modeling of the influence of system parameters on web application performance was carried out. A series of numerical experiments was conducted to investigate the dependence of the integral performance indicator on the main system parameters, and corresponding graphical dependencies were constructed.*

Key words: *web application, web application performance, web page performance, mathematical model, integral performance indicator, Core Web Vitals.*

Introduction. In the current context of web technology development, the performance of web applications has become one of the key factors determining the effective functioning of digital services and the quality of user interaction with the system. The increasing complexity of client-side interfaces, the spread of single-page applications, the growing volume of JavaScript code, and the number of network requests lead to higher system load, which negatively affects the performance indicators of a web resource [1; 2].

The issue of performance becomes particularly important in the context of high competition among digital platforms, where the performance of a web application directly influences user behavioral factors, conversion rates, and the ranking positions of web resources in search engines. In this regard, ensuring a high response speed of web pages is an important component of modern web development and software system optimization [3; 4]. In addition, the impact of various factors, such as the volume of JavaScript resources, the number of HTTP requests, caching parameters, server response time, and interface rendering characteristics, is complex in nature and requires a formalized description [5; 6]. In this context, the development of a mathematical model for evaluating the performance of web applications is a relevant task, as it will make it possible to formalize the relationship between the input parameters of the system and the performance indicators of a web page [7].

Analysis of previous studies. The issue of ensuring high performance of web applications is one of the key areas of modern research in the field of web development and software engineering. The active development of client-side web technologies, the emergence of single-page applications, and the increasing volume of dynamic content have led to a significant rise in performance requirements for web systems. In scientific works and technical literature, considerable attention is paid to studying the impact of page loading speed on user experience, behavioral factors, and the efficiency of digital platforms. Research findings indicate that even a slight increase in web page loading time can negatively affect the level of user interaction with a service and reduce conversion rates [8; 9].

One important area of research has been the development of systems for evaluating the performance of web applications based on standardized metrics. Google proposed the Core Web Vitals concept, which combines key indicators for assessing web page performance, in particular Largest Contentful Paint (LCP), Interaction to Next Paint (INP), and Cumulative Layout Shift (CLS). The use of these metrics has made it possible to move from a subjective assessment of performance to a formalized analysis of web resource productivity. A significant number of modern studies are devoted to analyzing the influence of JavaScript resources, interface rendering, the number of HTTP requests, and network connection parameters on Core Web Vitals values [10; 11].

At the same time, a substantial part of scientific research focuses on individual aspects of web application optimization, including resource minification and compression, the use of caching, optimization of server response, and the application of modern interface rendering mechanisms. In particular, studies devoted to frontend optimization have shown that an excessive amount of JavaScript code and an inefficient DOM tree structure can significantly increase the processing and rendering time of a web page in the user's browser [11; 12]. Research related to the use of automated performance analysis tools, such as Lighthouse and WebPageTest, is also of particular scientific interest, as these tools enable comprehensive evaluation of web resource performance under different load conditions [10; 13].

Despite the significant number of studies in the field of web application optimization, the issue of developing a generalized mathematical model for performance evaluation that would take into account the relationship between the main system parameters and a set of performance metrics has not been sufficiently addressed. Most existing approaches focus on the analysis of individual indicators or practical optimization recommendations, whereas the tasks of formalizing the dependencies between system input parameters and integrated performance indicators remain relevant. This highlights the need to develop a mathematical model for evaluating web application performance capable of providing a comprehensive analysis of the influence of individual factors on web page response speed [12; 14].

Aim of the article. The purpose of the article is to develop a mathematical model for evaluating the performance of web applications that makes it possible to formalize the relationship between the parameters of the client-side and server-side components of the system and the performance indicators of a web page. The proposed approach is aimed at analyzing the influence of key factors, in particular server response time, resource volume, the number of network requests, and interface rendering parameters, on the loading speed and user interaction with a web application.

Presentation of the main material. One of the key performance indicators of a web application is the response speed of a web page, which characterizes the time between a user sending a request and the complete rendering of the required content in the browser. High response time values lead to increased user waiting time, deterioration of system interactivity, and reduced efficiency of the web resource. In addition, web page performance directly affects the ranking of web resources in search engines, since modern SEO algorithms take performance indicators into account when assessing website quality [3].

Specialized performance metrics are used to evaluate the speed of web applications, making it possible to analyze individual stages of web page loading and rendering. One of the basic metrics is Time to First Byte, which determines the time between sending an HTTP request and receiving the first byte of the response from the server. This indicator characterizes the request processing speed of the server-side component of the system and depends on server performance, database efficiency, network connection parameters, and the use of caching mechanisms [4].

The First Contentful Paint indicator is used to evaluate the time required for the first content element to be displayed in the user's browser. The value of this metric determines the moment when the user receives the first visual response from the web application. Reducing FCP time positively affects the perceived performance of the system and improves the user experience.

One of the most important modern performance metrics is Largest Contentful Paint (LCP), which characterizes the time required to render the largest content element within the visible area of a page. This indicator is used to evaluate the speed of the main page loading process and is one of the key parameters of Core Web Vitals [5]. LCP is influenced by the volume of resources, data transfer speed, image optimization, the efficiency of JavaScript code execution, and interface rendering parameters.

The Interaction to Next Paint metric is used to evaluate the speed of the interface response to user interaction. The value of this indicator is determined by the delay between a user action and the moment the interface is updated. High INP values indicate an overloaded browser main thread, an excessive number of JavaScript operations, or inefficient event handling organization [6].

The Cumulative Layout Shift (CLS) indicator characterizes the stability of the page structure during loading. Significant shifts of interface elements negatively affect user interaction with the system and may lead to erroneous actions. Reducing the CLS value is an important condition for ensuring the stability of the web application interface.

An analysis of the above metrics shows that web application performance is formed under the influence of a significant number of interrelated factors. The use of individual indicators does not make it possible to comprehensively evaluate system performance or determine the degree of influence of specific parameters on web page performance. In this regard, there is a need to develop a mathematical model for evaluating web application performance that will provide a formalized description of the relationship between system parameters and performance indicators.

Within the framework of this study, a web application is considered as a system whose state is determined by a set of input parameters (1):

$$X = \{x_1, x_2, \dots, x_n\} \tag{1}$$

where x_i denotes the parameters that affect the performance of a web application. The main input parameters of the model include: x_1 — the number of HTTP requests; x_2— the volume of JavaScript resources; x_3— the volume of

CSS resources; x_4— the network connection speed; x_5 — caching parameters; x_6 — the size of the DOM structure; x_7 — JavaScript execution time; x_8 — interface rendering parameters; and x_9 — server response time.

Each of the aforementioned parameters directly affects specific stages of web page loading and processing. For instance, an increase in the number of HTTP requests leads to higher network latency and a longer resource transfer time. A substantial volume of JavaScript code increases the load on the browser and may result in blocking of the main execution thread. A large DOM structure complicates the page rendering process and negatively affects the speed of interface updates.

The output parameters of the model are the main performance metrics of a web page. The relationship between the input and output parameters of the model is described by a system of functional dependencies (2):

$$Y = \begin{bmatrix} TTFB \\ FCP \\ LCP \\ INP \\ CLS \end{bmatrix} = F\left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{bmatrix}\right), \qquad (2)$$

where F(X) is the function for evaluating the performance of a web application, which formalizes the relationship between the system parameters and the performance indicators of a web page. The model was developed on the basis of a systematic analysis of the factors influencing the performance of web applications. The characteristics of the client-side and server-side components of the system were selected as the input parameters of the model: the number of HTTP requests, the volume of JavaScript and CSS resources, the network connection speed, caching parameters, the size of the DOM structure, JavaScript execution time, interface rendering parameters, and server response time. Standardized web page performance metrics, namely TTFB, FCP, LCP, INP, and

CLS, which are used within the Core Web Vitals assessment framework, were selected as the output parameters.

The function $F(X)$ formalizes the dependence between the system parameters and the performance indicators of a web page. For the comprehensive evaluation of performance, an integral indicator Q was used, formed through the weighted aggregation of individual performance metrics. Within the proposed model, an increase in the value of the integral indicator Q is interpreted as a deterioration in web application performance, since it reflects the cumulative impact of delays and performance losses.

In contrast to simplified performance evaluation models, the proposed approach takes into account the multifactorial influence of system parameters on individual web application performance metrics. Each metric is formed under the influence of a set of parameters of the client-side and server-side components of the system, which makes it possible to comprehensively describe the process of web page performance formation. Within the model, the integral performance indicator of a web application is defined by the following expression:

$$Q = w_1 f_1(x_1, x_4, x_9) + w_2 f_2(x_2, x_3, x_6, x_8) + w_3 f_3(x_2, x_4, x_7, x_8) + w_4 f_4(x_2, x_6, x_7) + w_5 f_5(x_6, x_8), \quad (3)$$

where Q is the integral performance indicator of a web application; w_i denotes the weighting coefficients of the corresponding metrics; f_i denotes the functions used to form the performance indicators; and x_i denotes the system parameters that affect web page performance. Since the TTFB, FCP, LCP, INP, and CLS metrics have different units of measurement, their values were normalized to the dimensionless interval [0; 1] before aggregation. This ensures the correct application of weighting coefficients and the comparability of the contribution of individual indicators to the formation of the integral indicator Q. In the proposed model, the function f_1 describes the formation of the TTFB metric depending on the number of HTTP requests, the network connection speed, and the server response time. The function f_2 characterizes the influence of the volume of JavaScript and CSS resources, DOM structure parameters, and interface rendering parameters on the FCP indicator. The formation of the LCP metric is described by the function f_3, which takes into account the volume of client-side resources, rendering parameters, and JavaScript execution speed.

The INP metric is formed by the function f_4, which characterizes the influence of DOM structure complexity, browser main-thread load, and client-side script execution time on the speed of user interaction with the interface. To evaluate the stability of the page structure, the function f_5 is used; it determines the dependence of the CLS indicator on rendering parameters and the structure of the DOM tree. The use of weighting coefficients makes it possible to account for the varying degree of influence of individual metrics on the overall performance of a web application. In this study, the values of the weighting coefficients were determined based on Google Core Web Vitals recommendations and expert assessment of the importance of individual metrics for users' perception of web application performance. The highest weight was assigned to the LCP metric, which characterizes the rendering speed of the main page content. The lowest weight was assigned to the CLS metric, which characterizes interface stability.

Table 1

Metric	Abbreviation	Weight
Time To First Byte	TTFB	0.26
First Contentful Paint	FCP	0.22
Largest Contentful Paint	LCP	0.30
Interaction to Next Paint	INP	0.17
Cumulative Layout Shift	CLS	0.05

Within the framework of numerical simulation, the following assumptions were adopted: an increase in the volume of JavaScript resources leads to an increase in the LCP and INP indicators; an increase in the number of HTTP requests leads to an increase in TTFB and FCP; an increase in the complexity of the DOM structure deteriorates rendering indicators; and the CLS indicator varies within a limited range typical of modern web interfaces. The metric values presented in the tables were obtained as a result of scenario-based simulation in accordance with the specified assumptions. The numerical values of the parameters were selected in accordance with the typical ranges of web page performance metrics presented in Google Core Web Vitals recommendations and in contemporary studies on web application optimization. During the simulation, the values of the volume of JavaScript resources and the number of HTTP requests were varied within ranges typical of modern web applications.

The values of the TTFB, FCP, LCP, INP, and CLS metrics were determined in accordance with the functional dependencies embedded in the mathematical model. The obtained values were used to calculate the integral performance indicator Q. The results of the numerical simulation are presented in the form of tables and graphical dependencies, which make it possible to analyze the influence of individual parameters on web application performance. The obtained results were presented as graphical dependencies, which allow the nature of the influence of individual parameters on web application performance to be visually demonstrated, critical areas of performance degradation to be identified, and the contribution of individual metrics to the formation of the integral performance indicator to be assessed. Table 2 presents the results of numerical simulation of the dependence of performance indicators on the volume of JavaScript resources.

Table 2

JS, КБ	**TTFB, с**	**FCP, с**	**LCP, с**	**INP, мс**	**CLS**	**Q**
100	0.40	1.20	2.10	120	0.05	0.16
200	0.42	1.30	2.30	130	0.05	0.20
300	0.45	1.50	2.60	145	0.06	0.26
400	0.50	1.70	2.90	160	0.07	0.34
500	0.54	1.90	3.20	180	0.08	0.45
600	0.59	2.10	3.60	205	0.09	0.60
700	0.65	2.40	4.00	230	0.10	0.77
800	0.72	2.80	4.50	260	0.12	1.00

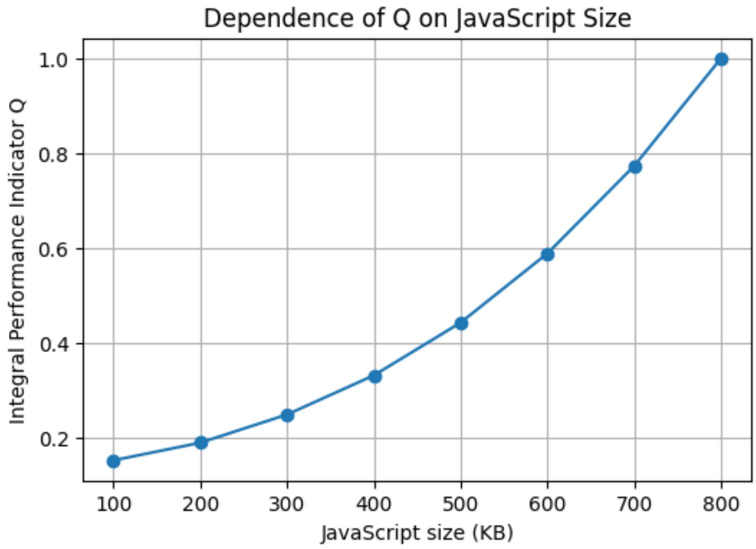

Fig. 1. Dependence of the integral performance indicator on the volume of JavaScript resources

Figure 1 presents the graph of the dependence of the integral performance indicator Q on the volume of JavaScript resources in a web application. The obtained dependence demonstrates the nonlinear nature of changes in system performance as the volume of client-side code increases. The analysis of the graph shows that, as the volume of JavaScript resources increases, there is a gradual increase in the value of the integral indicator Q, which indicates a deterioration in the overall performance of the web application. At relatively low values of JavaScript code volume, the change in the performance indicator is comparatively slow; however, after a certain threshold value is exceeded, an accelerated increase in the integral indicator is observed.

This nature of the dependence can be explained by the increased load on the browser's main thread during the processing and execution of JavaScript scripts. The greatest impact is observed on the LCP and INP metrics, which directly characterize the speed of content rendering and the responsiveness of the interface to user actions. This confirms the feasibility of applying client-side code optimization methods, in particular JavaScript file minification, code splitting mechanisms, lazy loading, and optimization of the rendering process of interface components. The obtained results demonstrate the consistency of the model behavior with the known patterns of web application functioning. Table 3 presents the results of numerical simulation of the dependence of performance indicators on the number of HTTP requests.

Table 3

HTTP	TTFB, с	FCP, с	LCP, с	INP, мс	CLS	Q
10	0.38	1.15	2.00	118	0.05	0.15
20	0.40	1.25	2.15	125	0.05	0.18
30	0.43	1.35	2.35	133	0.06	0.23
40	0.47	1.50	2.60	145	0.06	0.30
50	0.52	1.70	2.90	160	0.07	0.40
60	0.58	1.95	3.25	180	0.08	0.54
70	0.65	2.20	3.65	205	0.09	0.73
80	0.74	2.55	4.10	235	0.10	0.98

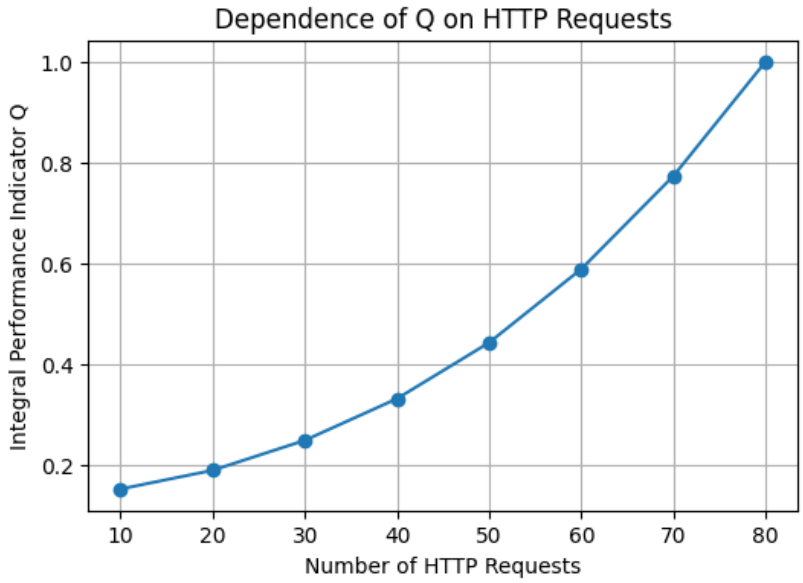

Fig. 2. Dependence of the integral performance indicator on the number of HTTP requests

Figure 2 presents the graph of the dependence of the integral performance indicator Q on the number of HTTP requests in a web application. The obtained dependence demonstrates a gradual deterioration in system performance as the number of network requests increases. The analysis of the graph shows that, with a small number of HTTP requests, the change in the integral performance indicator is relatively insignificant. However, as the number of network requests increases, an accelerated increase in the value of the indicator Q is observed, which indicates the nonlinear nature of the influence of this parameter on web application performance.

This nature of the dependence can be explained by the increased time required to establish network connections, transmit data, and process server responses. Each additional HTTP request creates an additional load on the network infrastructure and the server-side component of the system, which leads to an increase in user waiting time and a deterioration in the TTFB and FCP indicators. A particularly noticeable deterioration in performance is observed after a certain threshold number of HTTP requests is exceeded. This indicates that, under a high level of network interaction, the system begins to operate less efficiently due to the accumulation of delays during resource loading and the synchronization of the client-side and server-side components of the web application.

The obtained results confirm the need to optimize the resource structure of a web application and reduce the number of HTTP requests. To improve performance, it is advisable to use mechanisms such as resource bundling, caching, asynchronous data loading, and content delivery networks (CDNs), which make it possible to reduce the load on the network infrastructure and shorten the web page loading time.

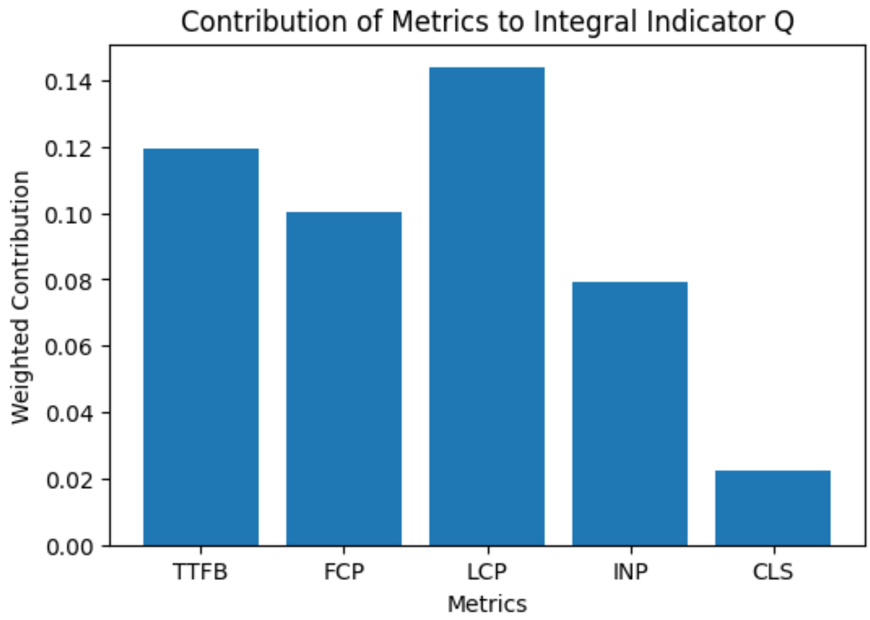

Fig. 3. Contribution of performance metrics to the formation of the integral performance indicator

Figure 3 presents a diagram of the weighted contribution of individual performance metrics to the formation of the integral indicator Q for one of the numerical simulation scenarios. The constructed diagram reflects not only the adopted weighting coefficients but also the relative contribution of each metric to the overall value of the integral performance indicator. The LCP metric makes the largest weighted contribution to the formation of the integral indicator, which is explained by its significant influence on users' perception of web page performance. The Time to First Byte (TTFB) and First Contentful Paint (FCP) metrics also have a considerable impact on the integral indicator, as they characterize the speed of the server response and the time required to display the first content element, respectively. High values of these indicators indicate the presence of delays at the initial stages of web page loading and negatively affect the perceived performance of the system.

The Interaction to Next Paint (INP) metric has a somewhat smaller, yet still significant, impact on the overall performance indicator. This metric characterizes the speed of the interface response to user interaction and reflects the level of interactivity of the web application. A deterioration in the INP value may be associated with an overloaded browser main thread or inefficient organization of client-side scripts.

The Cumulative Layout Shift (CLS) indicator has the smallest contribution within the proposed model, as it characterizes the stability of the page structure during loading. Despite its lower weighted influence, the CLS value remains important for ensuring correct user interaction with the web application and preventing undesirable shifts of interface elements. The use of weighting coefficients is appropriate within the proposed model, as it makes it possible to account for the different degrees of influence of individual metrics on the overall system performance and ensures a comprehensive evaluation of web page performance.

The practical significance of the developed model lies in its ability to predict the behavior of a web application under different system configurations and load levels. This makes it possible to identify the parameters whose optimization will provide the greatest improvement in web resource performance.

The obtained results confirm the feasibility of using multifactor mathematical modeling to evaluate the performance of modern web applications and demonstrate the possibility of applying the proposed model to analyze the effectiveness of web resource optimization methods.

Model Verification. The verification of the proposed mathematical model was carried out by analyzing the consistency of the numerical simulation results with the known patterns of influence of web application parameters on Core Web Vitals indicators.

The simulation results showed that an increase in the volume of JavaScript resources leads to a deterioration in the LCP and INP indicators, while an increase in the number of HTTP requests negatively affects the TTFB and FCP indicators. The obtained dependencies are consistent with the results of contemporary studies in the field of web application optimization and with Core Web Vitals recommendations.

This indicates the possibility of using the proposed model for the qualitative analysis of factors affecting the performance of web applications. Conclusions and Prospects for Further Research. The study analyzed the main factors affecting the performance of web applications and examined contemporary metrics for evaluating web page performance. The features of the formation of TTFB, FCP, LCP, INP, and CLS indicators were considered, as these metrics characterize page loading speed, interface stability, and the efficiency of user interaction with a web application. Within the framework of the study, a mathematical model for evaluating web application performance was proposed, which formalizes the relationship between the parameters of the client-side and server-side components of the system and web page performance indicators. The developed model takes into account the multifactorial influence of system parameters on the integral performance indicator and enables a comprehensive evaluation of web application performance.

In the course of the study, the main input parameters of the model were identified, including the number of HTTP requests, the volume of JavaScript and CSS resources, network connection speed, caching parameters, the size of the DOM structure, JavaScript execution time, and interface rendering parameters. Web page performance metrics were used as output parameters, making it possible to evaluate different aspects of web application performance.

Based on the proposed model, a series of numerical experiments was conducted to investigate the influence of individual system parameters on the integral performance indicator. The constructed graphical dependencies demonstrated the nonlinear nature of the influence of the volume of JavaScript resources and the number of HTTP requests on web application performance. The numerical simulation results showed that the volume of JavaScript resources and the number of HTTP requests have the greatest impact on web page performance;

their increase leads to the deterioration of LCP, INP, TTFB, and FCP indicators. The proposed mathematical model can be used to analyze the effectiveness of web application optimization methods, evaluate the impact of changes in system architecture on web page performance, and predict the behavior of a web resource under different load conditions. The practical application of the model provides an opportunity for the formalized analysis of web application performance and the identification of critical factors contributing to performance degradation.

Prospects for further research involve extending the proposed model by taking into account nonlinear dependencies between system parameters, temporal characteristics of load, and the combined influence of performance factors. In addition, the application of machine learning methods for the automatic determination of model weighting coefficients and the real-time prediction of web application performance indicators represents a promising direction for further research.

References

1. Grigorik, I. (2013). *High performance browser networking*. Sebastopol: O'Reilly Media.

2. Souders, S. (2007). *High performance web sites: essential knowledge for front-end engineers*. Sebastopol: O'Reilly Media.

3. Google. (n.d.). *Web Vitals*. URL: https://web.dev/vitals/ (access date: 16.05.2026).

4. Nielsen, J. (1993). Response times: The three important limits. Nielsen Norman Group. URL: https://www.nngroup.com/articles/response-times-3-important-limits/ (access date: 17.05.2026).

5. Han, S. (2022). Research on web front-end performance optimization based on XML. *Proceedings of the 2021 International Conference on Aviation Safety and Information Technology (ICASIT 2021)*, 700–704. DOI: 10.1145/3510858.3511366

6. Google Developers. (n.d.). *Lighthouse performance scoring*. URL: https://developer.chrome.com/docs/lighthouse/performance/performance-scoring/ (access date: 18.05.2026).

7. Laskar, N. M., Guha, K., & Paul, P. K. (2022). Random offset minimization in low frequency front-end amplifiers using swarm intelligence based techniques. *Evolutionary Intelligence, 15*(2), 1385–1397. DOI: 10.1007/s12065-020-00495-5

8. Everts, T. (2016). *Time is money: the business value of web performance*. Sebastopol: O'Reilly Media.

9. Nah, F. (2004). A study on tolerable waiting time: how long are web users willing to wait? *Behaviour & Information Technology, 23*(3), 153–163.

10. Jamshidi, P., et al. (2023). Optimize along the way: an industrial case study on web performance engineering of single-page applications. *Journal of Systems and Software, 197*, 111585. DOI: 10.1016/j.jss.2022.111593

11. Vogel, L., & Springer, T. (2022). An in-depth analysis of web page structure and efficiency with focus on optimization potential for initial page load. In *Web Engineering (ICWE 2022)*, 101–116. DOI: 10.1007/978-3-031-09917-5_7

12. Karka, N. R. (2025). Front-end performance optimization: a comprehensive guide to reducing main-thread blocking and DOM complexity. *ResearchGate Framework Studies, 44*(2), 89–104. DOI: 10.32628/CSEIT251112389

13. Patel, M., & Sharma, R. (2023). Web vitals and core metrics for web performance optimization using synthetic monitoring tools. *International Journal of Core Engineering & Management, 7*(6), 201–214. DOI: 10.5281/zenodo.14915137

14. Timanovskyi, Y., & Plechawska-Wojcik, M. (2020). Performance analysis of selected tools for building a single page application. *Journal of Computer Sciences Institute, 14*, 82–87. DOI: 10.35784/jcsi.1581