

CHECK OUT OUR TECHNICAL AUDIT REPORT SAMPLE

Last update 31 January 2024

Dive into the inner workings of our gaming platform as we conduct a rigorous technical audit. Uncover insights on performance enhancements, security fortifications, and overall system optimization. Elevate your gaming experience with the results of our in-depth examination.

Build performance

UI and Backend build separately, which positively affects on build performance. According to build info, UI build takes in average 2 minutes 55 seconds that looks nice and allows roll out new features and critical fixes really fast.

Ok, let's look at what an engagement model should do. Feel free to use it as a checklist while choosing your software development partner.

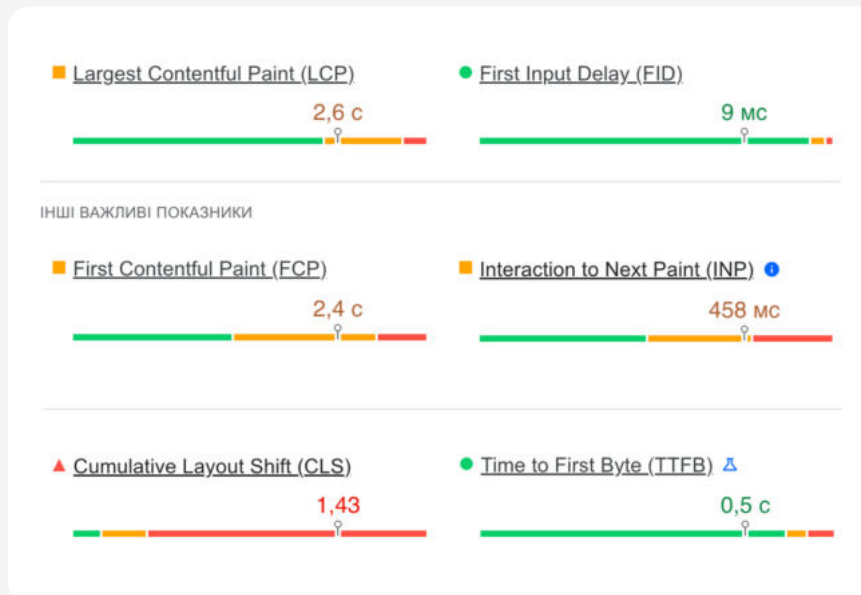
- ◆ Halo Lab's deployment went live
 - 2m 36s build 01:55:55 PM
- ◆ Halo Lab's deployment went live
 - 3m 46s build 08:32:43 PM
- ◆ Halo Lab's deployment went live
 - 1m 52s build 05:42:56 PM
- ◆ Halo Lab's deployment went live
 - 3m 58s build 07:22:34 PM
- ◆ Halo Lab's deployment went live
 - 2m 23s build 09:16:03 AM

But as usual applications may grown over years likewise their size. So it would be good point, from time to time refactor and clean code from legacy.

- ◆ Halo Lab's deployment went live
 - 2m 36s build 02:54:01 PM
- ◆ Halo Lab's deployment went live
 - 1m 48s build 01:28:48 PM
- ◆ Halo Lab's deployment went live
 - 0m 54s build 10:11:38 AM

API build time is more spread, but not critical. As we see last deploy took less than minute.

Runtime performance



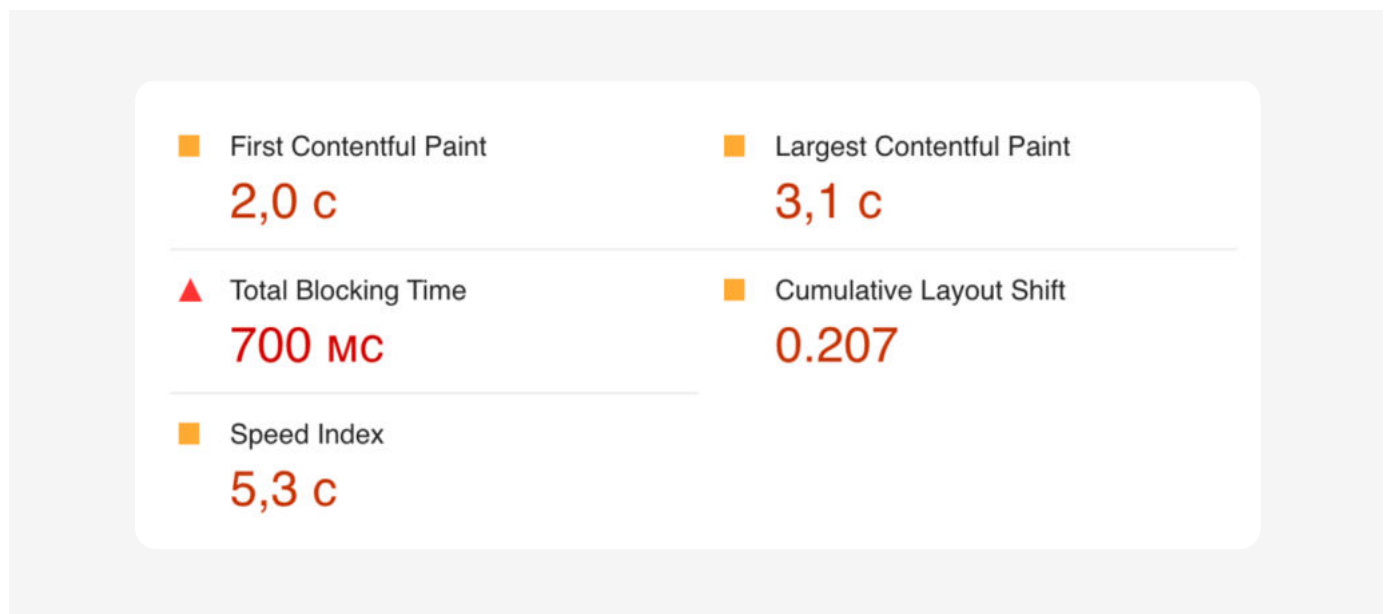
At the first sight metrics show us good result, but also there are things which can be improved:

- ❖ **FCP** - First Contentful Paint, measures how long it takes the browser to render the first piece of DOM content after a user navigates to your page. Yellow field means that it expects improvement. In most cases influence on this metric caused by size of project. So appropriate code refactoring, clean legacy and unused code should improve value.
- ❖ **LCP** - Largest Contentful Paint, measures when the largest content element in the viewport is rendered to the screen. Diving into diagnostic, the reason that affects on this value is load delay (Delay between TTFB and loading largest content). For optimization this value, browser has to detect this image earlier.

At the first sight metrics show us good result, but also there are things which can be improved:

- ❖ **FCP** - First Contentful Paint, measures how long it takes the browser to render the first piece of DOM content after a user navigates to your page. Yellow field means that it expects improvement. In most cases influence on this metric caused by size of project. So appropriate code refactoring, clean legacy and unused code should improve value.
- ❖ **CLS** - Cumulative Layout Shift, measures of the largest burst of layout shift scores for every unexpected layout shift that occurs during the entire lifespan of a page. This value looks really bad. However, visually, it's impossible to catch any shifts. Diagnostic information tell us that the problem is related to sliders which are built by third-party library "keen slider". Perhaps it this library is a root of a problem.
- ❖ **FID** - First Input Delay, measures the time from when a user first interacts with a page and browser can process them. The value looks satisfied.
- ❖ **TTFB** - Time To First Byte, measures the time between the request for a resource and when the first byte of a response begins to arrive. Also has a good value.
- ❖ **SI** - Speed Index, measures how quickly content is visually displayed during page load. The value is 1.8s which good enough and count as fast

Also there is one more option which should be mentioned TBT - Total Blocking Time. For now, for desktop value is not critical but highly recommended to be reviewed, as DOM is oversized which affects on styles computing and templates rerendering. This caused by non-optimized .svg files, which should be simplified, and large quantity of child elements in slider.



Mobile device performance test looks worse.

As we see from above SI is much slower and Total Blocking Time is critical. Things that impact on that the most, is script evaluation, which takes 2.6s from 4.1s of main thread work (63%). Perhaps it needs to optimize the code to avoid unnecessary calculations.

Deep-dive into code

Code structure

UI Application based on NextJs framework to implement SSR for better SEO, and optimization. However, such approach may cause some bad values described in **performance section**.

Code structured regarding to NextJS demands. But some of files looks messy and it is hard to maintain them.

Imports order can be described as reusable(src functions) -> framework built-in/ third-party modules -> neighbors modules -> styles.

Sometimes imports occupy a large part of file as shown below. The solution could be in creating file on path “@sm/[folder-name]/[storage-file]” that storage all modules related to parent folder responsibility.

However, folder structure is not always clear. For example src/types defined as a storage typescript data types, but among all code occur separate files responsible for type defining and types defined in modules. Perhaps, better idea is to storage All the types in one place.

```
1  import { useCategories } from '@sm/api/api-hooks/useCategories';
2  import { useTags } from '@sm/api/api-hooks/useTags';
3  import { useUsers } from '@sm/api/api-hooks/useUsers';
4  import { IGame, IOption } from '@sm/api/common';
5  import GameRequest from '@sm/api/request/GameRequest';
6  import { useAuth } from '@sm/components/Auth';
7  import { ClearButton } from '@sm/components/common/Buttons/ClearButton';
8  import { CustomSelect } from '@sm/components/common/FormComponents/CustomSelect';
9  import { CommonDatePicker } from '@sm/components/common/FormComponents/DatePicker/CommonDatePicker';
10 import { FilterSvg } from '@sm/components/common/Icons';
11 import { QueryPagination } from '@sm/components/common/LandingPagination/QueryPagination';
12 import { Loader } from '@sm/components/common/Loader';
13 import { NoData } from '@sm/components/common/NoData';
14 import { useSearch } from '@sm/components/common/Search';
15 import { TableHeaderItem } from '@sm/components/common/TableHeaderItem';
16 import { SearchFilterOptions } from '@sm/components/landing/Search/SearchFilters';
17 import { platformList } from '@sm/helpers/constants';
18 import { convertDate } from '@sm/helpers/convert-date';
19 import { makeAlias } from '@sm/helpers/makeAlias';
20 import * as notify from '@sm/helpers/notifications';
21 import { ISortDirection } from '@sm/types/blog';
22 import {
23   IGameInitialValues,
24   IGameRequestParams,
25   IGameStatus,
26   IOpen,
27   ISortType,
28 } from '@sm/types/game';
```

In order to create a separate json file with static data, all static objects are declared in the same file. On the one hand it makes folder structure more clear, but on the other hand, it affects also on file readability.

Styles

For styles is used preprocessor SCSS. Opportunities of this preprocessor are widely used among all application. In general scss file length average is 150-200 lines (in rare cases 300+), but amount of this files is quite big. Perhaps, it can be optimized and shortened. As a first example of solution, property “font-size” occurs 345 times and in the most cases with other font properties, like “font-weight”, “line-height” etc. It can make sense to create mixin that take just font-size as required argument, and other font-related params with default “inherit” value. Otherwise, from 345 ‘font-size’'s with value “14px” occurs 92 times (1st place) and only 4 times inside media query. Perhaps it should set as default size, and it may save 88+ strokes of code.

Sometimes occurs rewritten properties (Example below taken from “ForgotPassword.module.scss”, line 35)

```
.bigText {  
  font-size: 16px;  
  line-height: 155%;  
  color: var(--body);  
  font-size: 14px;  
  
  letter-spacing: -0.01em;  
  p {  
    margin-bottom: 30px;  
  }  
}
```

Sometimes properties are same for 2 selector with one difference. Looks like it could be easily separated via comma.(Example below taken from “QueryPagination.module.scss”, line 137)

```
.resultPage {  
  font-weight: 500;  
  font-size: 14px;  
  line-height: 125%;  
  color: $option;  
}  
.Items {  
  font-weight: 500;  
  font-size: 14px;  
  line-height: 125%;  
  color: $option;  
  cursor: pointer;  
}
```

Also occurs some minor lexical errors. For example, attribute “data-name=’dawn’”. According to context, it should be ‘down’ nor the ‘dawn’(period of day). It is not critical, in case it is not used by screen readers as well as pronunciation is different.

API

FAPI based on ExpressJs framework with module approach. The possibility of database migration is provided and described in README. Code is clear, concise, well-organized and complies with the OOP principles.

Infrastructure Analysis

Infrastructure is robust and satisfied. Deploy process is simple and doesn't demand any extra interventions except setup environment variables.

For UI, API and Database are used same cloud provider, that also positively affects on project speed. Services to prevent insecure interaction are provided.

1. Broken page links

The website has links to 404 pages.

For example:

❖ **Page:** <https://www.site.com/resources-old>

PDF for download with 404 error:

❖ <https://www.site.com/wp-content/uploads/2020/01/book-1.pdf>

❖ <https://www.site.com/wp-content/uploads/2020/01/book-1.pdf>

What we need to do

We need to replace them with accessible pages.

List of pages where you can find broken links:

❖ [Broken page links.](#)

2. 3xx redirected pages

Incorrect link to the Privacy Policy from the footer.

The site uses: <https://www.site.com/reference/privacy-policy>

In this block: <https://www.site.com/reference/privacy-policy-2021> – with 301 redirects to the correct version.

What we need to do

We need to replace the link with the correct one. There are also links to other internal pages that have redirects. They need to be replaced with the correct ones.

List of such links, on which pages they can be found and what to replace them with:

❖ [3xx page links.](#)

3. Duplicate pages

The site contains duplicate pages of 2 types:

❖ Pages with get-parameters (campaigns).

What we need to do

We need to configure Canonical on static pages to point to themselves.

❖ Pages in different directories.

We must select the main page and set a 301 redirect to it. After that, we need to replace the links on the website with the correct ones.

4. Multiple H1

The `<h1>` header is duplicated on some pages of the site. The largest number of them are in the “Learn more with our resources” block.

What we need to do

We need to decide whether they should remain h1. The error is not critical, but from the point of view of the logic of the page, it is better to fix it.

5. Hidden blocks and texts

On the page <https://www.site.com/offering/academy>, there is a hidden block. We can see it using the CSS property “display: none”, specifically a div with the class “page-wrapper hide”.

If we remove the “display: none” CSS property, we see that this block consists of a header, main content, and footer. In simple words, it’s another page within the page, hidden from users but visible for Google’s robots.

Such hidden content could be interpreted by search engine algorithms as manipulative behavior by site owners and may result in penalties or sanctions.

A screenshot of the page with the hidden block can be viewed [here](#).

A screen recording demonstrating the appearance of the hidden block when the display: none property is deactivated can be viewed [here](#).

On the page <https://www.site.com/services>, there is a section with the class “section_blog47 hide” that is not accessible to users due to the CSS property “display: none”

On the page <https://www.site.com/offering/keynote>, there’s a section with the class “section-faq2 hide” that’s also not accessible to users due to the CSS property “display: none”. This section contains a significant amount of content.

On the page <https://www.site.com/about-us>, sections with the classes “section-team3 hide” and “section-cta-block hide” are hidden. These sections are not accessible to website users but are visible to the Googlebot.

Additionally, there are some blocks of placeholder “Lorem Ipsum” text on some pages of the website. This can also have a negative impact on the overall quality of the site and its ranking in the top search results.

Placeholder text can be found on the pages <https://www.site.com/services> and <https://www.site.com/resources-search>.

How to fix

You need to remove hidden placeholder text from all pages. Also, remove any sections or blocks in the code of the pages that are hidden from users (but accessible to Google's robots) by utilizing the CSS "display: none" property. Such elements typically have the "hide" class.

6. The link to the 404 page

On the page <https://www.site.com/offering/academy>, there is a link that leads to a page displaying a 404 error. Although it's not accessible to website users, search engine bots can see this link in the page's code and subsequently access it.

How to fix

You should replace the link with the up-to-date one:

◆ <https://www.site.com/resources>.

7. Redirect chains

The website contains multiple redirect chains. These occur when several URLs redirect the user in sequence. For example, a chain of two redirects looks like this: URL 1 → URL 2 → URL 3.

How to fix

Eliminate redirect chains by redirecting straight to the final URL.

From <http://site.com/results> you need to set up a 301 redirect straight to <https://www.site.com/results>.

From <https://site.com/resources/> you need to set up a 301 redirect straight to <https://www.site.com/resources>.

8. Links to pages with the HTTP protocol

Within the website, there are multiple hyperlinks. In their code, the address is specified in the HTTP protocol format. While these links do redirect users to pages using the HTTPS protocol, this could potentially impact the page loading speed, especially when accessing the site through mobile internet.

How to fix

These links need to be replaced with those using the HTTPS protocol.

On the page <https://www.site.com/services> you need to change the link from <http://site.com/results> to <https://www.site.com/results>. Link anchor: "View all".

On the page <https://www.site.com/post/the-creative-act-of-management> you need to change the link from <http://www.site.com/> to <https://www.site.com/results>.

9. Links to pages with a 301 redirect

The website contains internal links to pages with a 301 redirect. Such links can complicate the process of site scanning for search engine bots and slow down the loading speed of the final page. For users, this situation might create a negative experience, leading them to leave the site before the desired page fully loads.

How to fix

Links in the code need to be replaced with the final URLs: [link to the document](#).

10. Pagination pages

10.1. Set up the URL

Pagination pages should feature friendly URLs, i.o. URLs without unnecessary numbers, symbols, or letters. Friendly URLs primarily serve to enhance user interaction with the website. They enable users to quickly understand which page of the site they are on.

As for the second page of pagination, its URL should include the expression "?page=2" in the GET parameter.

How to fix

You should establish user-friendly parameters for pagination page URLs and set up a 301 redirect from the existing pages to the new URLs. You can find the current list of pagination pages on the website through [this link](#). Additionally, don't forget to update the links within the "Next" and "Previous" buttons to the new and pertinent URLs.

10.2. Set up a 301 redirect from the first pagination page to the main category pages.

When transitioning from pagination pages to the first page, the URL should not contain any GET parameters.

For example, when transitioning from the second pagination page https://www.site.com/topic/team-happiness?ca589ac6_page=2 to the first pagination page, the URL of the first one should look like this: <https://www.site.com/topic/team-happiness>.

Currently, the URL of the first pagination page appears as follows: https://www.site.com/topic/team-happiness?ca589ac6_page=1, and this page is an exact duplicate of <https://www.site.com/topic/team-happiness>. This duplication negatively impacts the ranking of category pages.

How to fix

- ❖ Set up a 301 redirect from all first pagination pages with GET parameters to the main category pages. Check the redirects table we created: link to the document.
- ❖ Replace internal links with updated ones, i.e., without the GET parameter in the URL.

10.3. Optimize meta tags on pagination pages for uniqueness

When transitioning from pagination pages to the first page, the URL should not contain any GET parameters.

How to fix

The pagination page meta tags need to include the page numbers.

For example, Title of the page https://www.site.com/topic/team-happiness?ca589ac6_page=2 should look like this:

Title: Team Happiness | Page 2

Description: All successful teams in high growth organizations have certain things in common. Find out what they are and how to implement them on your team, starting today. | Page 2

11. The contact page is missing an H1 heading

Add tag H1 on the page <https://www.site.com/contact-us>. You can add the phrase “CONTACT US” to the H1, which is already present on the page.

This is not a critical error, but if possible, fix it.

12. Description of team members' pages

In some team members' pages, the Description tag is filled, but it is overly lengthy and duplicates the page's content entirely. In others, the Description tag remains unfilled. While team members' pages might not require tag optimization for specific queries, it's still recommended to fill them to enhance EAT ranking factors.

How to fix

For team pages, we suggest creating Descriptions using the template and adding unique meta tags to website pages. For author pages where Description needs to be changed or added, please refer to the document link. Description Template:

[Team Member's Name]: [Position]. Brief description of the individual's specialization and expertise.

Example Description: N***** M*****-Sy: Coach. N***** is very happy to provide services to people who are defining their career and life plans, determining their future path and moving forward.

13. Empty Description

In some pages, the Description tag is empty. But this is a crucial element of search engine optimization, as its content is displayed in search snippets. The more appealing the snippet, the higher the likelihood of users clicking through to the website.

If a page lacks a Description, search engines may automatically fill in this tag, and its content could potentially be irrelevant or uninteresting to users.

How to fix

We have provided unique Descriptions for the pages that required them. They need to be added to the website, please refer to the [document link](#).

14. Images without the “alt” attribute

The absence of “alt” attributes in image tags represents a missed opportunity for textual optimization of both images and the pages they are placed on. For pages where images constitute the primary content, using the “alt” attribute is critically important.

How to fix

Each significant image on the website needs to have an “alt” attribute, as it provides a brief and concise description of its content. Alternatively, the “alt” attribute can be filled using the template [Page H1 + Photo 1]. The list of pages with images lacking the “alt” tag can be found in the [document link](#).

15. Robots.txt file

Currently, the file is left empty, allowing Google bots to freely scan all pages of the website, including those that should be restricted from indexing.

Pages like search pages, registration pages, and others fall into this category. Therefore, it's necessary to block search engine bots' access to these pages.

How to fix

We have created a new robots.txt file, and it needs to be updated on the server to implement the necessary access restrictions.

User-agent: *

Disallow: /resources-search

Disallow: /login

Disallow: /log-in

Disallow: /sign-up

Disallow: /update-password

Disallow: *utm=
Disallow: ?ref=
Disallow: /search
Disallow: /access-denied
Disallow: #w-tabs
Sitemap: <https://www.site.com/sitemap.xml>

16. Link to the search page

To utilize the site search, you need to first navigate to the search page at <https://www.site.com/resources-search>. As a result, a non-priority page in terms of search engine optimization has a high number of internal links pointing to it.

How to fix

The search functionality needs to be modified. When users click on the search icon, they shouldn't be redirected to a new page. Instead, users should be able to enter their query in the search bar on the same page, and only after that, they need to be redirected to the search results page.

17. Page loading speed

Page loading speed analysis was conducted using the PageSpeed Insights tool, focusing on the page <https://www.site.com/offering/academy>. As you see, the loading speed of the mobile version can be improved. Some metrics show average or critical values.

While this isn't a critical issue, we recommend addressing certain aspects if possible.

Some pages have a too long server response time (over 0.7 s), [link to the document](#). Additionally, certain images have a large file size, contributing to poor loading metrics.

For instance, we see that scripts and videos have the longest loading times on the page. If possible, you should optimize them or set up a lazy loading.

Heavy images and photos slow down the website's performance. They should be optimized by reducing their size using specialized compressors before uploading them to the site. These tools compress images without compromising their quality. Additionally, utilizing more lightweight and modern formats (such as WEBP) and implementing lazy loading can further enhance the loading speed.

We recommend combining the main CSS files into a single file to reduce the number of server requests and removing any unused CSS styles. Afterwards, you can further optimize the file using a minifier to reduce its size.

Additionally, it's important to set up effective caching rules, especially for images, fonts, and heavy scripts. This way, when users revisit the website, page loading will be significantly faster.

Improvement ideas and proposed solutions

To start with, it would be nice to have well-written documentation, described project basis such as folder/file structure, commands, demanding, general code-writing rules and other things that will help newbie with onboarding and keep a good code quality. Encouraging developers to write explanation comments should be welcome.

Code refactoring and legacy removing also should be implemented before moving ahead as well as upgrading libraries.

More accurate recommendations were described in previous sections.