

There were issues affecting this run of Lighthouse:

- No ads were requested when fetching this page.
- No ads were rendered when rendering this page.
- The GPT tag was not requested.
- No ads were rendered when rendering this page.
- There may be stored data affecting loading performance in this location: IndexedDB. Audit this page in an incognito window to prevent those resources from affecting your scores.



Performance

Values are estimated and may vary. The [performance score is calculated](#) directly from these metrics. [See calculator.](#)

▲ 0–49 ■ 50–89 ● 90–100

METRICS

[Expand view](#)

● First Contentful Paint

0.9 s

● Total Blocking Time

0 ms

■ Speed Index

2.1 s

▲ Largest Contentful Paint

4.4 s

● Cumulative Layout Shift

0.017

 [View Treemap](#)

[View Original Trace](#)



DIAGNOSTICS

- ▲ Image elements do not have explicit `width` and `height` ^

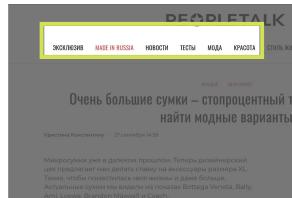
Set an explicit width and height on image elements to reduce layout shifts and improve CLS. [Learn how to set image dimensions](#) CLS

URL

yandex.net

...

img 230907_adfox_2614479_7672043.7abba6d....jpg/optimize.webp (avatars.mds.yandex.net)



...

img 230907_adfox_2614479_7672043.7abba6d....jpg/optimize.webp (avatars.mds.yandex.net)

img



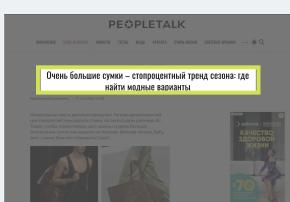
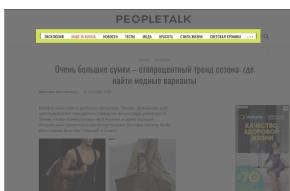
img ...231020_adfox_2692556_6bd7128...

<librederm300100.jpg/optimize.webp> (avatars.mds.yandex.net)

img

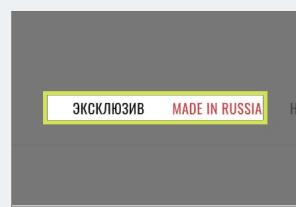
- Avoid large layout shifts — 5 elements found ^

These DOM elements contribute most to the CLS of the page. [Learn how to improve CLS](#) CLS

Element	CLS Contribution
 <code>h1.col-lg-9.mx-auto.text-center</code>	0.008
 <code>ul.site__menu-list.list-unstyled</code>	0.005

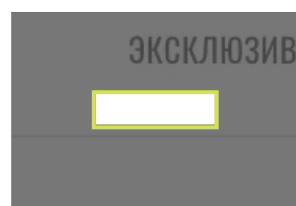
Element

CLS Contribution



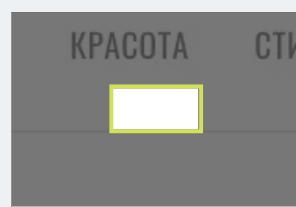
div.item-content

0.002



button.mm_marker_top

0.001



button.mm_button_top

0.001

More information about the performance of your application. These numbers don't [directly affect](#) the Performance score.

PASSED AUDITS

Hide

○ Avoid non-composited animations

^

Animations which are not composited can be janky and increase CLS. [Learn how to avoid non-composited animations](#)
[CLS]



Accessibility

These checks highlight opportunities to [improve the accessibility of your web app](#). Only a subset of accessibility issues can be automatically detected so manual testing is also encouraged.

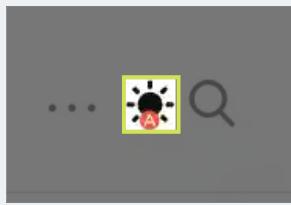
NAMES AND LABELS

▲ Buttons do not have an accessible name

^

When a button doesn't have an accessible name, screen readers announce it as "button", making it unusable for users who rely on screen readers. [Learn how to make buttons more accessible](#).

Failing Elements



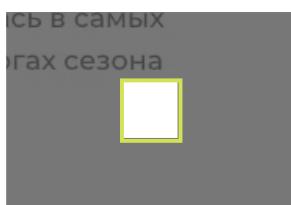
button.toggle__btn



button.btn.btn-favourite.btn-favourite-7663491



button.btn.btn-favourite.btn-favourite-7695411



button.btn.btn-favourite.btn-favourite-7558141

button.btn.btn-favourite.btn-favourite-7850651

button.btn.btn-favourite.btn-favourite-7850791

button.btn.btn-favourite.btn-favourite-7850551

button.btn.btn-favourite.btn-favourite-7850731

button.btn.btn-favourite.btn-favourite-7850571

button.btn.btn-favourite.btn-favourite-7834791

button.btn.btn-favourite.btn-favourite-7850341

button.btn.btn-favourite.btn-favourite-7850101

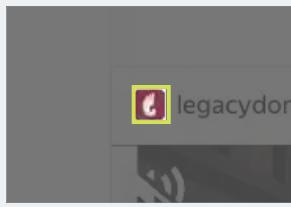
button.btn.btn-favourite.btn-favourite-7850111

⚠ Image elements do not have [\[alt\]](#) attributes

^

Informative elements should aim for short, descriptive alternate text. Decorative elements can be ignored with an empty alt attribute. [Learn more about the alt attribute.](#)

Failing Elements



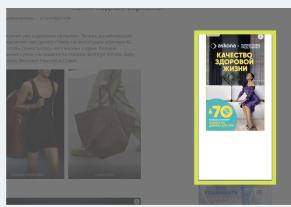
img.sd67f5606.h29430ea.a64e89e4b

▲ `<frame>` or `<iframe>` elements do not have a title

^

Screen reader users rely on frame titles to describe the contents of frames. [Learn more about frame titles.](#)

Failing Elements



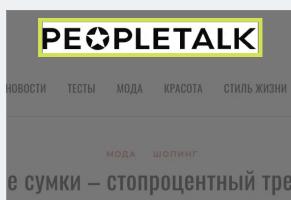
iframe

▲ Links do not have a discernible name

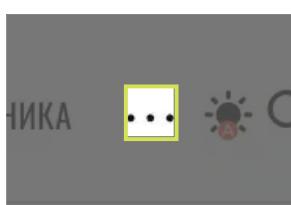
^

Link text (and alternate text for images, when used as links) that is discernible, unique, and focusable improves the navigation experience for screen reader users. [Learn how to make links accessible.](#)

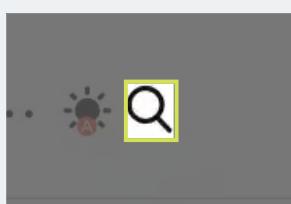
Failing Elements



a.logo-pick



a



a.btn-search--open.p-0

a.l9c3d820e

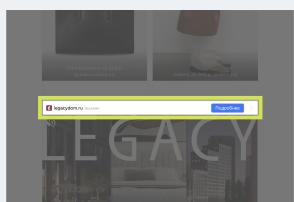
Failing Elements



a.l9c3d820e



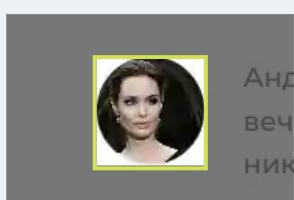
a



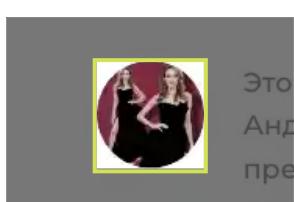
a.c4df1c7fa



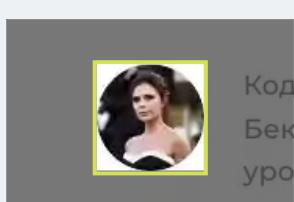
a



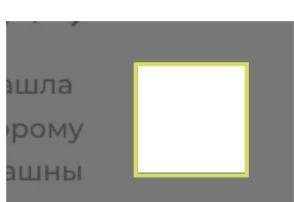
a.article-image



a.article-image



a.article-image



a.article-image



a.article-image

Failing Elements



a.article-image



a.article-image

a.article-image

a.article-image

a.article-image

a.article-image

a.article-image

These are opportunities to improve the semantics of the controls in your application. This may enhance the experience for users of assistive technology, like a screen reader.

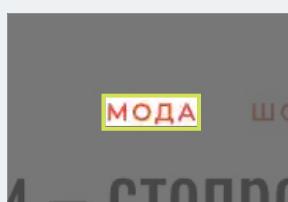
CONTRAST

- ▲ Background and foreground colors do not have a sufficient contrast ratio.

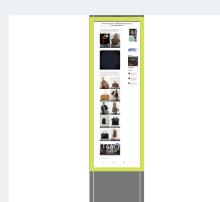


Low-contrast text is difficult or impossible for many users to read. [Learn how to provide sufficient color contrast.](#)

Failing Elements



span



div.js__read-more-article



span

Failing Elements

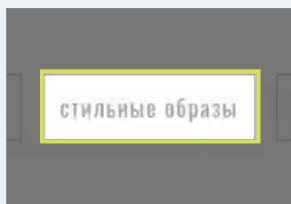
The image displays five vertically stacked screenshots of a website interface, each with a yellow box highlighting a specific element and a corresponding CSS class label below it.

- Screenshot 1:** A dark header bar with the text "А ШОПИНГ" and "ОПРОЦЕНТЫ". Below the header is a sidebar with a vertical list of items. The sidebar is highlighted with a yellow box. The label is "div.js__read-more-article".
- Screenshot 2:** A dark sidebar with the text "Made in Russia" and "тренды осень-зима 2". The "Made in Russia" text is highlighted with a yellow box. The label is "a.btn.btn-secondary.btn-secondary-gray.font-weight-normal.site__tags-item.p-2".
- Screenshot 3:** A dark sidebar with the text "сия", "аксессуары", "мо", "ть-зима 2023/2024", and "шоп". The "аксессуары" text is highlighted with a yellow box. The label is "a.btn.btn-secondary.btn-secondary-gray.font-weight-normal.site__tags-item.p-2".
- Screenshot 4:** A dark sidebar with the text "аксессуары", "мода", "стиль", "2024", and "шопинг". The "мода" text is highlighted with a yellow box. The label is "div.js__read-more-article".
- Screenshot 5:** A dark sidebar with the text "мода", "стиль", "стиль", "шопинг", and "шопинг". The "стиль" text is highlighted with a yellow box. The label is "a.btn.btn-secondary.btn-secondary-gray.font-weight-normal.site__tags-item.p-2".

Failing Elements



div.js__read-more-article



a.btn.btn-secondary.btn-secondary-gray.font-weight-normal.site__tags-item.p-2



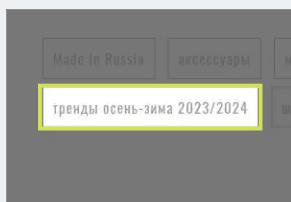
div.js__read-more-article



a.btn.btn-secondary.btn-secondary-gray.font-weight-normal.site__tags-item.p-2



div.js__read-more-article



a.btn.btn-secondary.btn-secondary-gray.font-weight-normal.site__tags-item.p-2



div.js__read-more-article

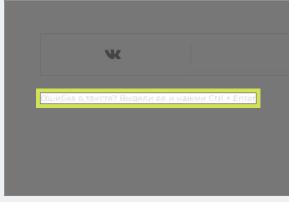


a.btn.btn-secondary.btn-secondary-gray.font-weight-normal.site__tags-item.p-2



div.js__read-more-article

Failing Elements



span

div.js__read-more-article

These are opportunities to improve the legibility of your content.

AUDIO AND VIDEO

- <video> elements contain a <track> element with [kind="captions"]

When a video provides a caption it is easier for deaf and hearing impaired users to access its information. [Learn more about video captions.](#)

Failing Elements



video.Gul

These are opportunities to provide alternative content for audio and video. This may improve the experience for users with hearing or vision impairments.

ADDITIONAL ITEMS TO MANUALLY CHECK (10)

Hide

- The page has a logical tab order

Tabbing through the page follows the visual layout. Users cannot focus elements that are offscreen. [Learn more about logical tab ordering.](#)

- Interactive controls are keyboard focusable

Custom interactive controls are keyboard focusable and display a focus indicator. [Learn how to make custom controls focusable.](#)

○ Interactive elements indicate their purpose and state ^

Interactive elements, such as links and buttons, should indicate their state and be distinguishable from non-interactive elements. [Learn how to decorate interactive elements with affordance hints](#).

○ The user's focus is directed to new content added to the page ^

If new content, such as a dialog, is added to the page, the user's focus is directed to it. [Learn how to direct focus to new content](#).

○ User focus is not accidentally trapped in a region ^

A user can tab into and out of any control or region without accidentally trapping their focus. [Learn how to avoid focus traps](#).

○ Custom controls have associated labels ^

Custom interactive controls have associated labels, provided by aria-label or aria-labelledby. [Learn more about custom controls and labels](#).

○ Custom controls have ARIA roles ^

Custom interactive controls have appropriate ARIA roles. [Learn how to add roles to custom controls](#).

○ Visual order on the page follows DOM order ^

DOM order matches the visual order, improving navigation for assistive technology. [Learn more about DOM and visual ordering](#).

○ Offscreen content is hidden from assistive technology ^

Offscreen content is hidden with display: none or aria-hidden=true. [Learn how to properly hide offscreen content](#).

○ HTML5 landmark elements are used to improve navigation ^

Landmark elements (<main>, <nav>, etc.) are used to improve the keyboard navigation of the page for assistive technology. [Learn more about landmark elements](#).

These items address areas which an automated testing tool cannot cover. Learn more in our guide on [conducting an accessibility review](#).

PASSED AUDITS (16)

Hide

● [aria-*] attributes match their roles ^

Each ARIA role supports a specific subset of `aria-*` attributes. Mismatching these invalidates the `aria-*` attributes.

[Learn how to match ARIA attributes to their roles.](#)

- `[aria-hidden="true"]` is not present on the document `<body>`

Assistive technologies, like screen readers, work inconsistently when `aria-hidden="true"` is set on the document `<body>`. [Learn how `aria-hidden` affects the document body.](#)

- `[role]`s have all required `[aria-*]` attributes

Some ARIA roles have required attributes that describe the state of the element to screen readers. [Learn more about roles and required attributes.](#)

- `[role]` values are valid

ARIA roles must have valid values in order to perform their intended accessibility functions. [Learn more about valid ARIA roles.](#)

- `[aria-*]` attributes have valid values

Assistive technologies, like screen readers, can't interpret ARIA attributes with invalid values. [Learn more about valid values for ARIA attributes.](#)

- `[aria-*]` attributes are valid and not misspelled

Assistive technologies, like screen readers, can't interpret ARIA attributes with invalid names. [Learn more about valid ARIA attributes.](#)

- `[user-scalable="no"]` is not used in the `<meta name="viewport">` element and the `[maximum-scale]` attribute is not less than 5.

Disabling zooming is problematic for users with low vision who rely on screen magnification to properly see the contents of a web page. [Learn more about the viewport meta tag.](#)

- `button`, `link`, and `menuitem` elements have accessible names

When an element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn how to make command elements more accessible.](#)

- Document has a `<title>` element

The title gives screen reader users an overview of the page, and search engine users rely on it heavily to determine if a page is relevant to their search. [Learn more about document titles.](#)

- `[id]` attributes on active, focusable elements are unique

All focusable elements must have a unique id to ensure that they're visible to assistive technologies. [Learn how to fix duplicate ids.](#)

● `<html>` element has a `[lang]` attribute ^

If a page doesn't specify a lang attribute, a screen reader assumes that the page is in the default language that the user chose when setting up the screen reader. If the page isn't actually in the default language, then the screen reader might not announce the page's text correctly. [Learn more about the lang attribute.](#)

● `<html>` element has a valid value for its `[lang]` attribute ^

Specifying a valid [BCP 47 language](#) helps screen readers announce text properly. [Learn how to use the lang attribute.](#)

● Lists contain only `` elements and script supporting elements (`<script>` and `<template>`). ^

Screen readers have a specific way of announcing lists. Ensuring proper list structure aids screen reader output. [Learn more about proper list structure.](#)

● List items (``) are contained within ``, `` or `<menu>` parent elements ^

Screen readers require list items (``) to be contained within a parent ``, `` or `<menu>` to be announced properly. [Learn more about proper list structure.](#)

● No element has a `[tabindex]` value greater than 0 ^

A value greater than 0 implies an explicit navigation ordering. Although technically valid, this often creates frustrating experiences for users who rely on assistive technologies. [Learn more about the tabindex attribute.](#)

● Heading elements appear in a sequentially-descending order ^

Properly ordered headings that do not skip levels convey the semantic structure of the page, making it easier to navigate and understand when using assistive technologies. [Learn more about heading order.](#)

NOT APPLICABLE (26) Hide

○ `[accesskey]` values are unique ^

Access keys let users quickly focus a part of the page. For proper navigation, each access key must be unique. [Learn more about access keys.](#)

○ `[aria-hidden="true"]` elements do not contain focusable descendants ^

Focusable descendants within an `[aria-hidden="true"]` element prevent those interactive elements from being available to users of assistive technologies like screen readers. [Learn how aria-hidden affects focusable elements.](#)

○ ARIA input fields have accessible names ^

When an input field doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more about input field labels](#).

○ ARIA `meter` elements have accessible names ^

When a meter element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn how to name meter elements](#).

○ ARIA `progressbar` elements have accessible names ^

When a progressbar element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn how to label progressbar elements](#).

○ Elements with an ARIA `[role]` that require children to contain a specific `[role]` have all required children. ^

Some ARIA parent roles must contain specific child roles to perform their intended accessibility functions. [Learn more about roles and required children elements](#).

○ `[role]`s are contained by their required parent element ^

Some ARIA child roles must be contained by specific parent roles to properly perform their intended accessibility functions. [Learn more about ARIA roles and required parent element](#).

○ ARIA toggle fields have accessible names ^

When a toggle field doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more about toggle fields](#).

○ ARIA `tooltip` elements have accessible names ^

When a tooltip element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn how to name tooltip elements](#).

○ ARIA `treeitem` elements have accessible names ^

When a treeitem element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more about labeling treeitem elements](#).

○ The page contains a heading, skip link, or landmark region ^

Adding ways to bypass repetitive content lets keyboard users navigate the page more efficiently. [Learn more about bypass blocks](#).

- `<dl>`'s contain only properly-ordered `<dt>` and `<dd>` groups, `<script>`, `<template>` or `<div>` elements.

^

When definition lists are not properly marked up, screen readers may produce confusing or inaccurate output. [Learn how to structure definition lists correctly.](#)

- Definition list items are wrapped in `<dl>` elements

^

Definition list items (`<dt>` and `<dd>`) must be wrapped in a parent `<dl>` element to ensure that screen readers can properly announce them. [Learn how to structure definition lists correctly.](#)

- ARIA IDs are unique

^

The value of an ARIA ID must be unique to prevent other instances from being overlooked by assistive technologies. [Learn how to fix duplicate ARIA IDs.](#)

- No form fields have multiple labels

^

Form fields with multiple labels can be confusingly announced by assistive technologies like screen readers which use either the first, the last, or all of the labels. [Learn how to use form labels.](#)

- `<html>` element has an `[xml:lang]` attribute with the same base language as the `[lang]` attribute.

^

If the webpage does not specify a consistent language, then the screen reader might not announce the page's text correctly. [Learn more about the lang attribute.](#)

- Input buttons have discernible text.

^

Adding discernable and accessible text to input buttons may help screen reader users understand the purpose of the input button. [Learn more about input buttons.](#)

- `<input type="image">` elements have `[alt]` text

^

When an image is being used as an `<input>` button, providing alternative text can help screen reader users understand the purpose of the button. [Learn about input image alt text.](#)

- Form elements have associated labels

^

Labels ensure that form controls are announced properly by assistive technologies, like screen readers. [Learn more about form element labels.](#)

- The document does not use `<meta http-equiv="refresh">`

^

Users do not expect a page to refresh automatically, and doing so will move focus back to the top of the page. This may create a frustrating or confusing experience. [Learn more about the refresh meta tag.](#)

- `<object>` elements have alternate text

^

Screen readers cannot translate non-text content. Adding alternate text to `<object>` elements helps screen readers convey meaning to users. [Learn more about alt text for object elements.](#)

- Tables use `<caption>` instead of cells with the `[colspan]` attribute to indicate a caption.

^

Screen readers have features to make navigating tables easier. Ensuring that tables use the actual caption element instead of cells with the `[colspan]` attribute may improve the experience for screen reader users. [Learn more about captions.](#)

- `<td>` elements in a large `<table>` have one or more table headers.

^

Screen readers have features to make navigating tables easier. Ensuring that `<td>` elements in a large table (3 or more cells in width and height) have an associated table header may improve the experience for screen reader users. [Learn more about table headers.](#)

- Cells in a `<table>` element that use the `[headers]` attribute refer to table cells within the same table.

^

Screen readers have features to make navigating tables easier. Ensuring `<td>` cells using the `[headers]` attribute only refer to other cells in the same table may improve the experience for screen reader users. [Learn more about the headers attribute.](#)

- `<th>` elements and elements with `[role="columnheader"/"rowheader"]` have data cells they describe.

^

Screen readers have features to make navigating tables easier. Ensuring table headers always refer to some set of cells may improve the experience for screen reader users. [Learn more about table headers.](#)

- `[lang]` attributes have a valid value

^

Specifying a valid [BCP 47 language](#) on elements helps ensure that text is pronounced correctly by a screen reader. [Learn how to use the lang attribute.](#)



Best Practices

GENERAL

- ▲ Registers an `unload` listener

^

The `unload` event does not fire reliably and listening for it can prevent browser optimizations like the Back-Forward Cache. Use `pagehide` or `visibilitychange` events instead. [Learn more about unload event listeners](#)

Show 3rd-party resources (1)

Source

peopletalk.ru 1st Party

[core.js:3](#)

Yandex CDN Cdn

[yastatic.net/vas-bun.../inpage.bundle.js:1](#)

⚠ Browser errors were logged to the console

^

Errors logged to the console indicate unresolved problems. They can come from network request failures and other browser concerns. [Learn more about this errors in console diagnostic audit](#)

Show 3rd-party resources (2)

Source

Description

peopletalk.ru

1st Party

peopletalk.ru/:1 Access to XMLHttpRequest at 'https://ssp.afp.ai/api/adfox/bids' from origin 'https://peopletalk.ru' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource.

peopletalk.ru/:1 Access to XMLHttpRequest at 'https://ssp.afp.ai/api/adfox/bids' from origin 'https://peopletalk.ru' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource.

afp.ai

ssp.afp.ai/api/adfox/bids:1 Failed to load resource: net::ERR_FAILED

ssp.afp.ai/api/adfox/bids:1 Failed to load resource: net::ERR_FAILED

⚠ Issues were logged in the [Issues](#) panel in Chrome Devtools

^

Issues logged to the Issues panel in Chrome Devtools indicate unresolved problems. They can come from network request failures, insufficient security controls, and other browser concerns. Open up the Issues panel in Chrome DevTools for more details on each issue.

Issue type

Cookie

/peopletalk/monetizeme.js (cdn.monetize-me.ru)
...system/header-bidding.js (yandex.ru)
...system/context.js (yandex.ru)
/ssp/adfox (exchange.buzzoola.com)
/getcookie (matchid.adfox.yandex.ru)
...getBulk/v2?pr=... (yandex.ru)
/adfoxhb (ssp-rtb.sape.ru)
...getBulk/v2?pr=... (yandex.ru)
...meta/1430717?target-ref=... (yandex.ru)
/yhb (adx.videonow.ru)
/watch/39370120?vsid=42704d3...xVASx3020x169... (mc.yandex.ru)
...tracking/WUmejl_zO... (yandex.ru)
/watch/1430717?wmode=... (mc.yandex.ru)
/watch/1430717?wmode=... (mc.yandex.ru)
/watch/1430717?page-url=... (mc.yandex.ru)
...1430717/1?page-url=... (mc.yandex.ru)
/watch/1430717?page-url=... (mc.yandex.ru)
...1430717/1?page-url=... (mc.yandex.ru)
/event_confirmation (an.yandex.ru)
...1430717/1?page-url=... (mc.yandex.ru)
...getBulk/v2?pr=... (yandex.ru)
...getBulk/v2?pr=... (yandex.ru)
...getBulk/v2?pr=... (yandex.ru)
...getBulk/v2?pr=... (yandex.ru)
/metrika/tag.js (mc.yandex.ru)

Issue type

```
/watch/26812653?wmode=... (mc.yandex.ru)  
/watch/27493635?wmode=... (mc.yandex.ru)  
...26812653/1?page-url=... (mc.yandex.ru)  
...26812653/1?page-url=... (mc.yandex.ru)  
...26812653/1?page-url=... (mc.yandex.ru)  
...26812653/1?page-url=... (mc.yandex.ru)  
...27493635/1?page-url=... (mc.yandex.ru)  
...27493635/1?page-url=... (mc.yandex.ru)  
  
/webvisor/27493635?wmode=... (mc.yandex.ru)
```

Detected JavaScript libraries



All front-end JavaScript libraries detected on the page. [Learn more about this JavaScript library detection diagnostic audit.](#)

Name	Version
Bootstrap	4.3.1
jQuery	3.2.1
React	
Preact	10
Mustache	3.1.0
Vue	2.6.11
WordPress	
core-js	core-js-pure@2.6.12

TRUST AND SAFETY

Ensure CSP is effective against XSS attacks



A strong Content Security Policy (CSP) significantly reduces the risk of cross-site scripting (XSS) attacks. [Learn how to use a CSP to prevent XSS](#)

Description	Directive	Severity
No CSP found in enforcement mode		High

PASSED AUDITS (10)

Hide

● Uses HTTPS



All sites should be protected with HTTPS, even ones that don't handle sensitive data. This includes avoiding [mixed content](#), where some resources are loaded over HTTP despite the initial request being served over HTTPS. HTTPS prevents intruders from tampering with or passively listening in on the communications between your app and your users, and is a prerequisite for HTTP/2 and many new web platform APIs. [Learn more about HTTPS](#).

● Avoids requesting the geolocation permission on page load



Users are mistrustful of or confused by sites that request their location without context. Consider tying the request to a user action instead. [Learn more about the geolocation permission](#).

● Avoids requesting the notification permission on page load



Users are mistrustful of or confused by sites that request to send notifications without context. Consider tying the request to user gestures instead. [Learn more about responsibly getting permission for notifications](#).

● Allows users to paste into input fields



Preventing input pasting is a bad practice for the UX, and weakens security by blocking password managers. [Learn more about user-friendly input fields](#).

● Displays images with correct aspect ratio



Image display dimensions should match natural aspect ratio. [Learn more about image aspect ratio](#).

● Serves images with appropriate resolution



Image natural dimensions should be proportional to the display size and the pixel ratio to maximize image clarity. [Learn how to provide responsive images](#).

● Page has the HTML doctype



Specifying a doctype prevents the browser from switching to quirks-mode. [Learn more about the doctype declaration](#).

● Properly defines charset



A character encoding declaration is required. It can be done with a <meta> tag in the first 1024 bytes of the HTML or in the Content-Type HTTP response header. [Learn more about declaring the character encoding](#).

● **Avoids deprecated APIs**

^

Deprecated APIs will eventually be removed from the browser. [Learn more about deprecated APIs.](#)

● **Page has valid source maps**

^

Source maps translate minified code to the original source code. This helps developers debug in production. In addition, Lighthouse is able to provide further insights. Consider deploying source maps to take advantage of these benefits. [Learn more about source maps.](#)

Show 3rd-party resources (1)

URL	Map URL
Yandex CDN Cdn	
/share2/share.js (yastatic.net)	/share2/share.js.map (yastatic.net)
peopletalk.ru 1st Party	
...js/core.js?ver=169... (peopletalk.ru)	...js/core.js.map (peopletalk.ru)

NOT APPLICABLE (1)

Hide

○ **Fonts with `font-display: optional` are preloaded**

^

Preload optional fonts so first-time visitors may use them. [Learn more about preloading fonts](#)



SEO

These checks ensure that your page is following basic search engine optimization advice. There are many additional factors Lighthouse does not score here that may affect your search ranking, including performance on [Core Web Vitals](#). [Learn more about Google Search Essentials](#).

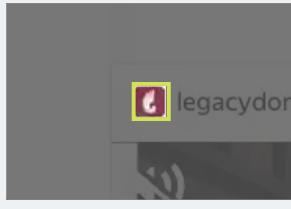
CONTENT BEST PRACTICES

▲ **Image elements do not have `[alt]` attributes**

^

Informative elements should aim for short, descriptive alternate text. Decorative elements can be ignored with an empty alt attribute. [Learn more about the alt attribute.](#)

Failing Elements



img.sd67f5606.h29430ea.a64e89e4b

Format your HTML in a way that enables crawlers to better understand your app's content.

ADDITIONAL ITEMS TO MANUALLY CHECK (1)

Hide

- Structured data is valid

Run the [Structured Data Testing Tool](#) and the [Structured Data Linter](#) to validate structured data. [Learn more about Structured Data](#).

Run these additional validators on your site to check additional SEO best practices.

PASSED AUDITS (11)

Hide

- Has a `<meta name="viewport">` tag with `width` or `initial-scale`

A `<meta name="viewport">` not only optimizes your app for mobile screen sizes, but also prevents [a 300 millisecond delay to user input](#). [Learn more about using the viewport meta tag.](#) (TBT)

- Document has a `<title>` element

The title gives screen reader users an overview of the page, and search engine users rely on it heavily to determine if a page is relevant to their search. [Learn more about document titles](#).

- Document has a meta description

Meta descriptions may be included in search results to concisely summarize page content. [Learn more about the meta description](#).

- Page has successful HTTP status code

Pages with unsuccessful HTTP status codes may not be indexed properly. [Learn more about HTTP status codes](#).

- Links have descriptive text

Descriptive link text helps search engines understand your content. [Learn how to make links more accessible](#).

- Links are crawlable

Search engines may use href attributes on links to crawl websites. Ensure that the href attribute of anchor elements links to an appropriate destination, so more pages of the site can be discovered. [Learn how to make links crawlable](#)

● Page isn't blocked from indexing ^

Search engines are unable to include your pages in search results if they don't have permission to crawl them. [Learn more about crawler directives](#).

● robots.txt is valid ^

If your robots.txt file is malformed, crawlers may not be able to understand how you want your website to be crawled or indexed. [Learn more about robots.txt](#).

● Document has a valid hreflang ^

hreflang links tell search engines what version of a page they should list in search results for a given language or region. [Learn more about hreflang](#).

● Document has a valid rel=canonical ^

Canonical links suggest which URL to show in search results. [Learn more about canonical links](#).

● Document avoids plugins ^

Search engines can't index plugin content, and many devices restrict plugins or don't support them. [Learn more about avoiding plugins](#).

NOT APPLICABLE (2) Hide

○ Document uses legible font sizes ^

Font sizes less than 12px are too small to be legible and require mobile visitors to “pinch to zoom” in order to read. Strive to have >60% of page text ≥12px. [Learn more about legible font sizes](#).

○ Tap targets are sized appropriately ^

Interactive elements like buttons and links should be large enough (48x48px), or have enough space around them, to be easy enough to tap without overlapping onto other elements. [Learn more about tap targets](#).



PWA

These checks validate the aspects of a Progressive Web App. [Learn what makes a good Progressive Web App.](#)

INSTALLABLE

- ▲ Web app manifest or service worker do not meet the installability requirements — 1 reason 

Service worker is the technology that enables your app to use many Progressive Web App features, such as offline, add to homescreen, and push notifications. With proper service worker and manifest implementations, browsers can proactively prompt users to add your app to their homescreen, which can lead to higher engagement. [Learn more about manifest installability requirements](#).

Failure reason

Page has no manifest <link> URL

PWA OPTIMIZED

- ▲ Does not register a service worker that controls page and `start_url` 

The service worker is the technology that enables your app to use many Progressive Web App features, such as offline, add to homescreen, and push notifications. [Learn more about Service Workers](#).

- ▲ Is not configured for a custom splash screen **Failures: No manifest was fetched.** 

A themed splash screen ensures a high-quality experience when users launch your app from their homescreens. [Learn more about splash screens](#).

- ▲ Does not set a theme color for the address bar.

Failures: No manifest was fetched, No `<meta name="theme-color">` tag found. 

The browser address bar can be themed to match your site. [Learn more about theming the address bar](#).

- Content is sized correctly for the viewport 

If the width of your app's content doesn't match the width of the viewport, your app might not be optimized for mobile screens. [Learn how to size content for the viewport](#).

- Has a `<meta name="viewport">` tag with `width` or `initial-scale` 

A `<meta name="viewport">` not only optimizes your app for mobile screen sizes, but also prevents [a 300 millisecond delay to user input](#). [Learn more about using the viewport meta tag](#). TBT

- ⚠ Manifest doesn't have a maskable icon [No manifest was fetched](#)

^

A maskable icon ensures that the image fills the entire shape without being letterboxed when installing the app on a device. [Learn about maskable manifest icons](#).

ADDITIONAL ITEMS TO MANUALLY CHECK (3)

Hide

- Site works cross-browser

^

To reach the most number of users, sites should work across every major browser. [Learn about cross-browser compatibility](#).

- Page transitions don't feel like they block on the network

^

Transitions should feel snappy as you tap around, even on a slow network. This experience is key to a user's perception of performance. [Learn more about page transitions](#).

- Each page has a URL

^

Ensure individual pages are deep linkable via URL and that URLs are unique for the purpose of shareability on social media. [Learn more about providing deep links](#).

These checks are required by the baseline [PWA Checklist](#) but are not automatically checked by Lighthouse. They do not affect your score but it's important that you verify them manually.



Publisher Ads

A Lighthouse plugin to improve ad speed and overall quality that is targeted at sites using GPT or AdSense tag. [Learn more](#)

PASSED AUDITS (1)

Hide

- Total ad JS blocking time — **0.0 s blocked**

^

Ad-related scripts are blocking the main thread. [Learn more](#).

NOT APPLICABLE (22)

Hide

- Tag load time — No tag requested

^

This metric measures the time for the ad tag's implementation script (pubads_impl.js for GPT; adsbygoogle.js for AdSense)

to load after the page loads. [Learn more](#).

○ First bid request time — No bids detected ^

This metric measures the elapsed time from the start of page load until the first bid request is made. Delayed bid requests will decrease impressions and viewability, and have a negative impact on ad revenue. [Learn More](#).

○ First ad request time — No ads requested ^

This metric measures the elapsed time from the start of page load until the first ad request is made. Delayed ad requests will decrease impressions and viewability, and have a negative impact on ad revenue. [Learn more](#).

○ Latency of first ad render — No ads rendered ^

This metric measures the time for the first ad iframe to render from page navigation. [Learn more](#).

○ Cumulative ad shift — No ads rendered ^

Measures layout shifts that were caused by ads or happened near ads. Reducing cumulative ad-related layout shift will improve user experience. [Learn more](#).

○ GPT and bids loaded in parallel — GPT not requested ^

To optimize ad loading, bid requests should not wait on GPT to load. This issue can often be fixed by making sure that bid requests do not wait on googletag.pubadsReady or googletag.cmd.push. [Learn More](#).

○ Header bidding is parallelized — No bids detected ^

Send header bidding requests simultaneously, rather than serially, to retrieve bids more quickly. [Learn more](#).

○ No bottleneck requests found — No ad-related requests ^

Speed up, load earlier, parallelize, or eliminate the following requests and their dependencies in order to speed up ad loading. [Learn More](#).

○ Ad scripts are loaded statically — No tag requested ^

Load the following scripts directly with `<script async src=...>` instead of injecting scripts with JavaScript. Doing so allows the browser to preload scripts sooner. [Learn more](#).

○ Ads not blocked by load events — No ad-related requests ^

Waiting on load events increases ad latency. To speed up ads, eliminate the following load event handlers. [Learn More](#).

○ Minimal render-blocking resources found — No tag requested ^

Render-blocking resources slow down tag load times. Consider loading critical JS/CSS inline or loading scripts asynchronously or loading the tag earlier in the head. [Learn more](#).

○ No long tasks blocking ad-related network requests — No tasks to compare ^

Tasks blocking the main thread can delay ad requests and cause a poor user experience. Consider removing long blocking tasks or moving them off of the main thread. These tasks can be especially detrimental to performance on less powerful devices. [Learn more](#).

○ Ad request waterfall — No ads requested ^

Consider reducing the number of resources, loading multiple resources simultaneously, or loading resources earlier to improve ad speed. Requests that block ad loading can be found below. [Learn more](#).

○ Few or no ads loaded outside viewport — No visible slots ^

Too many ads loaded outside the viewport lowers viewability rates and impacts user experience. Consider loading ads below the fold lazily as the user scrolls down. Consider using GPT's [Lazy Loading API](#). [Learn more](#).

○ Ad tag is loaded asynchronously — No tag requested ^

Loading the ad tag synchronously blocks content rendering until the tag is fetched and loaded. Consider using the `async` attribute to load gpt.js and/or adsbygoogle.js asynchronously. [Learn more](#).

○ Ad tag is loaded over HTTPS — No tag requested ^

For privacy and security, always load GPT/AdSense over HTTPS. Insecure pages should explicitly request the ad script securely. GPT Example: `<script async src="https://securepubads.g.doubleclick.net/tag/js/gpt.js">` AdSense Example: `<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js">`. [Learn more](#).

○ GPT tag is loaded from an official source — GPT not requested ^

Load GPT from 'securepubads.g.doubleclick.net' for standard integrations or from 'pagead2.googlesyndication.com' for limited ads. [Learn more](#).

○ Ads to page-height ratio is within recommended range — No visible slots ^

The ads to page-height ratio can impact user experience and ultimately user retention. [Learn more](#).

○ No ad found at the very top of the viewport — No visible slots ^

Over 10% of ads are never viewed because users scroll past them before they become viewable. By moving ad slots away from the very top of the viewport, users are more likely to see ads before scrolling away. [Learn more](#).

○ No duplicate tags found — No tags requested ^

Loading a tag more than once in the same page is redundant and adds overhead without benefit. [Learn more](#).

○ Deprecated GPT API Usage — GPT not requested ^

Deprecated GPT API methods should be avoided to ensure your page is tagged correctly. [Learn more](#).

○ GPT Errors — GPT not requested ^

Fix GPT errors to ensure your page is tagged as intended. [Learn more](#).

📅 Captured at Oct 23, 2023, 3:03

PM GMT+4

⌚ Initial page load

💻 Emulated Desktop with

Lighthouse 10.3.0

⚡ Custom throttling

🌐 Single page load

🌐 Using Chromium 116.0.0.0

with devtools

Generated by **Lighthouse** 10.3.0 | [File an issue](#)