

Defer unused CSS

Remove unused rules from stylesheets to reduce unnecessary bytes consumed by network activity. Learn more.

Efficiently encode images



Optimized images load faster and consume less cellular data. Learn more.

Serve images in next-gen formats



Image formats like JPEG 2000, JPEG XR, and WebP often provide better compression than PNG or JPEG, which means faster downloads and less data consumption. Learn more.

Enable text compression



Text-based responses should be served with compression (gzip, deflate or brotli) to minimize total network bytes. Learn more.

10 Avoid multiple, costly round trips to any origin

Potential savings of 0 ms <



Consider adding preconnect or dns-prefetch resource hints to establish early connections to important third-party origins. Learn more.

11 Keep server response times low (TTFB)



Time To First Byte identifies the time at which your server sends a response. Learn more.

12 Avoid multiple page redirects





Redirects introduce additional delays before the page can be loaded. Learn more.

13 Preload key requests





Consider using < link rel=preload> to prioritize fetching late-discovered resources sooner. Learn more.

14 Use video formats for animated content



Large GIFs are inefficient for delivering animated content. Consider using MPEG4/WebM videos for animations and PNG/WebP for static images instead of GIF to save network bytes. Learn more

Diagnostics



More information about the performance of your application.

15 Avoids enormous network payloads





Large network payloads cost users real money and are highly correlated with long load times. Learn more.

URL	Total Size	Transfe	er Ti	me
/z7/ (pas-gerc.pl)	1 KB		10	ms
16 Uses efficient cache policy on static assets		0 assets found	Ø	^
A long cache lifetime can speed up repeat visits to your page. Learn m	nore.			
17 Avoids an excessive DOM size		21 nodes	Ø	^

Browser engineers recommend pages contain fewer than ~1 500 DOM nodes. The sweet spot is a tree depth < 32 elements and fewer than 60 children/parent element. A large DOM can increase memory usage, cause longer style calculations, and produce costly layout reflows. Learn more.

Total DOM Nodes Maximum DOM Depth Maximum Children

Total DOM Nodes	Maximum DOM Depth	Maximum Children
21	5	9
		<form action=" " method="POST"></form>

18 Critical Request Chains



The Critical Request Chains below show you what resources are issued with a high priority. Consider reducing the length of chains, reducing the download size of resources, or deferring the download of unnecessary resources to improve page load. Learn more.

Maximum critical path latency: 50 ms

Initial Navigation

/z7/ (pas-gerc.pl) - 50 ms, 0.54 KB

19 User Timing marks and measures



Consider instrumenting your app with the User Timing API to create custom, real-world measurements of key user experiences. Learn more.

20 JavaScript boot-up time





Consider reducing the time spent parsing, compiling, and executing JS. You may find delivering smaller JS payloads helps with this. Learn more.

URL	Total	Script Evaluation	Script Parsing & Compilation
/polyfill.js (gighmmpiobklfepjocnamgkkbiglidom)	141 ms	128 ms	3 ms
/include.preload.js (gighmmpiobklfepjocnamgkkbiglidom)	56 ms	29 ms	17 ms

21 Minimizes main thread work





Consider reducing the time spent parsing, compiling and executing JS. You may find delivering smaller JS payloads helps with this.

Category	Time Spent
Script Evaluation	205 ms
Style & Layout	146 ms
Other	143 ms
Script Parsing & Compilation	45 ms
Parse HTML & CSS	27 ms
Garbage Collection	20 ms
Rendering	4 ms
22 All text remains visible during webfont loads	• ^

Leverage the font-display CSS feature to ensure text is user-visible while webfonts are loading. Learn more.

Progressive Web App

These checks validate the aspects of a Progressive Web App, as specified by the baseline PWA Checklist.



Page load is fast enough on 3G



A fast page load over a 3G network ensures a good mobile user experience. Learn more.

Does not respond with a 200 when offline 2



If you're building a Progressive Web App, consider using a service worker so that your app can work offline. Learn more.

User will not be prompted to Install the Web App 3



Failures: No manifest was fetched, Site does not register a service worker.

Browsers can proactively prompt users to add your app to their homescreen, which can lead to higher engagement. Learn more.

Does not use HTTPS





All sites should be protected with HTTPS, even ones that don't handle sensitive data. 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.

Insecure URL

/z7/ (pas-gerc.pl)

5 Does not redirect HTTP traffic to HTTPS



If you've already set up HTTPS, make sure that you redirect all HTTP traffic to HTTPS. Learn more.

Does not have a <meta name="viewport"> tag with width or initial-scale



No viewport meta tag found

Add a viewport meta tag to optimize your app for mobile screens. Learn more.

7 Does not register a service worker



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.

Contains some content when JavaScript is not available



Your app should display some content when JavaScript is disabled, even if it's just a warning to the user that JavaScript is required to use the app. Learn more.

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.

10 Address bar does not match brand colors





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.

11 Content is not sized correctly for the viewport



The viewport size is 725px, whereas the window size is 1280px.

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 more.

12 The short_name won't be truncated on the homescreen



Make your app's `short name` fewer than 12 characters to ensure that it's not truncated on homescreens. Learn more.

Additional items to manually check

3 audits ^

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.

Site works cross-browser

To reach the most number of users, sites should work across every major browser. Learn more.

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



Transitions should feel snappy as you tap around, even on a slow network, a key to perceived performance. Learn more.

Each page has a URL



Ensure individual pages are deep linkable via the URLs and that URLs are unique for the purpose of shareability on social media. Learn more.

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.

Elements Describe Contents Well

These are opportunities to make your content easier to understand for a user of assistive technology, like a screen reader.

The page does not contain a heading, skip link, or landmark region



Adding ways to bypass repetitive content lets keyboard users navigate the page more efficiently. Learn more.

Failing Elements

<html lang="pl">

Document doesn't have 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.

Failing Elements

<html lang="pl">

3 Form elements do not have associated labels



Labels ensure that form controls are announced properly by assistive technologies, like screen readers. Learn more.

Failing Elements

<input type="text" name="login">

<input type="password" name="haslo">

Meta Tags Used Properly

These are opportunities to improve the user experience of your site.

4 The document uses <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. <u>Learn more</u>.

Failing Elements

<meta http-equiv="refresh" content="20">

Q Additional items to manually check

10 audits ^

These items address areas which an automated testing tool cannot cover. Learn more in our guide on <u>conducting an accessibility review</u>.

1 The page has a logical tab order

^

Tabbing through the page follows the visual layout. Users cannot focus elements that are offscreen. Learn more.

2 Interactive controls are keyboard focusable

^

Custom interactive controls are keyboard focusable and display a focus indicator. Learn more.

3 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 more.

4 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 more.

5 Custom controls have associated labels

^

Custom interactive controls have associated labels, provided by aria-label or aria-labelledby. Learn more.

6 Custom controls have ARIA roles

7

^

Custom interactive controls have appropriate ARIA roles. Learn more.

^

Visual order on the page follows DOM order

- - -

DOM order matches the visual order, improving navigation for assistive technology. Learn more.

Offscreen content is hidden from assistive technology

Offscreen content is hidden with display: none or aria-hidden=true. <u>Learn more</u>.

^

Headings don't skip levels

Headings are used to create an outline for the page and heading levels are not skipped. Learn more.

10 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.

technology. Learn more. Passed audits 5 audits ^ **Elements Have Discernible Names** 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. Buttons 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 more. 2 Links 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 more. **Color Contrast Is Satisfactory** These are opportunities to improve the legibility of your content. 3 Background and foreground colors have a sufficient contrast ratio Low-contrast text is difficult or impossible for many users to read. Learn more. Page Specifies Valid Language These are opportunities to improve the interpretation of your content by users in different locales. 4 <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. 5 <html> element has a valid value for its [lang] attribute Specifying a valid BCP 47 language helps screen readers announce text properly. Learn more. Not applicable 26 audits ^ **Elements Use Attributes Correctly** These are opportunities to improve the configuration of your HTML elements. [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. 2 <audio> elements contain a <track> element with [kind="captions"]

Captions make audio elements usable for deaf or hearing-impaired users, providing critical information such as who is talking, what they're saying, and other non-speech information. <u>Learn more</u>.

3 Image elements have [alt] attributes

Informative elements should aim for short, descriptive alternate text. Decorative elements can be ignored with an empty alt attribute. <u>Learn more</u>.

4 <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 more.

5 No element has a [tabindex] value greater than 0

Ø .

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. <u>Learn more</u>.

6 Cells in a element that use the [headers] attribute only refer to other cells of that same table.



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

7 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. <u>Learn more</u>.

ARIA Attributes Follow Best Practices



These are opportunities to improve the usage of ARIA in your application which may enhance the experience for users of assistive technology, like a screen reader.

8 [aria-*] attributes match their roles



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

9 [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.

10 Elements with [role] that require specific children [role]s, are present



Some ARIA parent roles must contain specific child roles to perform their intended accessibility functions. <u>Learn more</u>.

11 [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.

12 [role] values are valid



ARIA roles must have valid values in order to perform their intended accessibility functions. Learn more.

13 [aria-*] attributes have valid values



Assistive technologies, like screen readers, can't interpret ARIA attributes with invalid values. Learn more.

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



Assistive technologies, like screen readers, can't interpret ARIA attributes with invalid names. Learn more.

Elements Are Well Structured



These are opportunities to make sure your HTML is appropriately structured.

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



When definition lists are not properly marked up, screen readers may produce confusing or inaccurate output. Learn more.

Lighthouse Report Viewer 16 Definition list items are wrapped in <d1> 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 more. 17 [id] attributes on the page are unique The value of an id attribute must be unique to prevent other instances from being overlooked by assistive technologies. Learn more. 18 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. 19 List items () are contained within or parent elements Screen readers require list items ('') to be contained within a parent '' or '' to be announced properly. Learn more. **Elements Describe Contents Well** These are opportunities to make your content easier to understand for a user of assistive technology, like a screen reader. 20 <frame> or <iframe> elements have a title Screen reader users rely on frame titles to describe the contents of frames. Learn more. 21 Presentational elements avoid using , <caption> or the [summary] attribute. A table being used for layout purposes should not include data elements, such as the th or caption elements or the summary attribute, because this can create a confusing experience for screen reader users. Learn more. 22 <object> elements have [alt] text Screen readers cannot translate non-text content. Adding alt text to `<object>` elements helps screen readers convey meaning to users. Learn more. 23 <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. 24 <video> elements contain a <track> element with [kind="description"] Audio descriptions provide relevant information for videos that dialogue cannot, such as facial expressions and

Meta Tags Used Properly

scenes. Learn more.

These are opportunities to improve the user experience of your site.

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



contents of a web page. Learn more.

Page Specifies Valid Language

These are opportunities to improve the interpretation of your content by users in different locales.

26 [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 more.

Best Practices



Does not use HTTPS

1 insecure request found A



All sites should be protected with HTTPS, even ones that don't handle sensitive data. 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.

Insecure URL

/z7/ (pas-gerc.pl)

Does not use HTTP/2 for all of its resources

1 request not served via HTTP/2 A ^



HTTP/2 offers many benefits over HTTP/1.1, including binary headers, multiplexing, and server push, Learn

	more.	erver pusn. <u>Learn</u>
UR	L	Protocol
/z7	/ (pas-gerc.pl)	http/1.1
✓	Passed audits	13 audits ^
1	Avoids Application Cache	
	Application Cache is deprecated. <u>Learn more</u> .	
2	Avoids WebSQL DB	• ^
	Web SQL is deprecated. Consider using IndexedDB instead. Learn more.	
3	Uses passive listeners to improve scrolling performance	• ^
	Consider marking your touch and wheel event listeners as `passive` to improve your page's Learn more.	s scroll performance.
4	Avoids document.write()	• ^
	For users on slow connections, external scripts dynamically injected via `document.write()` by tens of seconds. <u>Learn more</u> .	can delay page load
5	Links to cross-origin destinations are safe	• ^
	Add `rel="noopener"` or `rel="noreferrer"` to any external links to improve performance and vulnerabilities. <u>Learn more</u> .	prevent security
6	Avoids requesting the geolocation permission on page load	• ^
	Users are mistrustful of or confused by sites that request their location without context. Cor	nsider tying the

Page has the HTML doctype

request to user gestures instead. Learn more.

Specifying a doctype prevents the browser from switching to quirks-mode.Read more on the MDN Web Docs page

B Avoids front-end JavaScript libraries with known security vulnerabilities



Some third-party scripts may contain known security vulnerabilities that are easily identified and exploited by attackers. Learn more.

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. <u>Learn more</u>.

10 Avoids deprecated APIs



Deprecated APIs will eventually be removed from the browser. Learn more.

11 Allows users to paste into password fields



Preventing password pasting undermines good security policy. Learn more.

12 No browser errors logged to the console



Errors logged to the console indicate unresolved problems. They can come from network request failures and other browser concerns.

13 Displays images with correct aspect ratio



Image display dimensions should match natural aspect ratio. Learn more.

SEO

2



These checks ensure that your page is optimized for search engine results ranking. There are additional factors Lighthouse does not check that may affect your search ranking. <u>Learn more</u>.

Mobile Friendly

Make sure your pages are mobile friendly so users don't have to pinch or zoom in order to read the content pages. **Learn more**.

1 Does not have a <meta name="viewport"> tag with width or initial-scale



No viewport meta tag found

Add a viewport meta tag to optimize your app for mobile screens. Learn more.

Document doesn't use legible font sizes



Text is illegible because of a missing viewport config

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. <u>Learn more</u>.

Content Best Practices

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

3 Document doesn't have 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. <u>Learn more</u>.

Failing Elements

Failing Elements <html lang="pl"> Document does not have a meta description Meta descriptions may be included in search results to concisely summarize page content. Learn more. Additional items to manually check 2 audits ^ Run these additional validators on your site to check additional SEO best practices. Page is mobile friendly Take the Mobile-Friendly Test to check for audits not covered by Lighthouse, like sizing tap targets appropriately. Learn more. Structured data is valid Run the Structured Data Testing Tool and the Structured Data Linter to validate structured data. Learn more. ✓ Passed audits 5 audits ^ **Crawling and Indexing** To appear in search results, crawlers need access to your app. Page has successful HTTP status code Pages with unsuccessful HTTP status codes may not be indexed properly. Learn more. Page isn't blocked from indexing 2 Search engines are unable to include your pages in search results if they don't have permission to crawl them. Learn more. **Content Best Practices** Format your HTML in a way that enables crawlers to better understand your app's content. Links have descriptive text Descriptive link text helps search engines understand your content. Learn more. 4 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. Document avoids plugins Search engines can't index plugin content, and many devices restrict plugins or don't support them. Learn more. Not applicable 2 audits ^ Crawling and Indexing To appear in search results, crawlers need access to your app. 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. **Content Best Practices**

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

2 Document has a valid rel=canonical



Canonical links suggest which URL to show in search results. Learn more.

Runtime settings

- URL: http://pas-gerc.pl/z7/
- Fetch time: Nov 27, 2018, 8:39 PM GMT+1
- **Device**: No emulation
- Network throttling: 150 ms TCP RTT, 1,638.4 Kbps throughput (Simulated)
- CPU throttling: 4x slowdown (Simulated)
- User agent (host): Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.110 Safari/537.36

Generated by Lighthouse 3.0.3 | File an issue