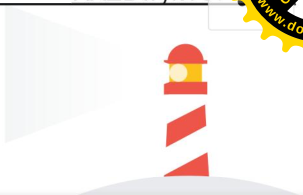
http://michchol.pl/Z7/index.php?fbclid=IwAR1bThqjPCU58CZvKkGUaM3zN9NZY9vxHOfEXnlZnxrDRXZDwyk1UP

28, 2018, 4:57 PM GMT+1

No emulation, Simulated Slow 4G network



100 Performance

31 Progressive Web App

65 Accessibility

80 Best Practices

67 SEO

Score scale: 90-100 50-89 0-49

Performance

100

Metrics

First Contentful Paint	770 ms	First Meaningful Paint	770 ms
Speed Index	770 ms	First CPU Idle	1 070 ms
Time to Interactive	1 130 ms	Estimated Input Latency	19 ms

Values are estimated and may vary.

Opportunities

These are opportunities to speed up your application by optimizing the following resources.

	Opportunity	Estimated Savings
1	Eliminate render-blocking resources	0.16 s

Resources are blocking the first paint of your page. Consider delivering critical JS/CSS inline and deferring all non-critical JS/styles. Learn more.

URL	Size (KB)	Download Time (ms)
/Z7/style.css (michchol.pl)	0 KB	160 ms

Diagnostics

More information about the performance of your application.

1 Critical Request Chains 1 chains found

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

Warning: Results may not display properly. Report was created with an earlier version of Lighthouse (3.0.3). The latest version is 4.0.0-alpha.1.

Initial Navigation

/Z7/index.php?fbclid=IwAR1bThq… (michchol.pl)

/Z7/style.css (michchol.pl) **- 90 ms, 0.18 KB**

Passed audits 20 audits

Opportunities

These are opportunities to speed up your application by optimizing the following resources.

1 Properly size images

Serve images that are appropriately-sized to save cellular data and improve load time. Learn more.

2 Defer offscreen images

Consider lazy-loading offscreen and hidden images after all critical resources have finished loading to lower time to interactive. Learn more.

3 Minify CSS

Minifying CSS files can reduce network payload sizes. Learn more.

4 Minify JavaScript

Minifying JavaScript files can reduce payload sizes and script parse time. Learn more.

5 Defer unused CSS

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

6 Efficiently encode images

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

7 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.

8 Enable text compression

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

9 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.

10 Keep server response times low (TTFB)

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

11 Avoid multiple page redirects 0 ms

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

12 Preload key requests Potential savings of 0 ms

Warning: Results may not display properly. Report was created with an earlier version of Lighthouse (3.0.3). The latest version is 4.0.0-alpha.1.

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

Diagnostics

More information about the performance of your application.

14 Avoids enormous network payloads — Total size was 1 KB

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

URL	Total Size	Transfer Time
/Z7/index.php?fbclid=IwAR1bThq… (michchol.pl)	1 KB	10 ms
/Z7/style.css (michchol.pl)	0 KB	0 ms

15 Uses efficient cache policy on static assets — 0 assets found

A long cache lifetime can speed up repeat visits to your page. Learn more.

16 Avoids an excessive DOM size — 28 nodes

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
28	6	10
	`<br>`	`<form action="zaloguj.php" method="post">`

17 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.

18 JavaScript boot-up time — 270 ms

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
//gighmmpiobklfepjocnamgkkbiglidom/polyfill.js ()	274 ms	261 ms	4 ms

19 Minimizes main thread work — 700 ms

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	344 ms

Warning: Results may not display properly. Report was created with an earlier version of Lighthouse (3.0.3). The latest version is 4.0.0-alpha.1.

Category	Time Spent
Script Parsing & Compilation	49 ms
Parse HTML & CSS	34 ms
Garbage Collection	15 ms
Rendering	3 ms

20 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

31

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

1 Page load is fast enough on 3G

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

2 Does not respond with a 200 when offline

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

3 User will not be prompted to Install the Web App

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.

4 Does not use HTTPS 2 insecure requests found

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/index.php?fbclid=IwAR1bThq… (michchol.pl)
/Z7/style.css (michchol.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.

6 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

Warning: Results may not display properly. Report was created with an earlier version of Lighthouse (3.0.3). The latest version is 4.0.0-alpha.1.

8 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.

9 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.

1 Site works cross-browser

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

2 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.

3 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

65

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.

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

Warning: Results may not display properly. Report was created with an earlier version of Lighthouse (3.0.3). The latest version is 4.0.0-alpha.1.

Failing Elements

```
<html lang="pl">
```

2 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">
```

Additional items to manually check 10 audits

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

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

Custom interactive controls have appropriate ARIA roles. Learn more.

7 Visual order on the page follows DOM order

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

8 Offscreen content is hidden from assistive technology

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

9 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

Warning: Results may not display properly. Report was created with an earlier version of Lighthouse (3.0.3). The latest version is 4.0.0-alpha.1.

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.

1 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.

Elements Describe Contents Well

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

4 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.

Elements Are Well Structured

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

5 `[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.

Page Specifies Valid Language

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

6 `<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.

7 `<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.

1 `[accesskey]` values are unique

Warning: Results may not display properly. Report was created with an earlier version of Lighthouse (3.0.3). The latest version is 4.0.0-alpha.1.

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. Learn more.

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. Learn more.

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

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.

6 Cells in a `<table>` 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 `<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.

7 `<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.

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. Learn more.

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.

Warning: Results may not display properly. Report was created with an earlier version of Lighthouse (3.0.3). The latest version is 4.0.0-alpha.1.

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.

16 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 more.

17 Lists contain only `<li>` 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.

18 List items (`<li>`) are contained within `<ul>` or `<ol>` parent elements

Screen readers require list items (`<li>`) to be contained within a parent `<ul>` or `<ol>` 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.

19 `<frame>` or `<iframe>` elements have a title

Screen reader users rely on frame titles to describe the contents of frames. Learn more.

20 Presentational `<table>` elements avoid using `<th>`, `<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.

21 `<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.

22 `<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.

23 `<video>` elements contain a `<track>` element with `[kind="description"]`

Audio descriptions provide relevant information for videos that dialogue cannot, such as facial expressions and scenes. Learn more.

Meta Tags Used Properly

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

Warning: Results may not display properly. Report was created with an earlier version of Lighthouse (3.0.3). The latest version is 4.0.0-alpha.1.

25 `[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.

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

80

1 Does not use HTTPS — 2 insecure requests found

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/index.php?fbclid=IwAR1bThq… (michchol.pl)
/Z7/style.css (michchol.pl)

2 Does not use HTTP/2 for all of its resources — 2 requests not served via HTTP/2

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

URL	Protocol
/Z7/index.php?fbclid=IwAR1bThq… (michchol.pl)	http/1.1
/Z7/style.css (michchol.pl)	http/1.1

3 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.

URL	Description
/Z7/style.css (michchol.pl)	`Failed to load resource: the server responded with a status of 404 (Not Found)`
/Z7/style.css (michchol.pl)	`Failed to load resource: the server responded with a status of 404 (Not Found)`

Warning: Results may not display properly. Report was created with an earlier version of Lighthouse (3.0.3). The latest version is 4.0.0-alpha.1.

1 Avoids Application Cache

Application Cache is deprecated. Learn more.

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 scroll performance. Learn more.

4 Avoids `document.write()`

For users on slow connections, external scripts dynamically injected via `document.write()` can delay page load by tens of seconds. Learn more.

5 Links to cross-origin destinations are safe

Add `rel="noopener"` or `rel="noreferrer"` to any external links to improve performance and prevent security vulnerabilities. Learn more.

6 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 user gestures instead. Learn more.

7 Page has the HTML doctype

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

8 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.

9 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.

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 Displays images with correct aspect ratio

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

SEO

67

These checks ensure that your page is optimized for search engine results ranking. There are additional

Warning: Results may not display properly. Report was created with an earlier version of Lighthouse (3.0.3). The latest version is 4.0.0-alpha.1.

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.

2 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. Learn more.

Content Best Practices

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

3 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.

1 Page is mobile friendly

Take the Mobile-Friendly Test to check for audits not covered by Lighthouse, like sizing tap targets appropriately. Learn more.

2 Structured data is valid

Run the Structured Data Testing Tool and the Structured Data Linter to validate structured data. Learn more.

Passed audits 6 audits

Content Best Practices

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

1 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.

2 Links have descriptive text

Descriptive link text helps search engines understand your content. Learn more.

3 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.

4 Document avoids plugins

Search engines can't index plugin content, and many devices restrict plugins or don't support them. Learn more.

Crawling and Indexing

To appear in search results, crawlers need access to your app.

Warning: Results may not display properly. Report was created with an earlier version of Lighthouse (3.0.3). The latest version is 4.0.0-alpha.1.

6 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.

Not applicable 2 audits

Crawling and Indexing

To appear in search results, crawlers need access to your app.

1 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://michchol.pl/Z7/index.php?fbclid=IwAR1bThqjPCU58CZvKkGUaM3zN9NZY9vxHOfEXnlZnxrDRXZDwyk1UPI592k`
- **Fetch time:** `Nov 28, 2018, 4:57 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

Warning: Results may not display properly. Report was created with an earlier version of Lighthouse (3.0.3). The latest version is 4.0.0-alpha.1.