Masters of Computer Applications

CAP 785

Submitted to

LOVELY PROFESSIONAL UNIVERSITY

PHAGWARA, PUNJAB

WEB PERFORMANCE OPTIMIZATION

Course Code: 785

Submitted By:

Name: Naveen Sony

Registration Number:12210854

Submitted To:

Mr. Mohit Mishra Sir

UID- 31132

Q1) Prepare report of hosted websites

Scope of Websites

Introduction:

where we showcase our passion for web design and development through a collection of our finest projects. From sleek and modern designs to robust and functional websites, we take pride in crafting digital experiences that captivate and inspire.

Portfolio Showcase:

Organize your portfolio into categories or projects for easy navigation. For each project, include a detailed description highlighting the problem, your approach, and the solution.

Incorporate high-quality images, videos, or other media to visually showcase your work. If possible, include metrics or statistics to demonstrate the impact of your projects.

Services Offered:

Clearly outline the range of services you provide. Describe each service in detail, including the benefits and outcomes clients can expect. If applicable, include pricing information or a call-to-action for inquiries.

Client Testimonials:

Display testimonials prominently on your website. Include a mix of testimonials from different types of clients or employers. Use real names and, if possible, include photos or company logos for authenticity.

About Me/Us:

Provide a compelling narrative about your background, experience, and expertise. Share personal anecdotes or stories that illustrate your journey and passion for your work. Include professional achievements, certifications, or awards that lend credibility.

Contact Information:

Create a dedicated contact page with a contact form for easy communication. Display your email address, phone number, and physical address (if applicable) prominently. Consider adding links to your social media profiles for additional points of contact.

Responsive Design:

Ensure that your website layout adjusts seamlessly across different devices and screen sizes. Test your website on various browsers and devices to ensure consistent performance.

SEO Optimization:

Conduct keyword research to identify relevant terms for your industry. Optimize on-page elements such as titles, headings, and meta descriptions. Create high-quality, valuable content that attracts organic traffic.

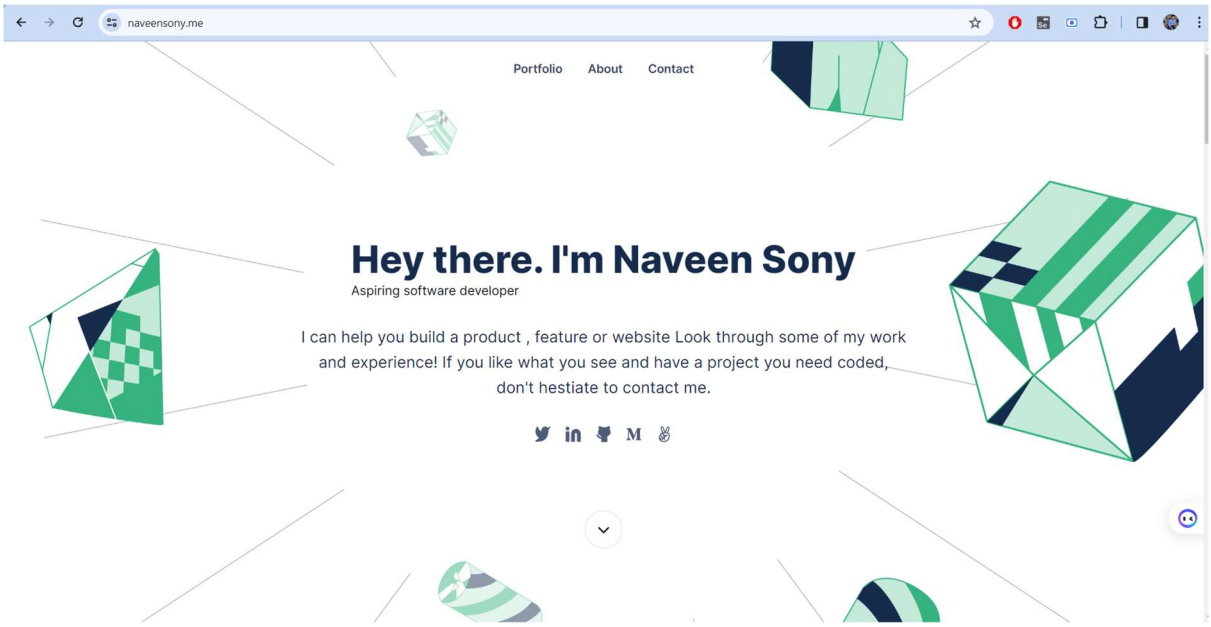
Analytics:

Integrate tools like Google Analytics to track website traffic, user behaviour, and conversions. Use data insights to refine your website and marketing strategies.

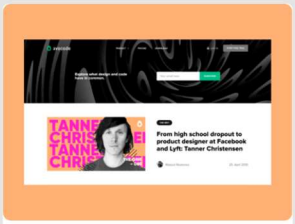
Security:

Implement SSL/TLS encryption to secure data transmission. Keep your website's software, plugins, and themes updated to prevent vulnerabilities. Use strong passwords and consider adding additional security measures such as firewalls or security plugins.

1) Minimum 8 Pages websites containing at least 20 different images.



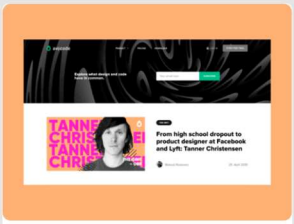
My Recent Works



Multi-Post Stories 1

Ruby on rails | css | JavaScript | html

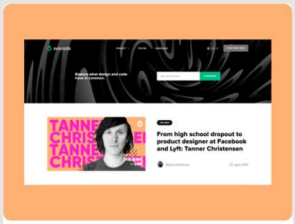
See Project



Multi-Post Stories Gain+Glory

Ruby on rails | css | JavaScript | html

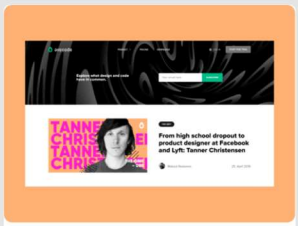
See Project



Multi-Post Stories Gain+Glory

Ruby on rails | css | JavaScript | html

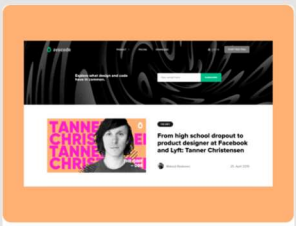
See Project



Multi-Post Stories Gain+Glory

Ruby on rails | css | JavaScript | html

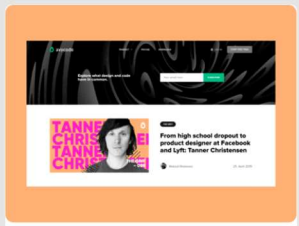
See Project



Multi-Post Stories Gain+Glory

Ruby on rails | css | JavaScript | html

See Project



Multi-Post Stories Gain+Glory

Ruby on rails | css | JavaScript | html

See Project

About me

I am a logical and organized Web Developer with a strong foundation in HTML, CSS, JavaScript, C++, MySQL, and Java. I possess the ability to develop innovative and realistic web applications. As a team member, I contribute effectively and thrive in collaborative environments. I have completed an internship as a Java Developer at UPTEC Computer Consultancy Ltd, gaining valuable experience in software development. I am currently pursuing a Master's degree in Computer Application with a focus on Information Technology from Lovely Professional University. Additionally, I hold certifications in web development, object-oriented programming in Java, and SQL. I have been recognized for my participation and achievement in sports competitions. Seeking an entry-level position with a reputed IT company, I am ready to contribute my skills and dedication to deliver exceptional results.

Get My Resume

Get My Resume

Languages

C C++ Python Java JavaScript

Frameworks

Bootstrap Jquery ReactJs Selenium

Skills

HTML CSS JavaScript ReactJs Linux Netwok GitHub

I'm always interested in hearing about new projects, so if you'd like to chat please get in touch.

Naveen Sony

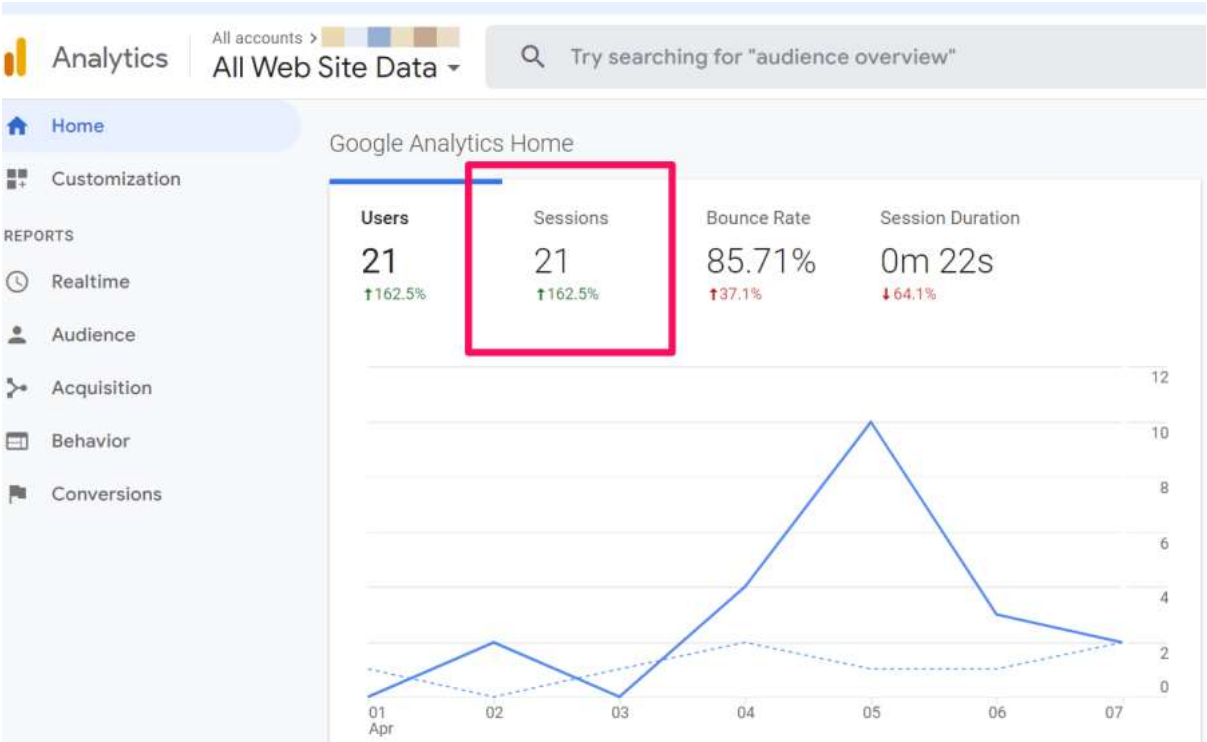
naveensoni0900@gmail.com

Typing something

Get in touch

2) Traffic on Websites should be there

Website traffic indicates the number of visitors accessing a website, which is crucial for businesses and individuals looking to reach a wider audience, increase brand visibility, generate leads, and drive conversions. Strategies such as search engine optimization (SEO), content marketing, social media promotion, and online advertising can help increase website traffic and attract more visitors.



Q2) Improve your websites by adopting following strategies

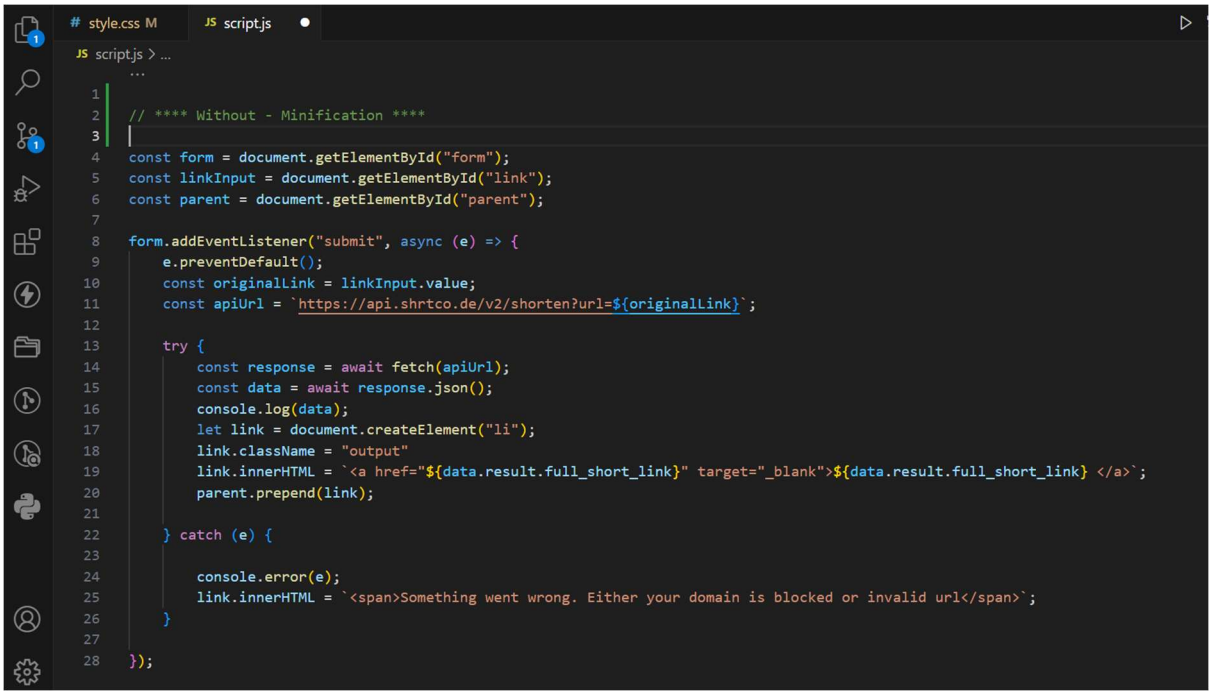
1) Minification

Minification is the process of reducing the size of code files by removing unnecessary characters without altering their functionality. This technique is commonly used in web development to optimize the performance of websites and applications.

Minification aims to decrease the size of code files, such as HTML, CSS, and JavaScript, transmitted over the internet. By removing unnecessary characters

like white spaces, comments, and line breaks, minification reduces file sizes, leading to faster downloads and improved website performance.

```css
}
/* ***** Before Manification ******** */
input {       naveensony07, 7 months ago • create url shortner project
    font-size: 15px;
    border-radius: 20px;
    border: 5px solid rgb(101, 5, 5);
    width: 100%;
    padding: 12px 20px;
    margin: 8px 0;
    box-sizing: border-box;

}
/* ****** After Manification ****** */
input {font-size: 15px;border-radius: 20px;border: 5px solid rgb(101, 5, 5);
    width: 100%;padding: 12px 20px;margin: 8px 0;box-sizing: border-box;}
```



2) UNCSS

UNCSS is a tool used to remove unused CSS (Cascading Style Sheets) from your web project's stylesheets. It stands for "Unused CSS."

Unused CSS: Over time, as you develop a website, your CSS files can accumulate styles that are no longer used. This can happen due to refactoring, changing design requirements, or removing features. However, these unused styles still add to the file size of your CSS, increasing page load times unnecessarily.

How UNCSS Works: UNCSS analyses your HTML files and identifies which CSS selectors are being used. It does this by parsing your HTML files and searching for matching selectors in your CSS files. Any selectors that are not found in your HTML files are considered unused.

Command to Install UNCSS

npm install -g uncss

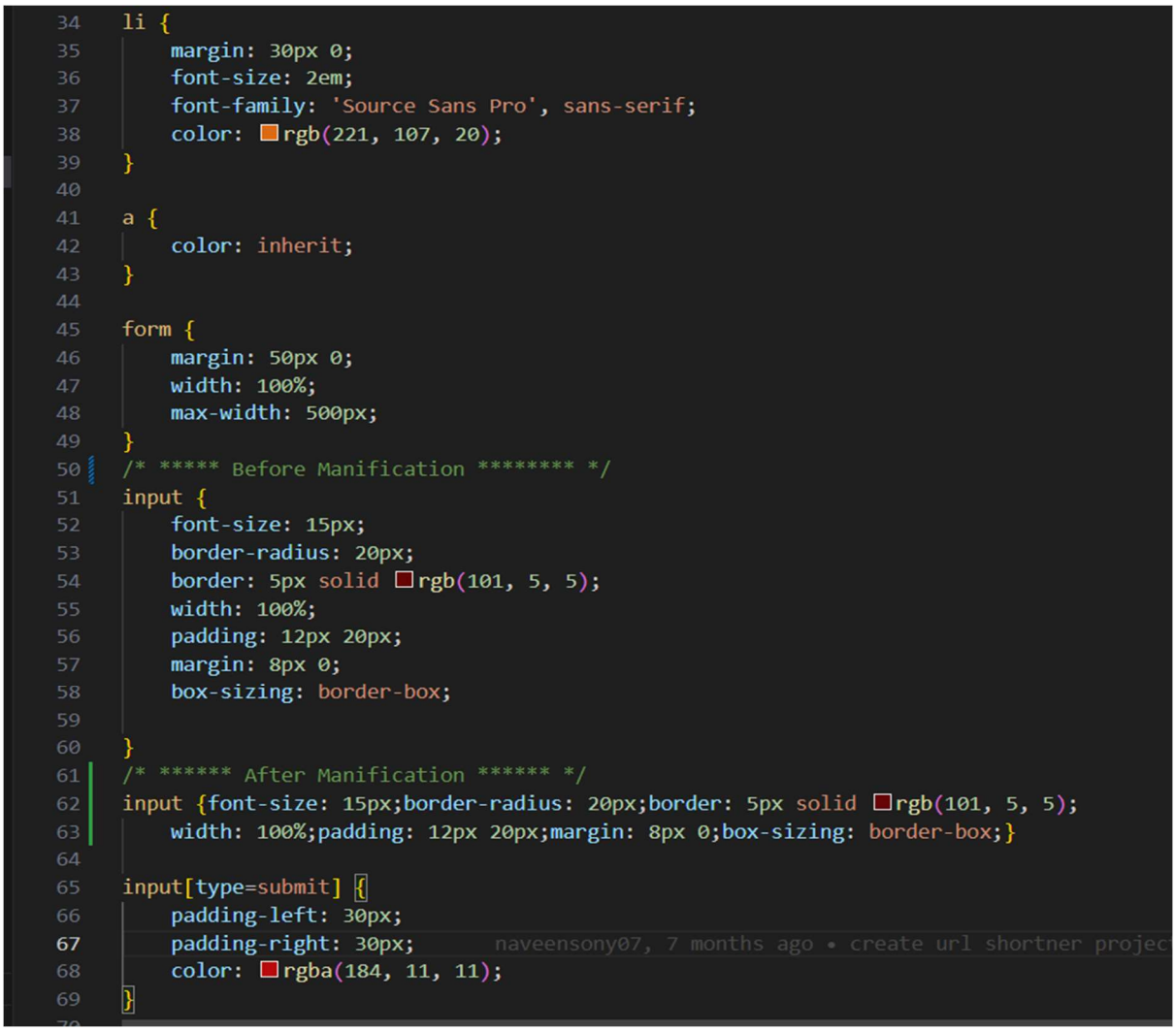
index.html > html > body > div.wrapper

naveensony07, 7 months ago | 1 author (naveensony07)

```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>URL Shortener</title>
    <link rel="stylesheet" href="style.css">
</head>

<body>
    <div class="wrapper">
        <h1>Welcome! URL Shortner</h1>
      naveensony07, 7 months ago • create url shortner project
        <form id="form">
            <input id="link" type="text" placeholder="Enter or Paste Your Long Link">
            <input type="submit" value="Shorten URL !">
        </form>

        <ul id="parent" class="shortLinks">
        </ul>
    </div>

    <script src="script.js"></script>

</body>

</html>
```

```
...
```

```css
@import url('https://fonts.googleapis.com/css2?family=Source+Sans+Pro:wght@900&display=swap');

* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;

}

body {

    display: flex;
    justify-content: center;
    align-items: center;
    min-height: 100vh;
}

.wrapper {
    width: 100%;
    min-height: 100vh;
    display: flex;
    align-items: center;
    flex-direction: column;
    background-image: url(image/background.jpeg);
    padding: 50px 0;
}

ul {
    list-style-type: none;

}
```



```
li {
    margin: 30px 0;
    font-size: 2em;
    font-family: 'Source Sans Pro', sans-serif;
    color: rgb(221, 107, 20);
}

a {
    color: inherit;
}

form {
    margin: 50px 0;
    width: 100%;
    max-width: 500px;
}
/* ***** Before Manification ******** */
input {
    font-size: 15px;
    border-radius: 20px;
    border: 5px solid rgb(101, 5, 5);
    width: 100%;
    padding: 12px 20px;
    margin: 8px 0;
    box-sizing: border-box;

}
/* ****** After Manification ****** */
input {font-size: 15px;border-radius: 20px;border: 5px solid rgb(101, 5, 5);
    width: 100%;padding: 12px 20px;margin: 8px 0;box-sizing: border-box;}

input[type=submit] {
    padding-left: 30px;
    padding-right: 30px;      naveensony07, 7 months ago • create url shortner projec
    color: rgba(184, 11, 11);
}
```

```css
input:focus {
    outline: none;
}

h1 {
    font-size: 50px;
    font-family: 'Source Sans Pro', sans-serif;
    color: rgb(184, 11, 11);
    font-weight: 900;
    font-size: 5em;
    text-transform: uppercase;

}

/* **** */
div.main.div.content div.listContainer .contentList{

    height: 350px;
    height: 250px;
    padding: 12px;
    margin-right: 12px;
    float: left;
    display: inline;
    list-style: none;
}


/* ****** Longhand Properties *******  */
.h1{
    font-family: "Times New Roman";
    font-size: 3px;
    font-style: bold;
    font-display: initial;
    font-weight: 100px;

}
```

Remove Un-used CSS

```css
/* ***************** Remove Unused CSS ************* */

@import
url('https://fonts.googleapis.com/css2?family=Source+Sans+Pro:wght@900&display=swap');

* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
```

```
}

body {

    display: flex;
    justify-content: center;
    align-items: center;
    min-height: 100vh;
}

.wrapper {
    width: 100%;
    min-height: 100vh;
    display: flex;
    align-items: center;
    flex-direction: column;
    background-image: url(image/background.jpeg);
    padding: 50px 0;
}

ul {
    list-style-type: none;

}

form {
    margin: 50px 0;
    width: 100%;
    max-width: 500px;
}
/* ***** Before Manification ******** */
input {
    font-size: 15px;
    border-radius: 20px;
    border: 5px solid rgb(101, 5, 5);
    width: 100%;
    padding: 12px 20px;
    margin: 8px 0;
    box-sizing: border-box;

}
/* ****** After Manification ****** */
input {font-size: 15px;border-radius: 20px;border: 5px solid rgb(101, 5, 5);
```

```
    width: 100%;padding: 12px 20px;margin: 8px 0;box-sizing: border-box;}

input[type=submit] {
    padding-left: 30px;
    padding-right: 30px;
    color: rgba(184, 11, 11);
}

input:focus {
    outline: none;
}

h1 {
    font-size: 50px;
    font-family: 'Source Sans Pro', sans-serif;
    color: rgb(184, 11, 11);
    font-weight: 900;
    font-size: 5em;
    text-transform: uppercase;

}

/* **** */


/* ****** Longhand Properties ********  */


/* ****** Shorthand Properties ******** */
```

3) Shorthand CSS Properties

Shorthand CSS properties allow you to specify multiple related properties using a single declaration. They can make your CSS more concise and easier to write, but they require an understanding of the syntax and potential implications.

What are Shorthand CSS Properties?

Shorthand CSS properties enable you to set multiple related CSS properties in a single declaration. Instead of writing out each individual property separately, you can combine them into one shorthand declaration.

```
96
97   /* ****** Longhand Properties ********  */
98   .h1{
99       font-family: "Times New Roman";
100      font-size: 3px;
101      font-style: bold;
102      font-display: initial;
103      font-weight: 100px;
104
105  }
106
107
```

Convert Longhand into Shorthand Properties

```
08
09
10  /* ****** Shorthand Properties ******** */
11
12  .h1 {
13
14      font: bold 100px/normal 3px "Times New Roman";
15  }
16
17
```

4) Shallow CSS

"Shallow CSS" is not a widely recognized term in the context of web development or CSS. However, we can interpret "shallow" CSS in a manner like "scoped" CSS, which focuses on containing styles within a specific component or module.

In this interpretation, shallow CSS would involve styling elements within a component in a way that confines the styles' scope to that component only. This concept aligns with the principles of modular design and encapsulation, where each component is responsible for its own appearance and behaviour, minimizing the risk of style conflicts and making it easier to manage and maintain the codebase.

```
div.main.div.content div.listContainer .genericList{

    height: 350px;
    height: 250px;
    padding: 12px;
    margin-right: 12px;
    float: left;
    display: inline;
    list-style: none;
}
```

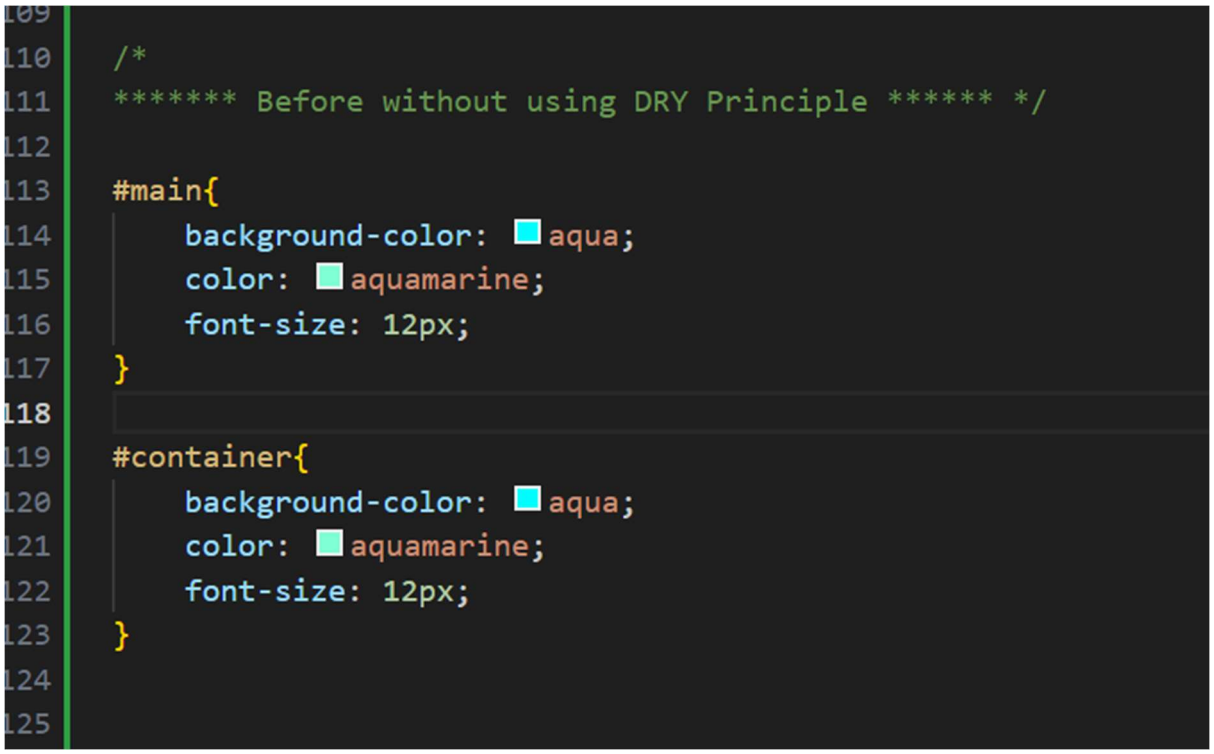
After Shallow CSS 82% smaller

```
.contentList{

    height: 350px;
    height: 250px;
    padding: 12px;
    margin-right: 12px;
    float: left;
    display: inline;
    list-style: none;
}
```

5) Apply DRY principle

The "DRY" principle, which stands for "Don't Repeat Yourself," is a fundamental concept in software development and web development. It emphasizes the importance of avoiding redundancy and duplication in code. The principle suggests that every piece of knowledge or logic within a system should have a single, unambiguous representation.

Avoid Repetition: When writing code, it is essential to identify patterns or repetitions and refactor them into reusable components, functions, or modules. Instead of duplicating code in multiple places, extract common functionality into a single source and reference it wherever needed.

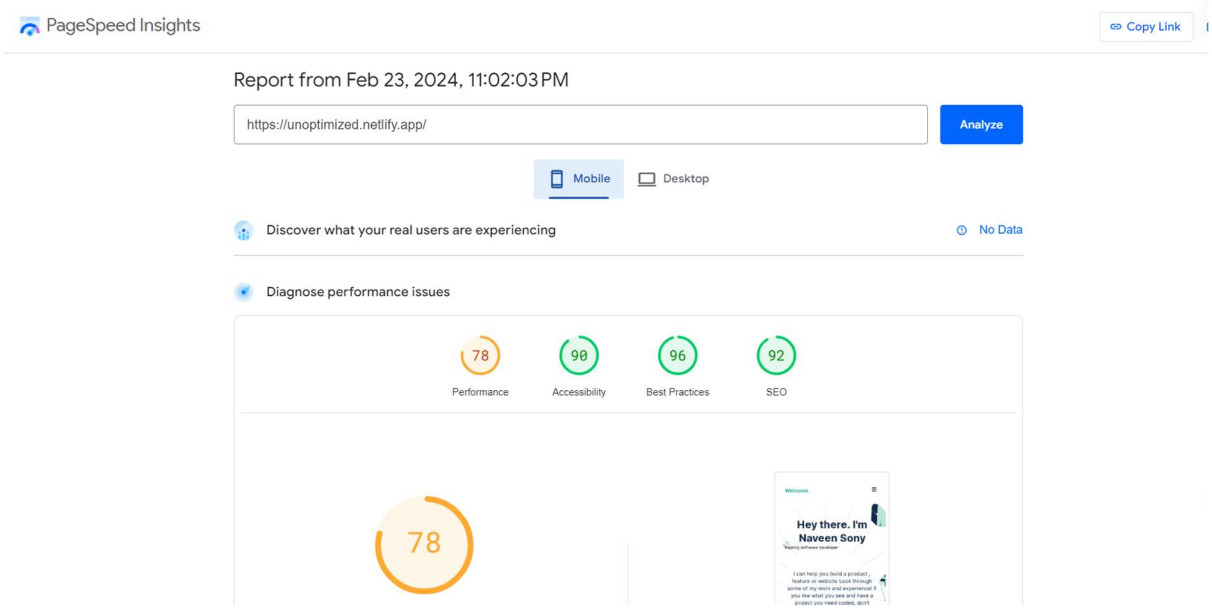


```
125
126  /* ***** Apply DRY Principle ****** */
127
128  #main,#container{
129
130      background-color: aqua;
131      color: aquamarine;
132      font-size: 12px;
133  }
134
135
```

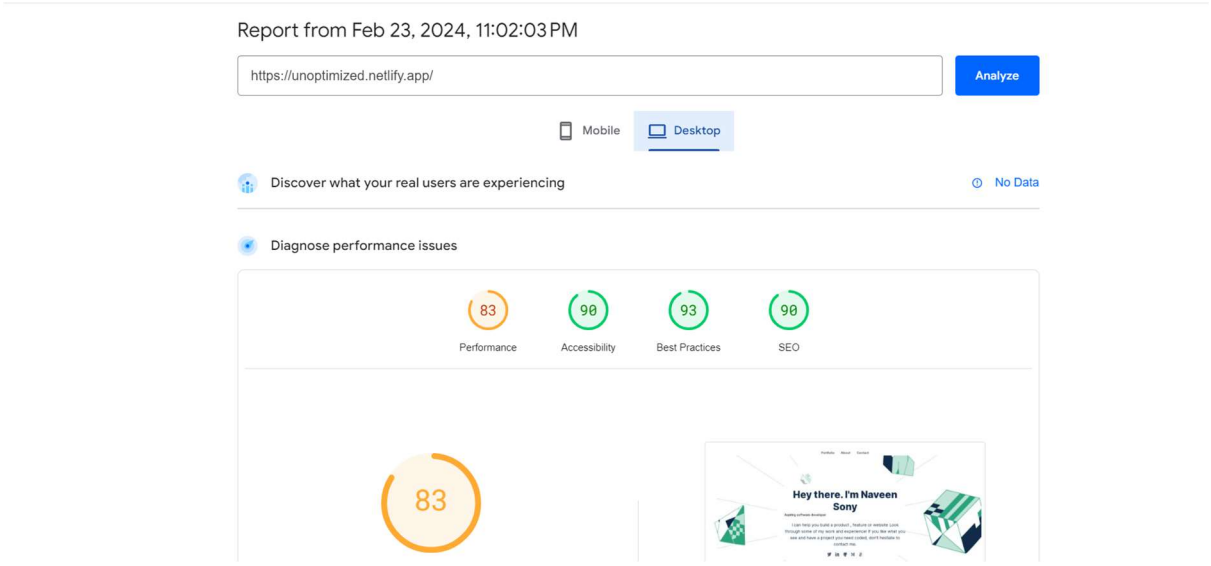
Q3) Comparative Analysis of websites (Unoptimized Vs Optimized)

1. Test the websites performance using any two-testing platform

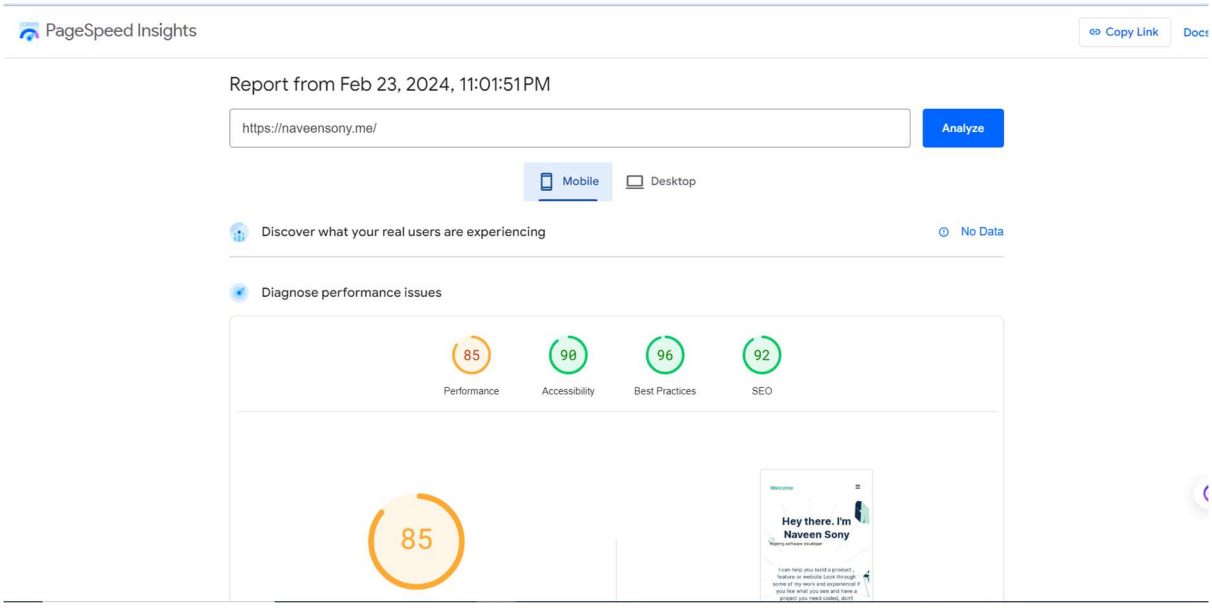
Unoptimized Websites (Mobile)



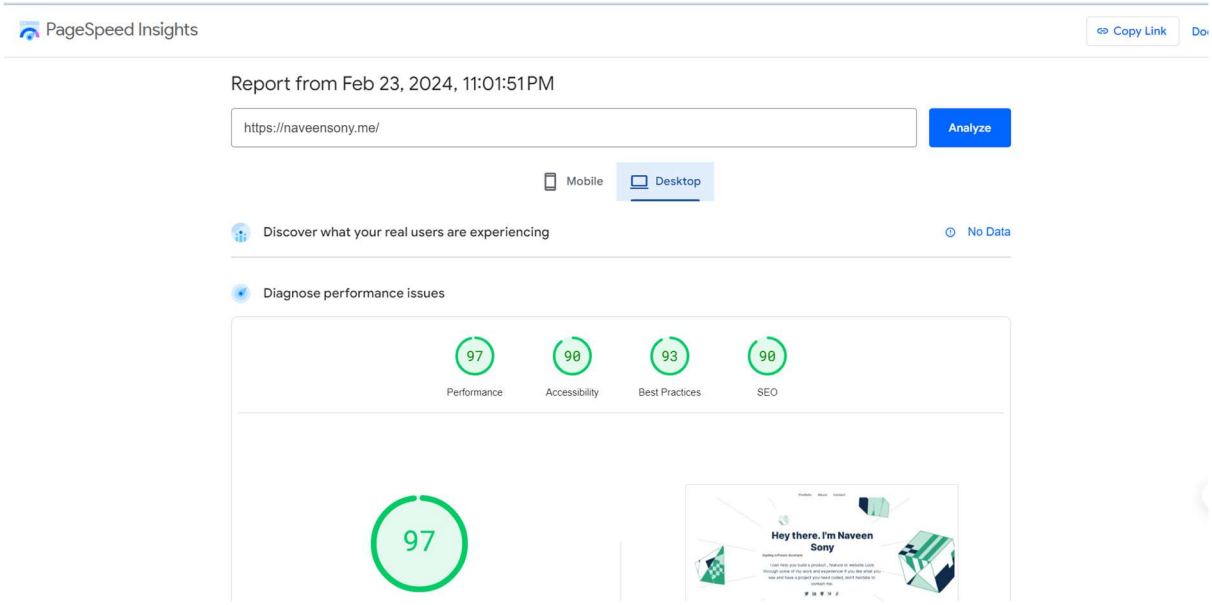
Unoptimized Websites (Desktop)



Optimized Websites (Mobile)



Unoptimized Websites (Desktop)



2. **Test and Analyse it on Various performance metrics i.e. FCP, LCP, TFB, Full Load Time, waiting time etc (Use the screenshot to show the performance parameter optimization.**

FCP (First Contentful Paint):

FCP measures the time it takes for the first piece of content to be rendered on the screen when a user navigates to a web page. This content could be text, images, or other elements. FCP is an important user-centric metric as it indicates when the user perceives that the page is starting to load.

LCP (Largest Contentful Paint):

LCP measures the render time of the largest content element visible within the viewport. This could be an image, video, or block-level element. LCP is a key metric for assessing the perceived loading speed of a web page because it indicates when the most significant content is fully rendered and visible to the user.

TFB (Time to First Byte):

TFB measures the time it takes for a user's browser to receive the first byte of response from the web server after making a request. It represents the server's initial response time and includes the time taken for processing the request, generating the response, and transmitting it back to the client. TFB is crucial for assessing server performance and optimizing backend processes.

Full Load Time:

Full Load Time, also known as Page Load Time or Load Time, refers to the total time it takes for all resources on a web page (HTML, CSS, JavaScript, images, etc.) to be fully loaded and rendered in the user's browser. It encompasses all stages of the loading process, including network latency, server response time, and client-side rendering. Full Load Time is a comprehensive metric that reflects the overall speed and responsiveness of a web page.

Waiting Time:

Waiting Time, also referred to as Time to Start Render or Time to First Paint, measures the time between the user initiating a page request (e.g., clicking a link or entering a URL) and the browser beginning to render the page content. Waiting Time includes the time spent on DNS resolution, TCP connection, and SSL negotiation before the browser can start fetching resources and rendering the page. It is an important component of overall page load time and directly impacts user perception of website speed**.**

PageSpeed Insights

Copy Link Docs

Report from Feb 23, 2024, 10:58:51 PM

https://unoptimized.netlify.app/

Analyze

Discover what your real users are experiencing

No Data

Diagnose performance issues

Performance	Accessibility	Best Practices	SEO
78	90	96	92

78

Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See

Mobile Desktop

0–49 50–89 90–100

METRICS Expand view

First Contentful Paint	Largest Contentful Paint
3.9 s	3.9 s

PageSpeed Insights

Copy Link Docs

Speed Index

4.4 s

Captured at Feb 23, 2024, 10:58 PM GMT+5:30

Emulated Moto G Power with Lighthouse 11.4.0

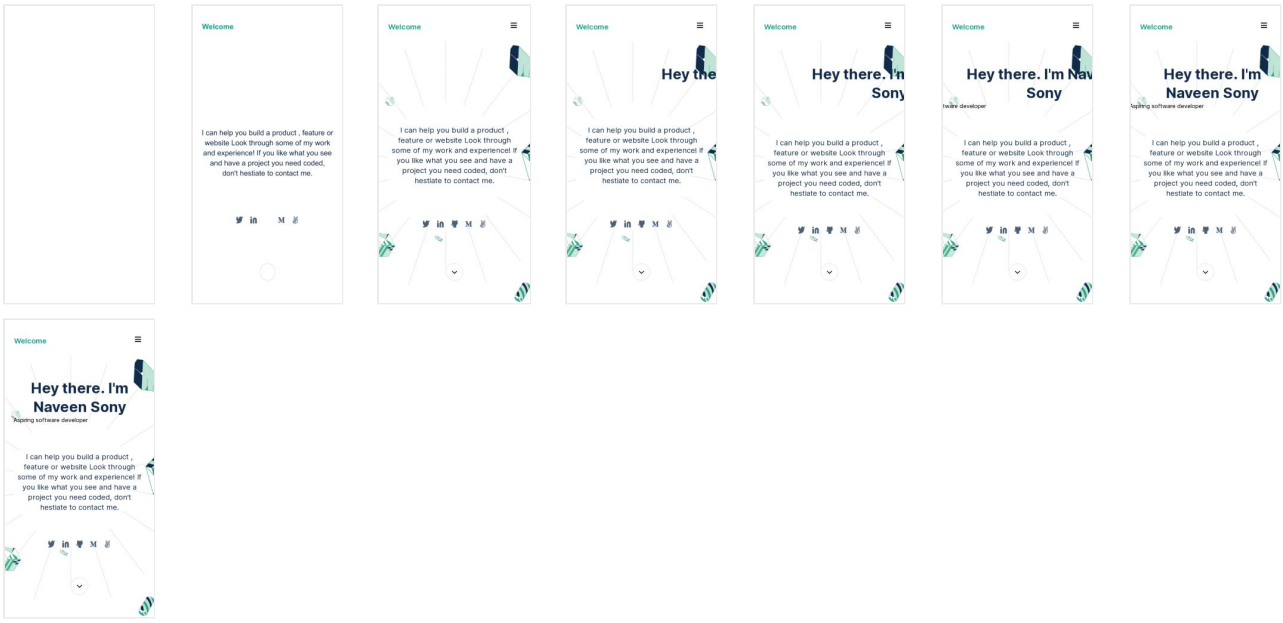
Single page session

Initial page load

Slow 4G throttling

Using HeadlessChromium 121.0.6167.139 with lr

View Treemap



Show audits relevant to: All FCP LCP TBT CLS

DIAGNOSTICS

- Eliminate render-blocking resources — Potential savings of 2,200 ms
- Largest Contentful Paint element — 3,860 ms
- Serve images in next-gen formats — Potential savings of 169 KiB
- Preload Largest Contentful Paint image
- Image elements do not have explicit `width` and `height`
- Properly size images — Potential savings of 236 KiB
- Reduce unused CSS — Potential savings of 19 KiB
- Initial server response time was short — Root document took 210 ms

PageSpeed Insights Copy Link Docs

- Avoids enormous network payloads — Total size was 565 KiB
- Avoids an excessive DOM size — 155 elements
- Avoid chaining critical requests — 3 chains found
- JavaScript execution time — 0.0 s
- Minimizes main-thread work — 1.6 s
- Minimize third-party usage — Third-party code blocked the main thread for 0 ms

More information about the performance of your application. These numbers don't directly affect the Performance score.

PASSED AUDITS (21) Show

Accessibility

These checks highlight opportunities to improve the accessibility of your web app. Automatic detection can only detect a subset of issues and does not guarantee the accessibility of your web app, so manual testing is also encouraged.

CONTRAST

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

These are opportunities to improve the legibility of your content.

NAMES AND LABELS

PageSpeed Insights Copy Link Docs

users of assistive technology, like a screen reader.

ADDITIONAL ITEMS TO MANUALLY CHECK (10) Show

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

PASSED AUDITS (16) Show

NOT APPLICABLE (43) Show

Best Practices

GENERAL

▲ Browser errors were logged to the console

TRUST AND SAFETY

○ Ensure CSP is effective against XSS attacks

PASSED AUDITS (13) Show

PageSpeed Insights

Copy Link Docs

92

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

▲ Document does not have a meta description

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

ADDITIONAL ITEMS TO MANUALLY CHECK (1) Show

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

PASSED AUDITS (11) Show

NOT APPLICABLE (2) Show

More on PageSpeed Insights

What's new

Related Content

Updates

Connect

Twitter

PageSpeed Insights

Copy Link Docs

Ask questions on Stack Overflow Podcasts

Mailing list

Google for Developers Chrome Firebase All products

Terms and Privacy Policy

For details, see the Google Developers Site Policies.

PageSpeed Insights

Copy Link Docs

Report from Feb 23, 2024, 10:52:26 PM

https://naveensony.me/ Analyze

Mobile Desktop

Discover what your real users are experiencing

No Data

Diagnose performance issues

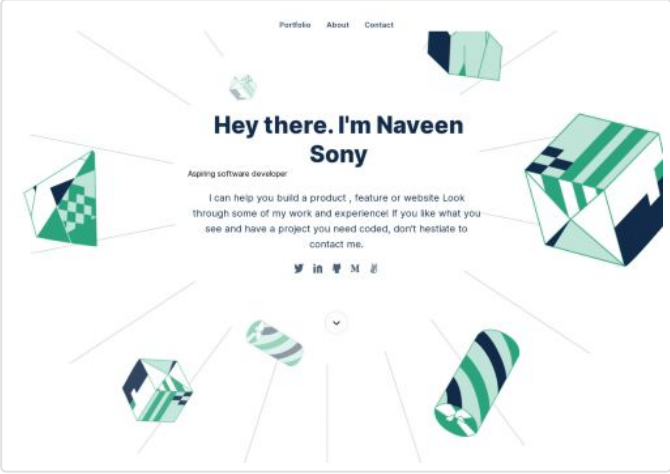
Performance	Accessibility	Best Practices	SEO
94	90	93	90

94

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
1.1 s

Largest Contentful Paint
1.2 s

PageSpeed Insights Copy Link Docs

Speed Index

1.2 s

Captured at Feb 23, 2024, 10:52 PM GMT+5:30

Emulated Desktop with Lighthouse 11.4.0

Single page session

Initial page load

Custom throttling

Using HeadlessChromium 121.0.6167.139 with lr

View Treemap

Show audits relevant to: All FCP LCP TBT CLS

DIAGNOSTICS

- Eliminate render-blocking resources — Potential savings of 500 ms
- Serve images in next-gen formats — Potential savings of 193 KiB
- Reduce unused CSS — Potential savings of 19 KiB
- Preload Largest Contentful Paint image
- Serve static assets with an efficient cache policy — 1 resource found
- Image elements do not have explicit `width` and `height`
- Properly size images — Potential savings of 31 KiB
- Avoid large layout shifts — 8 elements found
- Initial server response time was short — Root document took 100 ms
- Avoid non-composited animations — 2 animated elements found
- Avoids enormous network payloads — Total size was 568 KiB

PageSpeed Insights Copy Link Docs

- JavaScript execution time — 0.0 s
- Minimizes main-thread work — 0.5 s
- Minimize third-party usage — Third-party code blocked the main thread for 0 ms
- Largest Contentful Paint element — 1,200 ms

More information about the performance of your application. These numbers don't directly affect the Performance score.

PASSED AUDITS (20) Show

Accessibility

These checks highlight opportunities to improve the accessibility of your web app. Automatic detection can only detect a subset of issues and does not guarantee the accessibility of your web app, so manual testing is also encouraged.

CONTRAST

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

These are opportunities to improve the legibility of your content.

NAMES AND LABELS

- ▲ Links do not have a discernible name

PageSpeed Insights Copy Link Docs

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

PASSED AUDITS (16) Show

NOT APPLICABLE (43) Show

Best Practices

USER EXPERIENCE

▲ Displays images with incorrect aspect ratio

GENERAL

▲ Browser errors were logged to the console

TRUST AND SAFETY

○ Ensure CSP is effective against XSS attacks

PASSED AUDITS (12) Show

PageSpeed Insights

Copy Link Docs

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

▲ Document does not have a meta description

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

ADDITIONAL ITEMS TO MANUALLY CHECK (1)

Show

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

PASSED AUDITS (9)

Show

NOT APPLICABLE (4)

Show

More on PageSpeed Insights Related Content Connect

PageSpeed Insights

Copy Link Docs

Learn about Web Performance

Ask questions on Stack Overflow

Mailing list

Case Studies

Podcasts

Google for Developers Chrome Firebase All products

Terms and Privacy Policy

For details, see the Google Developers Site Policies.