dareboost

Web performance
and quality report

https://www.inflibnet.ac.in/ilms/

This report is provided by Dareboost, an online tool for web performance and quality analysis and monitoring.

Table of contents

Summary

82%

5 Issues

7 Improvements

78 Successes
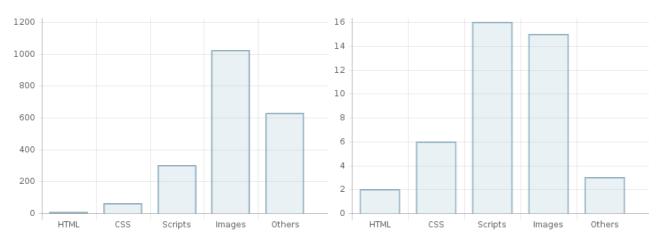
SIMULATED VISITOR: Chrome   Washington DC   10.0/2.0Mbps (Latency: 28 ms)

First Byte	Start Render	Fully Loaded	Speed Index
1.02 sec	2.60 sec	4.76 sec	3773 ms

Weight by resources type

	HTML	CSS	Scripts	Images	Others

Requests by resources type

	HTML	CSS	Scripts	Images	Others

Detected technologies

Apache   CodeIgniter   OWL Carousel   Twitter Bootstrap   jQuery

PHP

Tips and best practices

Accessibility

Did you know?

#72

ℹ️ **No <noscript> tag is detected**

When a web page uses scripts, it is advised to set at least one `noscript` tag. It is required to display a message when JavaScript is disabled by the user.

```
                                                                              Example
<script  type="text/javascript">
   document.write('Hello World!')
</script>
<noscript>Your browser does not support JavaScript!</noscript>
```

Well done, these best practices are respected

100/100

#2569

✅ **No empty element detected**

Some tags, such as `<p>` , `<li>` , `<button>` , `<legend>` , `<caption>` , `<figcaption>` and `<quote>` elements are not allowed to be empty. Empty tag make it difficult for some screen readers to understand the page.

Either add `aria-hidden` attribute to the empty element or remove it from your page entirely.

```
                                                                              Example
<p aria-hidden="true"></p>
```

100/100

#2334

✅ **This page defines a lang**

Perfect. Your page defines a `lang` attribute that will allow screen readers to correctly understand your website.

100/100

#2395

✅ **No 'refresh' <meta> tag**

It is not recommended to automatically force a page refresh. This behavior disturbs users who are not able to control it.

This page does not use `meta refresh` tag. This is a best practice.

100/100

#2410

✅ **No empty "src" attribute detected**

The "src" attributes link the current page to other resources. There is no justification for using them with empty values.

All your `src` attributes precise a target. That's a best practice.

100/100

#2423

✅ **You specify a consistent label on your links**

A link is more attractive if the text describes what is behind it. You can also take the opportunity to use keywords in these texts, to improve your page's SEO.

Describe the link in your `<a>` tag, rather than indicating the link itself. Example:
`<a href="http://mylink.com/">My description</a>`

The content of your `<a>` tags are different from the link itself. That's a best practice.

100/100

#2468

✅ **The main title of the page is the first stated title**

Facilitate the work of screen readers by indicating your main title (`<h1>`) before any other title.

We recommend you to keep a coherent hierarchy among your titles (h1, then h2, then h3, etc).

100/100

#2523

✅ **All labels refer to an element**

The `for` attribute associates the label to another element of the page and help screen readers to better interpret your content.

Label and for attribute

A label describes an element (a text to fill, a checkbox, etc.). When a user clicks on a label associated with a radio button, the option will be directly selected, improving the user experience.

How to use a label?

Associate the label to an element of the page by indicating the ID of the element. Example:

```
<form action="/action">            Example
  <label for="myId">
  <input type="radio" name="myOptions" id="myId" value="1" >
```

Well done, these best practices are respected

100/100

#2566

✅ Your HTML response is not too heavy

Why reduce the code amount of a page?

Before a web page can be displayed, the browser must, among other things, download it, parse it and model it into a document that can be understood by the rendering engine. If the amount of code contained in the page is too large, these steps are slowed down and the rendering is delayed.

How to reduce the amount of code?

Your HTML response should contain only the information that is immediately necessary to display the visible area of the page. Move inline information to external files (JS for scripts, CSS for styles, asynchronous queries for additional content) and simplify the HTML structure of your page.

100/100

#2575

✅ No Mutation Events detected in your scripts

To capture DOM events, do not use Mutation Events. Alternatives exist.

Good concept, bad implementation

When developing complex JavaScript applications, you may need to know when the DOM node tree has changed. Introduced in 2000 in the DOM, Level 2 specification to provide a solution to this need, Mutations Events are browser-initiated events that let you know when a DOM node is added, removed, or deleted.

Mutation Events, however, present major performance problems. First, they are synchronous, i.e. they prevent other events in the queue from being fired (if those events are used to update the UI, this will cause some lag). Second, they are implemented as browser events, thus traverse the DOM tree from the targeted HTML element to the parent element which listens for the event, clogging the JavaScript thread along the way.

Mutation Events have been deprecated in 2016 in the DOM, Level 3 specification.

Mutation Observers to the rescue

If you need to watch for changes being made to the DOM tree, you should use the `MutationObserver` interface (DOM4 Living Standard). Mutation Observers are asynchronous, processed in batches, and observe specific or all changes to a node. They are more efficient in terms of CPU usage than browser events and therefore cause fewer to no UI freeze.

Learn how to use Mutation Observers (Mozilla Developer Network).

100/100

#2353

✅ Your JavaScript resources don't block your page loading

JavaScript can significantly slow down a page display, especially if it is necessary to download an external script.

Defer the use of JavaScript as much as possible to provide a faster start for the page display.

How can I fix this?

First of all, distinguish what portions of your JS is critical and must be loaded as soon as possible, and put them in a specific external file. Keep this file as streamlined as possible, and defer the parsing or execution of all other JS files.

Use one of the methods below to defer parsing for external JavaScript files:

- use the async attribute;
- use the defer attribute;
- append the script to the DOM in JavaScript during the onload event;
- make sure your scripts are placed at the bottom of the page (ideally at the end of the body).

100/100

✅ **You are using persistent connections**

In HTTP/1.1, each resource is downloaded over a dedicated connection (TCP). Some headers can indicate to the browser that it can reuse the same connection multiple times, which improves performance.

The HTTP Keep-Alive headers

The `Connection` and `Keep-Alive` HTTTP headers inform the browser how it can keep a TCP connection (enabled by default with HTTP 1.1) open and reuse it for future requests.

For these persistent connections to be enabled, the `Connection` header must be set to `keep-alive`. The `Keep-Alive` header must indicate the minimum time during which the connection must remain open and the maximum number of requests to answer to. For example:
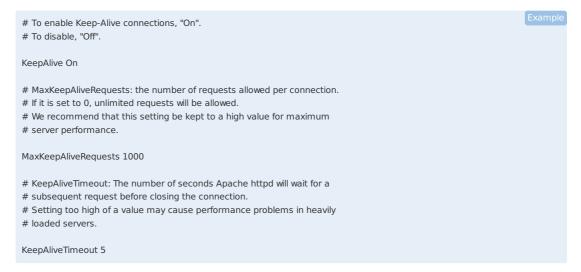
```
Connection: Keep-Alive                                          Example
Keep-Alive: timeout=5, max=1000
```

indicates to the browser that the connection must remain open for at least 5 seconds and answer, at most, 1000 requests.

🔖 **On an Apache server, you can enable both headers from your.htaccess file, defining the connection as `keep-alive`.**

```
<IfModule mod_headers.c>                                        Example
    Header set Connection keep-alive
</IfModule>
```

If you have access to the Apache configuration, you can more precisely define the different values (note: you will have to restart the web server).

```
# To enable Keep-Alive connections, "On".                      Example
# To disable, "Off".

KeepAlive On

# MaxKeepAliveRequests: the number of requests allowed per connection.
# If it is set to 0, unlimited requests will be allowed.
# We recommend that this setting be kept to a high value for maximum
# server performance.

MaxKeepAliveRequests 1000

# KeepAliveTimeout: The number of seconds Apache httpd will wait for a
# subsequent request before closing the connection.
# Setting too high of a value may cause performance problems in heavily
# loaded servers.

KeepAliveTimeout 5
```

100/100

✅ **You specify a character set in the response HTTP Header**

Specify the character set used in the **Content-Type** HTTP header allows the browser to parse immediately the page.

When the browser receives bytes from your server, it needs to identify the collection of letters and symbols that were used in writing the text that was converted into these bytes, and the encoding used for this conversion, in order to reverse it. If no information of this kind has been transmitted, the browser will try to find recognizable patterns within the bytes to determine the encoding itself, and eventually try some common charsets, which will take time, delaying further processing of the page.

How to specify a charset in the Content-Type header?

In the following explanation, we will consider UTF-8 as the targeted character set but please remember that the character set declared in your Content-Type HTTP Header must reflect the character set used to encode the file, which may not be UTF-8.

On **Apache 2.2+**, the configuration of UTF-8 as a default character set for your `text/plain` and `text/html` files involves the `AddDefaultCharset` directive:

```
AddDefaultCharset utf-8                                    Example
```

For other types of files, you'll need the `AddCharset` directive:

```
AddCharset utf-8 .js .css ...                             Example
```

On **nginx**, you'll need to make sure that the `ngx_http_charset_module` is loaded, then use the `charset` directive.

```
charset utf-8;                                            Example
```

Here too, it is possible to refine the scope so that other types of files than text/html are delivered in utf-8, using the directive charset_types:

```
charset_types text/html text/css application/javascript   Example
```

100/100

✅ **You do not use CSS @import**

Using CSS @import allows to add external stylesheet. In fact, browsers cannot download them at the same time, this may add a delay to the rendering of the page. It is better to use the `link` tag. See more information.

100/100

✅ **You do not use a useless "http-equiv" <meta> tag**

HTTP headers are more efficient than the http-equiv meta tags.

The <meta http-equiv=""/> tags

The `http-equiv` meta tags provide the web browser with information similar to HTTP headers. For example, defining the meta <meta http-equiv="content-type"/> is the same as sending the HTTP Content-Type header.

There are two disadvantages in using http-equiv meta tags:

- Going through the meta requires to interpret the beginning of the HTML page, which is slower than going through the HTTP headers in terms of performance
- If the HTTP header is already present, the meta is ignored

In which cases are the <meta http-equiv=""/> useful?

Only one case can justify the presence of these meta tags: if you don't have access to the configuration of your server, and that is to say to the HTTP headers.

However, we recommend that you use a configurable server so that you can establish the most efficient site possible.

This page does not contain a http-equiv meta tag. That's a best practice.

100/100

#2416

✅ **This page uses an appropriate number of DOM elements**

The number of DOM elements influences the complexity of the webpage and DOM access in JavaScript.

A well-designed webpage can offer rich content while maintaining a reasonable number of DOM elements. Read more about this here.

We recommend creating pages that contain less than 1000 DOM elements.

There are 212 DOM elements on this page.

100/100

#2445

✅ **You don't execute the same script several times**

A library or an external script is usually intended to be called once per page. However, the use of widgets can lead to several useless executions.

Duplicate scripts

It is common to see scripts used multiple times on the same page. The most common cause is the integration of social network widgets. It can be useful to find them several times on the page. That is not a bad practice. However, you should be careful that those scripts don't slow down your page.

What happens when a script is included 2 times in the code? How many times is it loaded? Parsed? Executed?

Most modern web browsers download only once a script included 2 times. An exception persists: Firefox, which will load the resource as many times as mentioned if no effective caching policy is configured.

Apart from this exception, performance issues come during the parsing and execution of the scripts. Indeed, if a script is placed three times in the code, it will be parsed and executed 3 times, on all browsers.

Do not hesitate to read this article on that topic.

How to fix it?

There is a solution to use a script several times without parsing and executing more than once. You need to write some JavaScript code that checks if the script is present. If the script is already included, it just uses it, otherwise it injects it and uses it.

Consider the following example with the Facebook widget, described in the article. Whenever you want to integrate this functionality into your page, it is necessary to include the following code:

```
(function(d, s, id){
  var js, fjs = d.getElementsByTagName(s)[0];
  if (d.getElementById(id)) {return;}
  js = d.createElement(s); js.id = id;
  js.src = "//connect.facebook.net/en_US/sdk.js";
  fjs.parentNode.insertBefore(js, fjs);
}(document, 'script', 'facebook-jssdk'));
```
Example

The bold line checks the presence of the script in the document. So the script is included, parsed and executed only during the first call in the page. Other calls will fall in the case of the bold line, and therefore will just use the script that is already included and executed.

100/100

#2539

✅ **The page doesn't use client-side redirection**

No client-side redirection (window-location, meta refresh...) has been triggered to display the page. Client-side redirections are to be avoided in any way possible.

100/100

✅ **Your scripts seem to be injected efficiently**

The script injection through the `document.write` instruction delays the rendering of your page and/or the interactivity for the user.

Do you need to inject a script?

`document.write` is sometimes used to inject a script. That is a bad practice. Example:

```
document.write('<script src="' + src + '" type="text/javascript"><\/script>');
```
Example

As described in this article, you should prefer the "createElement-insertBefore" pattern:

```
var sNew = document.createElement("script");
sNew.async = true;
sNew.src = "http://ajax.googleapis.com/ajax/libs/jquery/1.5.1/jquery.min.js";
var s0 = document.getElementsByTagName('script')[0];
s0.parentNode.insertBefore(sNew, s0);
```
Example

On this page, no `document.write` injects a script.

Cache policy

Your Optimization Priorities

93/100

#2437

✅ **7 requests from third parties don't adopt a far future cache policy**

Defining several days of cache retention for your static resources will reduce the load on your server.

The Expires header explained

Some of your resources use the `Expires` HTTP header to get an effective caching policy—this is a best practice. However, you should consider improving its configuration to make the most of the caching mechanisms. Here is an example of the `Expires` HTTP header:

> Expires: Thu, 25 Dec 2014 20:00:00 GMT
> `Example`

When you **deploy a new version of your website, remember to rename static resources** that have been modified. If you do not change their names, your users will keep resources corresponding to the old versions stored in their caches, and they may find themselves on an unstable version of your page. For example:

> myresource.min.20140101.js
> `Example`

Read the Yahoo! guidelines on this subject.

Recommended Expires header setting

We recommend setting the `Expires` HTTP header, so the date is between 2 days and 1 year.

This page contains 7 resources that **do not have a far expiration date**:

It appears these files are hosted by a third-party, so they may not be within your control. However, you should consider any alternative to these resources to improve your page performance.

- maps.googleapis.com/maps/api/js?client=[...]callback=onApiLoad
- www.google.com/maps/vt?pb=!1m5!1m4!1i16[...]-embed&token=83430
- www.google.com/maps/vt?pb=!1m5!1m4!1i16[...]-embed&token=26977
- www.google.com/maps/vt?pb=!1m5!1m4!1i16[...]-embed&token=30281
- www.google.com/maps/vt?pb=!1m5!1m4!1i16[...]-embed&token=76758
- www.google.com/maps/vt?pb=!1m5!1m4!1i16[...]-embed&token=83872
- www.google.com/maps/vt?pb=!1m5!1m4!1i16[...]embed&token=124908

Well done, these best practices are respected

100/100

#70

✅ **You do not use too long inline scripts**

Any script with a significant size should let the browser cached them in order to reduce loading time/improve performance of your returning visitor.

Inline scripts / cache policy

"inline" scripts allow to integrate easily small portions of scripts directly in the HTML code. Example:

```
<script type="text/javascript">
    (function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']...,'/analytics.js','ga');
    ga('create', 'UA-11111111-1', 'mywebsite.com');
</script>
```
`Example`

By doing so, you avoid making a request to the server to retrieve the resource. So inline scripts represent a performance gain if you want to integrate small scripts.

However, once a script has a fairly substantial size, we advise you to outsource it and perform a request to retrieve it. So you will benefit from the cache mechanism.

What should I do?

Outsource your scripts with more than 1500 characters in one or more separate files.

100/100

✅ **The 'Vary: Accept-Encoding' header is defined**

The following publicly cacheable, compressible resources should have a "Vary: Accept-Encoding" header:

It appears these files are hosted by a third-party, so they may not be within your control. However, you should consider any alternative to these resources to improve your page performance.

- maps.googleapis.co[...]Load

The `Vary: Accept-Encoding` header allows to cache two versions of the resource on proxies: one compressed, and one uncompressed. So, the clients who cannot properly decompress the files are able to access your page via a proxy, using the uncompressed version. The other users will get the compressed version.

100/100

✅ **You specify resource cache expiry headers**

Cache headers (`Cache-Control` , `ETag` , formerly `Expires`) are essential for an effective cache policy, and will greatly impact the loading time of your pages during future visits.

The `Cache-Control` header

Each resource can define its caching rules via the `Cache-Control` HTTP header. The `max-age` property defines the duration of the caching (in seconds), and can be accompanied by instructions for caching resources on proxy servers, located between the browser and the server issuing the resource.

The following header indicates that the response can be cached on proxy servers and on the browser (`public` , as opposed to `private` where only the browser is allowed to do the caching) for two hours:

Cache-Control: private, max-age=7300 `Example`

If some of your resources do not need to be cached, you can also indicate this explicitly:

Cache-Control: no-store `Example`

The `Expires` header

`Expires` is the earliest HTTP header for managing resource caching, and will help you manage the cache for browsers that do not support `Cache-Control` .

When using the `Expires` header, you can define an expiry date for each resource: as long as the date has not expired, the browser will either store or use the resource stored in the cache.

The expiry date of the resources is set using the `Expires` HTTP header:

Expires: Thu, 25 Dec 2014 20:00:00 GMT `Example`

You can specify a long expiry date for static resources (maximum 1 year), and a shorter expiry date for resources that may change (minimum 48 hours).

Reassessment of resources when the cache expires

If no cache policy is set for a resource or the duration of its caching is exceeded, the browser makes a new request to download a required resource.

To prevent the browser from downloading a resource that has not been modified since it was cached, use the `ETag` HTTP header. Each version of a resource can be associated with a validation token. When a resource's cache expires, the browser will ask the server again for the resource, passing this token with the `If-None-Match` HTTP header containing the token value. The server will compare its version of the token with the one provided. If the resource has not been modified, the server will allow the browser to renew the caching of the resource without downloading the resource again, via an 304 "Not Modified" HTTP response.

Deliberate cache invalidation

When releasing a new version of your site, remember to rename static resources that have been modified (versioning), in order to force browsers to download these new versions, instead of using cached resources, to prevent users from finding themselves in an unstable version of your page. For example:

maressource.min.20140101.js `Example`

To learn more about HTTP caching, please see Google's recommendations.

Congratulations! Your resources are cached.

Well done, these best practices are respected

100/100

#80

✅ **No frameset, frame and noframes tags detected**

These tags are obsolete, due to several issues related to the navigation consistency, SEO or browsers' bookmark features for example.

None of these tags is detected on this page.

The use of the iframe tag is prefered.

100/100

#82

✅ **No Java applets detected**

Java applets are considered obsolete in 2015. HTML5 is powerful and more widely supported. Using Java applets can lead to compatibility issues and may send negative signals to your users (e.g. the browser indicating that content was blocked because it could be dangerous).

Congratulations, this page doesn't contain Java applets.

100/100

#89

✅ **Do not use <bgsound> tag**

No `bgsound` tag detected. This is a good practice: this element is not a HTML standard. See more information.

Use the audio tag to deliver audio content on your page.

```
<audio src="my-audio-file.ogg" autoplay>
    Your browser doesn't support the audio element.
</audio>
```
Example

100/100

#99

✅ **You do not use links to Word documents**

.doc and *.docx* documents do not guarantee compatibility with all major operating systems. It's recommended to use PDF documents.

This page contains only standard links.

100/100

#2382

✅ **No deprecated attributes are detected in the <body> tag**

Some layout attributes are deprecated in HTML 5: `alink` / `background` / `bgcolor` / `link` / `text` / `vlink`

Congratulations, this page doesn't use deprecated attributes in the `body` tag.

Prefer using CSS instructions instead.

100/100

#2424

✅ **No BOM (Byte Order Mark) detected**

Some parsers are not able to interpret a page with a BOM in it.

What is the BOM?

The BOM is a hidden character located at the beginning of the page, aiming at helping to determine what encoding the page uses. But the best practices of the web prompt the use of the HTTP `Content-Type` header to define the encoding used by the page. The BOM has no reason to be in this context.

Apart from the fact that this type of indicator is useless on the web, it can lead to a certain number of issues. This is the case for example of the W3C validation that is going to try to interpret the first character which corresponds to the BOM. Then, the document will not be valid.

No resource uses a BOM.

100/100

#2454

✅ **Your characters are encoded in UTF-8**

Your content is readable by the largest number of web users.

Encoding of characters

The encoding of characters indicates to the web browser how to interpret the bytes of the web page in order to convert them in readable characters by the user. Lots of encodings exist so that all the characters specific to each language are represented. Be aware that the supported groups of encoding are different from an engine to another one. If you ever use an encoding too specific, a lot of web users won't be able to accurately interpret the page.

Why choosing the UTF-8?

The UTF-8 is known for being an encoding supported by almost every web user, and taking into account a large range of characters. Its universality is then particularly well suited to the web environment.

100/100

#2488

✅ **No Flash resource detected**

Flash is considered obsolete in 2015. HTML5 is more powerful and more widely supported. Using Flash can lead to compatibility issues and may send negative signals to your users (eg the browser warning that content was blocked because it could be dangerous).

Moreover, Google shows directly in its mobile search results that the page may not work on the user's device.

Congratulations, this page doesn't contain Flash resources.

100/100

#2493

✅ **Silverlight plugin is not used**

Silverlight plugin is considered obsolete in 2015. HTML5 is more powerful and more widely supported. Using Silverlight plugin can lead to compatibility issues and may send negative signals to your users (e.g. the browser indicating that content was blocked because it could be dangerous).

Congratulations, this page doesn't use Silverlight plugin.

100/100

✅ **The EOT font format is well declared**

Make sure that the EOT format will be well used for your visitors using Internet Explorer <= 8.

Use the EOT format for Internet Explorer <= 8

The web browsers (Firefox, Chrome, Safari, Internet Explorer, Opera ...) do not support the same font formats. Your goal is to use the lightest possible format for each platform. When you use a font, you can specify several formats, via the @font-face CSS instruction: so the browser will use the first supported format.

This is true for all browsers except Internet Explorer <= 8, that misinterpret the @font-face instruction: it will use only the first format declared. And the lighter font format for this kind of browser is the EOT format.

How to use EOT format only for IE <= 8?

Only Internet Explorer supports the EOT format. So declare it in the first position in the `@font-face` 's `src` property: Internet Explorer will be able to understand that it must use the EOT format, and other browsers, that do not support EOT, will use another format.

All fonts with EOT format are placed in the first position in the @font-face instruction.

100/100

#2496

✅ **Your font formats are well used**

Be sure to deliver the most suitable font format for each platform.

Declaring several font formats

The web browsers (Firefox, Chrome, Safari, Internet Explorer, Opera ...) do not support all the same font formats. Your goal is to use the lightest possible format for each platform. When you use a font, you can specify several formats, via the @font-face CSS instruction: so the browser will use the first compatible format that it supports.

How to declare your font formats?

Check if your formats are well ordered in the `@font-face` 's `src` property. **Here's the order to respect for your webfont formats: EOT, WOFF2, WOFF, TTF, and SVG**.

All fonts are well ordered in the `@font-face` instructions.

Your Optimization Priorities

0/100

#2443

🔴 **Reduce the page weight (2.0MB)**

The page weight is too high, slowing down its display, especially on low-speed connections. This can lead to frustration for users paying for data (see whatdoesmysitecost.com).

Evaluate the Weight of my Web Page

According to HTTPArchive, in July 2019, the average weight of a web page is 1,95MB.

How to reduce the weight of my page?

You can report to our "Data amount" category to discover the possible optimizations in your case. Images are often involved.
Moreover, make sure to build your web pages to load data that is essential to the user experience (rendering optimization of the critical path).
For other content (social networking plugins, advertising, content at the bottom of the page ...), it is better to delay the loading (asynchronous, lazy-loading ...), so they don't override priority contents.

We have established the weight distribution of the page by resource type:

- **Images :** 50,57% of total weight
- **JavaScript :** 14,94% of total weight
- **CSS :** 3,09% of total weight
- **Texts :** 0,33% of total weight
- **JSON :** 0,09% of total weight

Here is the weight of the 10 heaviest resources over the network, and that are necessary to load the page:

- https://www.inflibnet.ac.in/ilms/assets/img/hero-bg.jpg (870kB)
- www.inflibnet.ac.in/ilms/assets/vend[...]ts/icofont.woff2 (538kB)
- www.inflibnet.ac.in/ilms/assets/vend[...]s/boxicons.woff2 (90kB)
- maps.gstatic.com/maps-api-v3/embed/j[...]/3/init_embed.js (65kB)
- https://maps.googleapis.com/maps-api-v3/api/js/43/3/util.js (55kB)
- https://www.inflibnet.ac.in/ilms/assets/img/ilms%20logo.png (50kB)
- maps.googleapis.com/maps/api/js?clie[...]llback=onApiLoad (43kB)
- www.inflibnet.ac.in/ilms/assets/vend[...]ry/jquery.min.js (31kB)
- maps.googleapis.com/maps-api-v3/api/[...]s/43/3/common.js (29kB)
- www.inflibnet.ac.in/ilms/assets/vend[...]ootstrap.min.css (24kB)

90/100

#2389

✅ **Optimize your images**

Properly formatting and compressing images can save many bytes of data.

Optimize the following images to reduce their size by 23.4KiB (3% reduction).

- Losslessly compressing www.inflibnet.ac.i[...].png could save 16.7KiB (35% reduction).
- Losslessly compressing www.inflibnet.ac.i[...].jpg could save 2.7KiB (1% reduction).
- Losslessly compressing www.inflibnet.ac.i[...].png could save 2.5KiB (32% reduction).
- Losslessly compressing www.inflibnet.ac.i[...].png could save 1.5KiB (23% reduction).

Images may contain data unnecessary for their use on the web. This data can increase their size significantly. Some tools automatically remove this unnecessary data without loss of quality and thus reduce your image sizes.

Many image optimization algorithms depend on each image format. Some of them are included in graphic software like Photoshop or GIMP:

- PNG: Zopfli-png, PNGOUT, OptiPNG, AdvPNG, PNGCrush, PNGQuant...
- JPG: JPEGOptim, MozJPEG, Jpegtran, Guetzli...

FileOptimizer (Windows), ImageOptim (Mac) or Trimage (Linux) are software that combine several algorithms in one place. They will find the best possible optimization and encoding for every image, with or without quality loss.

Did you know?

ℹ️ **1 domain sends 608 bytes of cookies**

HTTP cookies are used to track a user to costumize the page according to their profile. They are sent as a HTTP header from the web server to the browser. Then, each time the browser accesses to the server, it sends a request containing the cookie received at the first response. See more information.

Here, 1 domain sends 608 bytes of cookies:

Domain name: inflibnet

- `ci_session` : 608 bytes distributed on 19 request(s)

Well done, these best practices are respected

100/100

✅ **All your resources are served from a consistent URL**

Resources with identical content should be served from the same URL to avoid duplicate downloads and additional requests.

Different URLs for the same resource

If you use different URLs to serve identical content, browsers have to request this content from the server several times, and your page will trigger unnecessary requests for data. If the same URL is used, the browser will only send one request and will use the response wherever the resource is requested. You should also use the same URL to reference the same content on several pages to benefit from the cache mechanism.

How to fix this?

Serve each resource from a unique URL. If you have to request the same resource several times, the same URL should be used to retrieve it every time.

100/100

✅ **You do not need to minify the HTML resources**

Your HTML is minified. Learn more about minifying HTML.

You get 100/100, while your resources are not minified? It means that we considered that the gains provided by the minification process were not significant enough to be reported.

100/100

✅ **Compression is enabled**

You have compression enabled. Learn more about enabling compression.

100/100

✅ **Your CSS resources are minified or don't need to be**

Your CSS is minified. Learn more about minifying CSS.

There are many tools to minify CSS files. You can try YUI Compressor or cssmin.js, recommended by Google.

100/100

✅ **Your JavaScript resources are minified**

Your JavaScript content is minified. Learn more about minifying JavaScript.

There are many tools to minify JavaScript files. You can try YUI Compressor or JSMin, recommended by Google.

100/100

#2414

✅ **16 static resources contain 512 bytes of cookies**

You send some unnecessary data on the network through cookies on static resources.

Cookies and static resources

Cookies are used to customize the content of a page depending on the user profile. There is no need to send this kind of information when requesting resources that deliver always the same content (static resources).

The requests to images, JavaScript or CSS resources do not need to use cookies.

How to fix the issue?

Do not use cookies on static resources requests. However, some domain names require the sending of cookies on all resources. In that case, you can create another domain name that will deliver your static resources. Example: `static.mydomain.com`

You should stop serving the following resources with cookies:

- www.inflibnet.ac.in/ilms[...].min.css (32 bytes of cookies)
- www.inflibnet.ac.in/ilms[...].min.css (32 bytes of cookies)
- www.inflibnet.ac.in/ilms[...].min.css (32 bytes of cookies)
- www.inflibnet.ac.in/ilms[...]tyle.css (32 bytes of cookies)
- and 12 others

100/100

#2421

✅ **All your resources have a size < 1MB**

Too heavy files should not be loaded on a web page. Be sure you need this resource to load the page.

100/100

#2436

✅ **You serve scaled images**

If your images are larger than their display area, the browser will download unnecessary data (and perform unsupervised resizing).

Avoid resizing images on the browser side

Resizing images on the browser side to reduce their rendering size is not recommended.

When the browser needs to display an image on your page, it does everything it can to adapt it to its rendering surface. If the image is too large, it will reduce it.

Provide images adapted to the display dimensions to prevents unnecessary data from being sent over the network, which reduces page loading time.

And because embedded browser algorithms are not as good as those of image manipulation tools, you will get a more satisfying visual result by resizing your images upfront, rather than letting the browser do it.

Serve Responsive Images

Several methods exist, to serve images adapted to the browser regardless of screen resolution or device pixel density. We recommend reading the following resources:

- "Responsive images" on the Mozilla Developer Network
- Picturefill, to start using the <picture> element in browsers that do not support it
- RICG, group of developers working on responsive images

The downloaded images match their display area: performance and visual quality are preserved.

100/100

#2446

✅ **Lazyloading is set (or not required)**

Load images above the fold line first, that means all the images that are visible without any scroll of the page by the visitor.

Webpage and images

Images commonly represent more than 60% of the total weight of webpages. By loading initially the only images that are visible without page scrolling, you'll reduce bandwidth consumption as for your server and your visitors. If an image has to display after any visitor's action (as scrolling), then you'd better load it on demand, when necessary (lazyloading).

How to set up lazyloading?

You can get information concerning the features and plugins proposed by your page's frameworks/CMS. You should find some easy-to-set solutions.

If you have to implement lazy-loading yourself, the simplest way may be to activate the native feature introduced by Chrome in 2019: on all targeted images, add `loading="lazy"` . The feature is implemented by recent browsers or will be in the near future. In the meantime: on browsers that don't support native lazy-load, the attribute will have no effect.

```
<img src="image.jpg" alt="my image" loading="lazy" width="50" height="50" >
```
Example

Native lazy-loading, however, is the solution that offers the least control, and is sometimes far too eager, loading images that do not require to be loaded so soon (read more).

Some Javascript libraries could help you implement lazyloading with more granularity.

🌊 **With jQuery, you could use a plugin like jQuery Lazy Load, dedicated to images lazyloading.**

100/100

#2450

✅ **This page does not send cookies > 100kB**

Keep the size of cookies as low as possible to minimize the impact on the loading time.

HTTP cookies are used to track a user to customize the page according to their profile. They are sent as an HTTP header from the web server to the browser. Then, each time the browser accesses to the server, it sends a request containing the cookie received at the first response. See more information.

100/100

#2453

✅ **This page does not send too many cookies**

Keep the size of cookies as low as possible to minimize the impact on the loading time.

HTTP cookies are used to track a user to customize the page according to their profile. They are sent as an HTTP header from the web server to the browser. Then, each time the browser accesses to the server, it sends a request containing the cookie received at the first response. See more information.

Here are listed the 608 bytes of cookies used on this page:

- `ci_session` : 608 bytes distributed on 19 request(s) (domain name: inflibnet)

100/100

#2461

✅ **11 images use the PNG format in the right way**

The choice of the right format for an image allows to reduce its weight.

The PNG format

The PNG image format is intended to the images requiring the transparency, or else to the small images having little details and colours.

Prefer the JPEG format…

The main problem of PNG format is to not support quality loss. Indeed, a format such as JPEG offers to "downgrade" the quality of the image without being perceived by the user. Doing so, you can reduce the quality of the image of about 25% whithout the user realizes it.

An image with an consequent weight will be therefore better compressed by using the JPEG format.

…or the PNG-8 format

In the case where your image necessarily requires using the transparency mecanism, not borne by the JPEG format, you should convert your "standard" PNG image, into PNG-8. This format, based upon a 256 color palette maximum allows to decrease the weight of the image whithout significantly affecting its overall quality. Tools such as pngquant or else tinypng will suport you in this procedure.

At last, if the quality given by the PNG-8 format is not appropriate for your image, you can get information on the possible advanced technics to get this behaviour without having an image in PNG format. For example, it is possible to halve your image in 2 JPEG images, one including the transparency data and the other one including the data related to the colours, and to gather the image on the customer's side with a CANVAS element.

For further information, please visit this article about image compression.

The choice of the PNG format on the images of this page doesn't have negative effect.

100/100

#2486

✅ **Redirects are lightweight**

An HTTP redirect's content is not used by web browsers. So, it provides an unnecessary weight that should be as small as possible.

Redirects on the web

Redirects can be temporary (302 HTTP code) or permanent (301).

This mechanism is for example used for secondary domains (.net to .com), the language detection mechanisms, etc.

How to fix the issue?

First, ensure that the redirect is unavoidable. In this case, it is necessary to reduce or completely remove the contents of the server responses for redirects.

A redirect should be < 1kB.

There isn't too heavy redirect.

100/100

#2487

✅ **HTTP error responses are lightweight**

The content of a HTTP error response is not used by web browsers. So, it provides an unnecessary weight that should be as small as possible.

Responses in error on the web

The resources that can be reached by the browser return a HTTP code between 400 and 599.

How to fix the issue?

First, you have to avoid requesting resources that return an error. Moreover, reduce or completely remove the contents of the server error responses.

A response error should be < 10kB.

There is no heavy resource in error.

100/100

#2490

✅ **The favicon is lightweight**

The favicon must be as lightweight as possible.

The favicon on the web

This small image allows you to display a logo on the browser tabs for instance.

How to fix the issue?

A favicon should be < 10kB.

The favicon (https://www.inflibnet.ac.in/ilms/assets/img/favicon.ico) isn't too heavy (1kB).

Your Optimization Priorities

90/100

#2418

✅ **Group 3 JavaScript files**

Each HTTP request affects the performance of your webpage (e.g., roundtrip time and bandwidth usage).

For example, it is better to request a unique 50 kB file instead of requesting 10 files that are 5 kB in size.

How should I distribute scripts?

Distribute your scripts by integrating them directly into your HTML or grouping them in files. We recommend using the latter method to take advantage of caching mechanisms.

You should consider grouping the following resources:

- www.inflibnet.ac.in/ilms/assets/vendor/[...]uery.easing.min.js
- https://www.inflibnet.ac.in/ilms/assets/vendor/aos/aos.js
- https://www.inflibnet.ac.in/ilms/assets/js/main.js

92/100

#2419

✅ **Group 3 CSS files**

Each HTTP request affects the performance of your webpage (e.g., roundtrip time and bandwidth usage).

For example, it is better to request a unique 50 kB file instead of requesting 10 files that are 5 kB in size.

How should I distribute styles?

Distribute your styles by integrating them directly into your HTML or grouping them in files. We recommend using the latter method to take advantage of caching mechanisms.

You should consider grouping the following resources:

- www.inflibnet.ac.in/ilms/assets/vendor/[...]l.carousel.min.css
- https://www.inflibnet.ac.in/ilms/assets/css/style.css
- https://www.inflibnet.ac.in/ilms/assets/vendor/aos/aos.css

Did you know?

#2543

ℹ️ **Resources distribution by domain**

This page loads data from 5 domains. This best practice retrieves the following metrics for each of these domains:

- Loading Time (Cumulative): total time spent to load all the resources
- Server Time (Cumulative): total time spent to retrieve the responses from the server (TCP connection + wait for first byte)
- Weight: data amount loaded
- Number of requests

Here is the list of all the domains used by the page:

Domain	Time (ms)	Server Time (ms)	Weight (kB)	Requests
www.inflibnet.ac.in	14518	9280	1705	20
maps.googleapis.com	899	567	179	10
www.google.com	1570	1429	54	8
maps.gstatic.com	309	182	68	3
khms0.googleapis.com	226	177	19	1

Well done, these best practices are respected

100/100

#2339

✅ **No redirect detected**

The redirects trigger avoidable roundtrips on the network and increase the page loading time.

HTTP redirects

The HTTP redirects inform the browser that the desired content is accessible from a different URL. They trigger a new HTTP request to retrieve the target resource and return an HTTP code between 300 and 399. See the specifications of HTTP redirects.

How to solve the issue?

Allow the user to directly access your content without redirects, or determine and improve what causes these excessive loading times on your redirects. See recommendations from Google.

This page does not use redirects before accessing the right content.

100/100

#2344

✅ **Only reachable resources are requested**

You should avoid requesting unreachable resources.

All of the page's resources are reachable.

These errors can affect content on and behaviors of your website and cause unnecessary network traffic, which affects the loading time of your page.

100/100

#2403

✅ **You do not use too much "prefetch" on the links**

No resource is prefetched on this page.

You should limit the number of prefetched requests to avoid network congestion. Identify your users' behavior to prefetch the most requested resources.

Note that some browsers automatically block the number of prefetched resources. For instance, Internet Explorer 11 limits this mechanism to 10 requests.

100/100

#2442

✅ **You use CSS sprites**

Combining images into CSS sprites reduces the number of files the browser has to download and accelerates the loading time.

CSS sprites explained

A CSS sprite is a single file in which several smaller images are grouped and positioned one beside the other. You can display each small image in your page by applying CSS styles, and a single request is necessary to recover all the images. Use this method only for small images, such as icons, so the CSS sprite is not too heavy.

Example

Here is a CSS sprite example :

The page also applies the styles associated with the sprite:

```
.sprite {                                                    Example
    background-image: url(img/sprite.png);
    background-repeat: no-repeat;
    display: block;
}

.sprite-browsers-firefox {
    width: 31px;
    height: 28px;
    background-position: -74px 0;
}
```

Then you have just to define the right class in your HTML file, and the icon appears:

```
<span class="sprite sprite-browsers-firefox"></span>          Example
```

How should I create CSS sprites?

Their creation can be complex, so we recommend using tools that easily generate them for you. Here are a few sprite generators:

- https://draeton.github.io/stitches/
- http://spritegen.website-performance.org/

100/100

#2466

✅ **None of your images uses only one color**

One request to an image composed of a unique color unnecessarily increases the data amount transmitted on the network.

Fetching an image composed of a single color is useless. CSS styles can achieve the same result in a more efficient way.

For instance, to draw a simple circle, you can use the following code:

```
#myElement {                                                  Example
 background:#ff0000;
 border-radius:50%;
 width:160px;
 height:160px;
}
```

All the images of this page are composed of several colors.

Your Optimization Priorities

0/100

#2449

⚠️ 2 resources do not define their content type

Each resource should define its content type in order to facilitate their interpretation by web browsers.

You have no reason to hide the type of a resource.

We advise you to set the "Content-Type" HTTP header on every resources of the page.

2 resources should define their content type:

- www.inflibnet.ac.in/ilms/asset[...]cofont.woff2
- www.inflibnet.ac.in/ilms/asset[...]xicons.woff2

0/100

#2517

⚠️ The !important declaration is used 16 times

If you abuse of this declaration, you should consider a review of your CSS code. We tolerate 10 occurrences of the !important declaration before penalizing your score.

Here are the !important detected:

https://www.inflibnet.ac.in/ilms/assets/css/style.css

- `.mobile-nav-toggle {outline: none !important}` (line 264, col 5)
- `.testimonials .owl-dot {background-color: #ddd !important}` (line 786, col 5)
- `.testimonials .owl-dot.active {background-color: #EE7642 !im...` (line 790, col 5)
- `.pt-50 {padding-top: 50px !important}` (line 1470, col 5)
- `.mb-25 {margin-bottom: 25px !important}` (line 1474, col 5)
- `.moreChannel-yt {text-align: right !important}` (line 1522, col 5)

www.inflibnet.ac.in/ilms/assets/vendor/box[...]css/boxicons.min.css

- `.bx {font-family: 'boxicons'!important}` (line 1, col 363)
- `.bx-xs {font-size: 1rem!important}` (line 1, col 5472)
- `.bx-sm {font-size: 1.55rem!important}` (line 1, col 5504)
- `.bx-md {font-size: 2.25rem!important}` (line 1, col 5539)
- `.bx-fw {margin-top: -.2em!important}` (line 1, col 5649)
- `.bx-lg {font-size: 3.0!important}` (line 1, col 5705)
- `.bx-pull-left {margin-right: .3em!important}` (line 1, col 5754)
- `.bx-pull-right {margin-left: .3em!important}` (line 1, col 5809)

www.inflibnet.ac.in/ilms/assets/vendor/owl[...]owl.carousel.min.css

- `.owl-carousel .owl-nav button.owl-next, .owl-carousel .owl-n...` (line 6, col 1397)

https://www.inflibnet.ac.in/ilms/assets/vendor/icofont/icofont.min.css

- `[class*=" icofont-"], [class^=icofont-] {font-family: IcoFon...` (line 7, col 192)

95/100

#2527

✅ **1 CSS property is duplicated**

Using several times the same property within a same CSS rule can affect the readability of the CSS. It is also an optimization opportunity: by removing duplicated properties, you will reduce the file size.

CSS properties

The CSS properties allow to apply a style to a set of elements. It is unnecessary to define 2 times the same property with the same value in a same rule.

How to improve it?

Remove one occurrence of the duplicated property. For example, the following properties:

```
.myClass {                                              Example
  margin: 10px;
  ...
  margin: 10px;
}
```

Should be replaced by:

```
.myClass {                                              Example
  margin: 10px;
}
```

The following files define the same property several times (with the same value) in a single rule:

https://www.inflibnet.ac.in/ilms/assets/css/style.css

- `.testimonials .testimonial-item p {position: relative}` (line 770, col 5)

The other tips

92/100

#2383

✅ **Separate the CSS styles from the HTML tags**

Separating HTML tags and CSS directives improves code readability and promotes factorization.

How to define CSS styles

CSS styles are used to format the page. You can use one of three main methods to define them:

- declare styles in a specific CSS file;
- declare "inline" styles (<style> tag in your HTML template);
- declare styles with the "style" attribute of a HTML tag.

How can I improve my page?

We recommend grouping your CSS styles in `<style>` tags or in separate files. That way, the HTML is only responsible for providing the structure of the page, and its layout is outsourced. The `<style>` attribute should only be generated by some JavaScript code (e.g., if you need to know the screen size).

This page uses 1 `style` attribute(s):

- ```
  <iframe style="border:0; width: 100%; height: 290px;" src="https://www.google.com/maps/embed?pb=!1m18!1m12!1
  m3!1d3667.482082517673!2d72.63115481454946!3d23.189095584868983!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1
  !3m3!1m2!1s0x395c2a3eaf8798b9%3A0x40c5e4c74f34fdcc!2sInformation+and+Library+Network+Centre+(INFLIBNET
  )!5e0!3m2!1sen!2sin!4v1503911902820"
                      frameborder="0" allowfullscreen>
  ```

Did you know?

ℹ️ 5 CSS properties are overridden

Generally, it is unnecessary to define 2 times the same property within the same rule: the second one overrides the first. This may affect the readability of CSS code. Eliminate unnecessarily overridden properties also reduce the file size.

How to improve it?

Remove one occurrence of the duplicated property. For example, the following properties:

```
.myClass {
  margin: 20px;
  ...
  margin: 10px;
}
```
Example

Should be replaced by:

```
.myClass {
  margin: 10px;
}
```
Example

The following files define the same property several times in a single rule.
Note that CSS fallbacks can justify a duplicated property.

Make sure that the following properties are duplicated on purpose:

https://www.inflibnet.ac.in/ilms/assets/css/style.css

- `#header {transition: all 0.5s}` (line 77, col 5)
- `.nav-menu>ul>li>a:before {width: 0px}` (line 157, col 5)
- `.portfolio #portfolio-filters {padding: 8px 20px}` (line 816, col 5)
- `#footer .footer-top .foo[...]type="submit"] {background: #EE7642}` (line 1440, col 5)

www.inflibnet.ac.in/ilms/assets/vendor/owl[...]owl.carousel.min.css

- `.owl-carousel.owl-grab {cursor: grab}` (line 6, col 1764)

ℹ️ 206 selectors are superfluous

It's often inefficient to bring too much preciseness in CSS selectors. For example, in the following rule:

> `body div .myClass {}`   `Example`

The body element provides no details to select the desired tags. So you should prefer the following selector:

> `div .myClass {}`   `Example`

In the same way, you do not need to specify a <ul> selector if you specify a <li> behind.

Here are listed the elements considered as potentially redundant: `ul li` , `ol li` , `table tr` , `table th` , `body` .

The following files declar some useless selectors:

https://www.inflibnet.ac.in/ilms/assets/css/style.css

- `.nav-menu .drop-down ul li:hover>a` (line 206, col 1)
- `#footer .footer-top .footer-links ul li` (line 1396, col 1)
- `#footer .footer-top .footer-links ul li:first-child` (line 1402, col 1)
- `.box-Part ul li` (line 1549, col 1)
- `.no-style ul li` (line 1571, col 1)
- `.subjects_body ul li a` (line 1579, col 1)

https://www.inflibnet.ac.in/ilms/assets/vendor/aos/aos.css

- `body[data-aos-duration="50"] [data-aos]` (line 1, col 1)
- `body[data-aos-delay="50"] [data-aos]` (line 1, col 111)
- `body[data-aos-delay="50"] [data-aos].aos-animate` (line 1, col 209)
- `body[data-aos-duration="100"] [data-aos]` (line 1, col 334)
- `body[data-aos-delay="100"] [data-aos]` (line 1, col 445)
- `body[data-aos-delay="100"] [data-aos].aos-animate` (line 1, col 545)
- `body[data-aos-duration="150"] [data-aos]` (line 1, col 671)
- `body[data-aos-delay="150"] [data-aos]` (line 1, col 783)
- and 192 others

ℹ️ 68% of the colors are used only once

The analysed page uses 63 different color codes within your own CSS files, and 43 of them are used only once (some might be very close to others, eg #FF0000 and #FE0000).

We recommend you to try to harmonize the colors of your website to simplify your CSS, but also to avoid inconsistencies in your UI.

The following colors are used only once:

`#000000` `#03aa67` `#062822` `#0c0b1c` `#10c67d` `#111111` `#14132d` `#14806c`
`#18302c` `#18967e` `#2282ff` `#22d8b6` `#333333` `#34dfbf` `#3fcdc7` `#41cf2e` `#4ae3c6`
`#4d4d4d` `#515756` `#5e5e5e` `#848484` `#888888` `#999999` `#a8d3cc` `#dddddd`
`#dedede` `#dfdfdf` `#e0e5e9` `#ed3c0d` `#ee7844` `#eff7f6` `#f1f1f1` `#f8f8ff` `#ff689b`
`#ffffff` `rgba(0,0,0,0.6)` `rgba(0,0,0,0.8)` `rgba(0,0,0,0.9)` `rgba(0,212,255,0)` `rgba(127,137,161,0.25)`
`rgba(16,31,29,0.6)` `rgba(18,25,47,0.46)` `rgba(37,37,37,0.1)`

ℹ️ 1 CSS selector is duplicated

Using several times the same selector for several declarative blocks within a single CSS file can affect the readability and maintainability of the code. It is also an optimization opportunity: by grouping these elements within a single rule, you will reduce the file size and optimize the rendering times.

The CSS Object Model (CSSOM)

The CSS rules allow to select elements from the HTML code in order to apply styling properties.

In order to do that, the browser constructs its own model of the CSS markup it has fetched: the CSS Object Model. This transformation takes several steps: reading the file, converting the strings of text into browser tokens, transforming these tokens into objects with properties and rules and organizing the elements in a tree-like model. These operations will take place more quickly if the CSS code is written in a simple and non-redundant way.

How to improve the CSS rules?

For maintenance as well as performance, you have to reduce the number of rules contained into your CSS files and group rules whenever it's possible. For instance, the following rules:

```
.myClass {
  margin: 0;
}
...
.myClass{
  border: 1px solid black;
}
```
Example

Should be merged into a single one:

```
.myClass {
  margin: 0;
  border: 1px solid black;
}
```
Example

If the duplication is the result of a legacy process or is justified for maintenance reasons, you can consider using a CSS minification tool able to do automatically this improvement.

You can save 11 characters. The following files define the same selector in different rules:

https://www.inflibnet.ac.in/ilms/assets/css/style.css

- `.box-Part p` (2 times)

Well done, these best practices are respected

✅ You provide a favicon

Well done, you defined a favicon on this page.

Favicon is a small image providing an icon to a website. It's located in the root of your server and the browser will always request it. It is better not to respond with a 404 HTTP code (not found).

Moreover, this file will be asked on every requested web page, so make it cacheable: the client will request it only once. See more information.

100/100

#2398

✅ **You do not gzip/deflate PNG images**

Compression has a cost, on the server and the client browser. You should enable it only if it is effective.

PNG and compression

PNG images does not support gzip compression. It is useless to perform compression operations on server side, and decompression on client side for this kind of files.

This page does not compress images. This is a best practice.

100/100

#2448

✅ **The extensions of your resources are consistent**

The extension of a resource allows to identify easily its content type. You have no reason to indicate a different extension of the actual content type of the resource.

For instance, the file `resource.js` must use a `Content-Type` HTTP header equal to `application/javascript`. You can check the `Content-Type` HTTP header value using a debugger tool (e.g.: DevTools on Chrome).

100/100

#2491

✅ **IDs within your HTML code are unique**

Using the same ID on several elements can have side effects, especially during JavaScript executions or when applying CSS rules.

IDs explained

Each element of a web page can be identified thanks to the `id` attribute:

```
Example
<p>
 <span id="mySpan1"></span>
</p>
```

These IDs allow you to manipulate your elements with CSS or JavaScript instructions.

How to properly use IDs?

You must ensure identifiers are not duplicated within the page. If you want to share a property or a behavior between multiple items, you have to use the `class` attribute, which is dedicated to this purpose:

```
Example
<p>
 <span class="mySpans"></span><span class="mySpans"></span>
</p>
```

100/100

‌

#2515

✅ **No CSS property is overridden by a shorthand**

CSS has some pitfalls that can cause side effects on your styles.

Override CSS properties

When a property is used 2 times within a CSS rule, the second one overrides the first.
CSS provides some shorthand properties, that can lead to some mistakes. For instance, the "border" property is applied to all borders of an element, avoiding the use of all "border-top", "border-right", "border-left" and "border-bottom" properties.

For instance:

```
.myClass {
 border-color:red;
 border:5px solid; // border will take again the default color
 }
```
Example

"border-color" is overridden by the "border" shorthand property.

Avoid side effects

The use of shorthand properties is risky, because it comes with an implicit override. If the resulting style on your page is the desired one, you should delete the overridden property (it will also reduce the file size).

100/100

#2516

✅ **Your CSS selectors are not too complex**

Using simple CSS selectors improves your code's readability and performance.

CSS rules

The CSS rules allow to select elements from the HTML code, and to apply layout properties.

They use selectors (identifiers, classes, tag names, etc.) for selecting the HTML elements. A well-structured page allows writing simple CSS rules, quickly readable and efficient.

How to improve it?

Simplify your selectors as much as possible. Do not hesitate to introduce new classes in your code to avoid complex selectors.
For instance, the following selector:

```
body td .myClass div .myClass2 span .myClass3 {}
```
Example

Should be replaced by:

```
#myId .myClass3 {}
```
Example

SEO

Your Optimization Priorities

0/100

#84

⛔ You should define a 'description' meta tag

The page should define a unique description.

Description in search engines

The description of the page may be directly displayed in search engine results pages (SERP):

Amazon.com: Online Shopping for Electronics, Apparel, Computers ...
https://www.amazon.com/ ▾ Traduire cette page
Online retailer of books, movies, music and games along with electronics, toys, apparel, sports, tools, groceries and general home and garden items. Region 1 ...

It allows you to control at best the entry preview in search engines, and to improve the click rate to your page. Learn more.

How to define a page's description?

Use `<meta name="description" content="page description">` and place it in the `<head>` tag.

No <meta> `description` has been found on this page. Please provide a <meta> `description` .

0/100

#2503

⚠ robots.txt file should be defined

Indicate to web crawlers which URLs should be explored on your website.

The robots.txt file

Place your robots.txt file in the root of the website. It will be interpreted by the robots in charge of your SEO. It delivers instructions to specify the pages to explore by robots, like Google bot.

Note that these directives are indicative only. A lambda robot will not be blocked by the restrictions specified by the file.

We have not detected the robots.txt file on this website, you should define one:

- https://www.inflibnet.ac.in/robots.txt

0/100

⚠️ **This page does not specify a breadcrumb for search engines**

The breadcrumb enables to visually enhance the results matching your website during a research. On Google, you can find it instead of the URL for example:

Tim Minchin: how my beat poem took the world by Storm ...
www.theguardian.com › Arts › Books › Poetry ▾ The Guardian ▾
Oct 6, 2014 - Join **Tim Minchin** for a live streamed Q&A on Monday 6 October (7 ...
extract from the introduction to Storm by **Tim Minchin** with DC Turner and ...

On a search made through a desktop device, the breadcrumb only replaces the URL if it is defined on the page.

About the mobile search, Google has been consistently displaying the breadcrumb since April 2015. If you do not define a breadcrumb, Google will try to generate one for the page, with uncertain results. Define a breadcrumb for your page to ensure proper display in Google and other search engines.

Defining its own breadcrumb

You must define your breadcrumb using one of the accepted implementations (see Google recommendations).

```
Example
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "BreadcrumbList",
  "itemListElement": [{
    "@type": "ListItem",
    "position": 1,
    "name": "Books",
    "item": "https://example.com/books"
  },{
    "@type": "ListItem",
    "position": 2,
    "name": "Authors",
    "item": "https://example.com/books/authors"
  },{
    "@type": "ListItem",
    "position": 3,
    "name": "Ann Leckie",
    "item": "https://example.com/books/authors/annleckie"
  },{
    "@type": "ListItem",
    "position": 4,
    "name": "Ancillary Justice",
    "item": "https://example.com/books/authors/ancillaryjustice"
  }]
}
</script>
```

On this analyzed page, we couldn't find a breadcrumb.

The other tips

0/100

⚠️ **Your site doesn't use Open Graph properties**

#2399

You can help social networks understand information related to the page by using Open Graph properties.

The Open Graph properties explained

Several properties allow social networks to learn more about the page's content. We recommend using at least the required properties:

- `<meta property="og:title" content="The title" />` `Example`

- `<meta property="og:type" content="The type" />` `Example`

- `<meta property="og:url" content="http://url.com/" />` `Example`

- `<meta property="og:image" content="http://image.jpg" />` `Example`

This information is used to improve links between your page and various social networks, including Facebook. Read more about Open Graph here.

This page does not provide information to social networks.

Did you know?

#2457

ℹ️ **This page contains 9 links**

Two kind of links exist:

- **Internal links** that refer to pages with the same domain name;
- **External links** that point to other websites (must be relevant and point towards quality content).

If you reference many links, you can ask the SEO crawlers to consider only some of them, by adding the `rel=nofollow` attribute to the irrelevant ones (e.g., advertisements).

Here is the distribution of 9 links present in the page:

- 8 internal links (88,89%)
- 1 "follow" external link (11,11%)
- No "nofollow" external link (0,00%)

#2507

ℹ️ **Does this page handle the www redirect?**

Redirect `https://inflibnet.ac.in/ilms/` to `https://www.inflibnet.ac.in/ilms/` in order to avoid duplicate content.

What is the difference ?

Adding the "www" in front of a web address is a convention, that remains completely optional.

Some users are accustomed to access a website by adding www, and others do not. We advise you to deliver your content for these two addresses.

To avoid duplicate content issues with search engines, it is necessary to use a redirect (permanent) to point one of the addresses to the other.

How to fix the issue?

You should perform a 301 redirect from `https://inflibnet.ac.in/ilms/` to `https://www.inflibnet.ac.in/ilms/` .

🖊️ **You can configure a redirect thanks to your Apache server.** Here is an example :

```
RewriteEngine On

RewriteCond %{HTTP_HOST} ^mywebsite.com
RewriteRule (.*) https://www.mywebsite.com$1 [R=301,L]
```
`Example`

Well done, these best practices are respected

100/100

#69

✅ **Your <img> tags use an alt attribute**

Moreover, the `alt` attribute is also an important criterion for SEO. Indeed, search engines crawlers cannot parse graphic contents. That is why they use the alternative text to return consistent results, like in Google images.

>   `Example`

The `alt` attribute is used in several cases unrelated to SEO:

- When a screen reader is in use for accessibility purposes;
- While image is loading, particularly for slow connections;
- When the image file is not found.

You have 3 `img` tags and they all have the `alt` attribute.

If you cannot think of a suitable description for an image, set the text to an empty string "". Make an effort to ensure that as many images as possible on the page include a relevant alternative. Read the W3C recommendations here.

100/100

#78

✅ **This page defines <h1> and <h2> tags**

We recommend putting page keywords in at least the h1 and h2 tags. Search engines use the h1, h2, and h3 tags for SEO purposes.
This page contains:

- 1 <h1> element(s)
- 3 <h2> element(s)
- 3 <h3> element(s)

100/100

#81

✅ **This page uses only standard image formats**

The images that use a non-standard format may not be indexed by search engines.

Only these image formats are considered standard on the web: jpeg, jpg, png, gif, svg, ico, webp. You should consider an alternative to any other format.

Moreover, remember to treat the text around your images: some search engines analyze approximately the 10 words preceding and following the image in order to add a context to the image.

✅ This page specifies a <title> tag

100/100

#83

The page should define a unique title (using a <title> tag).

Use of titles by search engines

Once properly configured, the page title can be displayed in the search engine results page:

Amazon.com: Online Shopping for Electronics, Apparel, Computers ...
https://www.amazon.com/ ▾ Traduire cette page
Online retailer of books, movies, music and games along with electronics, toys, apparel, sports, tools,
groceries and general home and garden items. Region 1 ...

Using a suitable title is a major criterion for SEO. It allows you to control at best what is displayed in search results pages and determine the keywords you want your site pops out.

How to define the title of a web page?

The title of the page is specified into the `<title>` tag, which must be placed into the `<head>` tag, at the beginning of the code.

This page defines the title trough the `title` tag.

Here is the page's title:

LMS as a service

✅ You do not use query strings in the URL

100/100

#90

This URL does not contain any parameter.

A URL should be as readable as possible. For instance, `http://example.com?userId=332&group=MyGroup` is less readable than `http://example.com/mygroup/me` .

✅ The words are well separated in this URL

100/100

#2345

You should prefer the use of dashes in the URL.

URL and SEO

The words in the URL are among the many factors impacting the SEO: if you search the words `web performance` in a search engine, one criterion used by Google will be to check if the URL contains the words `performance` and `web` .

But for Google, underscores are not word separators: If your URL contains `web_performance` it will not help to highlight the page on the query `web performance` (`web_performance` is regarded as a unique word).

This behavior is however not common to all search engines. For example, Bing does not differentiate dashes and underscores.

Please note that Google does not penalize you for using an underscore in the URL.

How to resolve the issue?

We recommend using `-` instead of `_` on your new web pages.

This is more complex fix the issue on existing web pages, such as the one analyzed here, because you can't just rename the URL (you will lose all your SEO efforts). It is then necessary to set up a permanent redirect (HTTP code 301) retaining the old URL, that forwards to the new URL. Beware: many redirects on your site can also affect the visibility of your web pages. So do not use redirects if the number of relevant pages remains limited.

This URL respects the tip.

100/100

#2444

✅ **Your <title> tag is an appropriate length**

This page defines one title which contains less than 75 characters (see more information):

> LMS as a service

The longer your title is, the more your chances are to see the search engines to truncate it or even to select another one from your page content.

100/100

#2467

✅ **None of your titles are empty**

`<h1>` , `<h2>` and `<h3>` tags should contain keywords related to the content.

The titles included on this page provide content.

Your Optimization Priorities

0/100

#2434

⛔ The Content Security Policy is missing

Protect your website from cross-site scripting (XSS) attacks by setting up a restrictive Content-Security-Policy.

XSS attacks explained

XSS attacks are a type of attack in which malicious data is maliciously added to websites. The number of vulnerabilities allowing these attacks is quite large, which is why it is as useful to prevent them as to limit their harmful effects.

You can protect your pages against these attacks and their effects by restricting execution to code portions either legitimized by the domain to which they belong or by a unique integrity token. The code that does not match this security policy will not be executed and the user will be informed.

You can learn more about XSS attacks on the Open Web Application Security Project (OWASP) Website.

Configure a "Content-Security-Policy" (CSP) HTTP header

Set up a "Content-Security-Policy" (CSP) HTTP header to prevent or limit the damage caused by an XSS attack. To specify a security policy configure your server so the response of the first resource contains the "Content-Security-Policy" HTTP header.

Here's an example:

```
Content-Security-Policy: script-src 'self' https://apis.google.com                    Example
```

In this case, only scripts coming from the current host or https://apis.google.com will be executed.

Read more about the CSP HTTP header by consulting the CSP directives specification.

Please, be careful, if the header is misconfigured, some of your content, scripts, or styles may be blocked. That could cause unwanted side effects. Moreover, the restrictions apply to all pages of the website. We recommend you test the different pages of your website before deploying this header in your production environment.

🖊 **CSP can be configured with your Apache server. Make sure that the mod_headers module is enabled. Then, you can specify your content security policy (in your .htaccess file, for example).** Here is an example :

```
<IfModule mod_headers.c>                                                               Example
Header set Content-Security-Policy "script-src 'self' https://www.google.com"
</IfModule>
```

This example allows scripts from the same origin (same scheme, host and port) and google.com.

No Content Security Policy on this page: it is more easily exposed to XSS attacks.

0/100

⛔ **1 cookie is not secure**

A cookie sent from the server to a web browser via the HTTPS protocol should only transits on a secure connection (except for some specific cases).

HTTP cookies

HTTP cookies are set by the server to the web browser via the `Set-Cookie` HTTP header. Then, the browser transmits the cookies to the server for the next requests by using the `Cookie` HTTP header. When the server uses a secure connection (HTTPS), the cookie probably contains some sensitive data: you have to guarantee that the cookie cannot be exploited on an insecure connection.

The Secure directive

By adding the `Secure` instruction in the `Set-Cookie` HTTP header, the server informs the browser that it is allowed to transmit the cookie over secure connection only.

Caution: Ensure that the HTTP to HTTPS redirect is activated on your website. Otherwise, the Secure cookie may not be sent on HTTP request.

🖌 **The `Set-Cookie` HTTP header can be configured with your Apache server. Make sure that the mod_headers module is enabled. Then, you can specify the header (in your .htaccess file, for example). Here is an example:**

```
<IfModule mod_headers.c>
# only for Apache > 2.2.4:
Header edit Set-Cookie ^(.*)$ $1;HttpOnly;Secure

# lower versions:
Header set Set-Cookie HttpOnly;Secure
</IfModule>
```
`Example`

The following Cookies are not secure, you should add the `Secure` instruction in the `Set-Cookie` HTTP header:

https://www.inflibnet.ac.in/ilms/

- Set-Cookie: ci_session=d32br3u7nogctkuqr237cns53jbhgph9; expires=Fri, 01-Jan-2021 12:12:38 GMT; Max-Age=7200; path=/; HttpOnly

Shared infrastructure cookies

With some hosting providers, especially on shared infrastructures, cookies may be set without any control on your part. The information they contain is usually not sensitive. You can check this and ignore this warning if that applies to you.

0/100

#2481

⛔ **This page is exposed to "clickjacking" type attacks**

Keep malicious people from integrating your pages into their websites.

Clickjacking explained

This kind of attack happens when your page gets integrated with a malicious website via <frame> or <iframe> tags. By doing this, attackers can persuade users that they are on your own page when they are not. The unsuspecting user may enter personal information that is visible on and thus vulnerable to the malicious website.

To avoid this, always indicate which domains have permission to integrate your pages.

How to prevent clickjacking?

There are two main ways to prevent that behavior.

1/ Configure a "X-Frame-Options" HTTP header. Configure your server so the main resource response includes the "X-Frame-Options" HTTP header.

Three values may be defined:

- `DENY` to prevent any frame or iframe from integrating the page;
- `SAMEORIGIN` to authorize only frames from the same domain name;
- `ALLOW-FROM uri` to indicate the domains allowed to integrate a page into frame (however is not compatible with some browsers)

2/ Define an explicit `frame-ancestors` directive into a Content-Security-Policy HTTP Header. "frame-ancestors" directive is a newer, hence supported by fewer browsers, approach that will allow your website to authorize multiple domains instead of only the current origin. Setting this directive to 'none' is similar to `X-Frame-Options: DENY`.

Which approach to choose? If you only have the current domain to allow, do set up the two security features, for better compatibility with older browsers. If you want to allow multiple domains, you should only implement the frame-ancestors security policy.

🖊 **The "X-Frame-Options" HTTP header can be configured with your Apache server. Make sure that the mod_headers module is enabled. Then, you can specify the header (in your .htaccess file, for example). Here is an example:**

```
<IfModule mod_headers.c>
Header always set X-FRAME-OPTIONS "DENY"
</IfModule>
```
Example

Neither the "X-Frame-Options" HTTP header nor the "frame-ancestors" security police are configured on this page; you are more likely to be exposed to clickjacking.

The other tips

0/100

#2492

⚠️ **1 iframe could be secure with a sandbox attribute**

Restrict as much as possible the actions that can be processed by external content embedded on your website.

External contents

You should pay a special attention to external contents (social networks widgets, ads, etc.) embedded via the <iframe> tag. To limit the risks, the W3C has added the `sandbox` attribute in the HTML5 specifications. It restricts the available actions from an iframe (on major modern browsers).

How to fix the issue?

Add the `sandbox` attribute on your iframe tags to control as much as possible the behavior of its content. Be sure to **use the correct values for this attribute**.

A security policy is missing for the following iframe:

- <iframe style="border:0; width: 100%; height: 290px;" src="https://www.google.com/maps/embed?pb=!1m18!1m12!1 m3!1d3667.482082517673!2d72.63115481454946!3d23.189095584868983!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1 !3m3!1m2!1s0x395c2a3eaf8798b9%3A0x40c5e4c74f34fdcc!2sInformation+and+Library+Network+Centre+(INFLIBNET)!5e0!3m2!1sen!2sin!4v1503911902820"
                        frameborder="0" allowf...

0/100

⚠️ **Disable the auto detection of resource type**

Protect yourself from malicious exploitation via MIME sniffing.

MIME-Type sniffing explained

Internet Explorer and Chrome browsers have a feature called "MIME-Type sniffing" that automatically detects a web resource's type. This means, for example, that a resource identified as an image can be read as a script if its content is a script.

This property allows a malicious person to send a file to your website to inject malicious code. We advise you to disable the MIME-Type sniffing to limit such activity.

Chrome has been working on a feature called Site Isolation which provides extensive mitigation against exploitation of these types of vulnerabilities. Site Isolation is more effective when MIME types are correct.

How to prevent MIME-Type sniffing

Configure a "X-Content-Type-Options" HTTP header. Add the "X-Content-Type-Options" HTTP header **in the responses of each resource**, associated to the "nosniff" value. It allows you to guard against such misinterpretations of your resources.

🖌 **The "X-Content-Type-Options" HTTP header can be configured with your Apache server. Make sure that the mod_headers module is enabled. Then, you can specify the header (in your .htaccess file, for example). Here is an example:**

```
<IfModule mod_headers.c>
Header always set X-Content-Type-Options "nosniff"
</IfModule>
```
Example

On this page, **you should configure the following resources**, that risk being misinterpreted:

Resources from "inflibnet"

- https://www.inflibnet.ac.in/ilms/assets/vendor/bootstrap/css/bootstrap.min.css
- https://www.inflibnet.ac.in/ilms/assets/vendor/icofont/icofont.min.css
- www.inflibnet.ac.in/ilms/assets/vendor/owl.carou[...]ets/owl.carousel.min.css
- https://www.inflibnet.ac.in/ilms/assets/css/style.css
- https://www.inflibnet.ac.in/ilms/assets/vendor/jquery/jquery.min.js
- https://www.inflibnet.ac.in/ilms/assets/vendor/boxicons/css/boxicons.min.css
- https://www.inflibnet.ac.in/ilms/assets/vendor/aos/aos.css
- www.inflibnet.ac.in/ilms/assets/vendor/bootstrap[...]/bootstrap.bundle.min.js
- www.inflibnet.ac.in/ilms/assets/vendor/jquery.ea[...]ing/jquery.easing.min.js
- https://www.inflibnet.ac.in/ilms/assets/vendor/owl.carousel/owl.carousel.min.js
- https://www.inflibnet.ac.in/ilms/assets/vendor/aos/aos.js
- https://www.inflibnet.ac.in/ilms/assets/js/main.js

Resources hosted by a third-party

It appears these files are hosted by a third-party, so they may not be within your control. However, you should consider any alternative to these resources to improve your page performance.

- maps.googleapis.com/maps/api/js?client=google-ma[...]on=in&callback=onApiLoad
- maps.googleapis.com/maps/api/js/ViewportInfoServ[...]e-maps-embed&token=54660
- maps.googleapis.com/maps/api/js/ViewportInfoServ[...]-maps-embed&token=110654

Did you know?

ℹ️ **SSL Certificate**

Your SSL certificate will expire on 08/20/2022. Update your certificate before that date.

What happens if my certificate expires?

Letting a certificate expire can have consequences for end users who will then see many error or alert messages while browsing the site, warning them of possible frauds, identity thefts or traffic interceptions. These alerts can have a very negative impact on the user's perception of the visited domain.

ℹ️ Your server should be able to communicate with HTTP while it uses a HTTPS connection

Take precautionary measures against attacks like "man in the middle" by making sure to only communicate in HTTPS with the server.

The HTTP Strict Transport Security (HSTS) Header

When you communicate with a server through a secure connection, every sent request towards this server should use the HTTPS protocol. The HTTP HSTS header allows to indicate to the browser that all the requests sent to the domain concerned must be done via HTTPS. If the URL is presented under "http://...", the web browser is automatically going to replace it by "https://...".

However, we advise you to not set this header unless your entire website serves its resources in HTTPS.

🖊️ **HSTS can be configured with your Apache server. Make sure that the mod_headers module is enabled. Then, you can specify your HSTS (in your .htaccess file, for example).** Here is an example :

```
<IfModule mod_headers.c>
Header always set Strict-Transport-Security "max-age=63072000; includeSubDomains"
</IfModule>
```
Example

No HSTS header has been detected on this page.

ℹ️ 9 resources on this page are for public use

By default, the browser accepts to perform AJAX requests, or to retrieve web fonts, only on the same domain name of the page. So a font provided by toto.com can only be used by the pages of toto.com. This prevents misuse of your resources by any site.

Some resources are public, and explicitly want to be available to everyone (eg Google Fonts). In this case, the HTTP header `Access-Control-Allow-Origin` can be used with the value "*". You should, however, use this property if your resource has aimed to be used by the greatest number. Otherwise, we recommend that you keep the default, or set a specific domain name in the "Access-Control-Allow-Origin" HTTP header.

You should be aware of the following resources, that use a `Access-Control-Allow-Origin: *` HTTP header. Make sure they are actually intended to be used by pages from all domain names:

It appears these files are hosted by a third-party, so they may not be within your control. However, you should consider any alternative to these resources to improve your page performance.

- https://maps.gstatic.com/mapfiles/embed/images/google4.png
- https://maps.gstatic.com/mapfiles/openhand_8_8.cur
- www.google.com/maps/vt?pb=!1m5!1m4!1i16!2i45990![...]e-maps-embed&token=83430
- www.google.com/maps/vt?pb=!1m5!1m4!1i16!2i45989![...]e-maps-embed&token=26977
- www.google.com/maps/vt?pb=!1m5!1m4!1i16!2i45989![...]e-maps-embed&token=30281
- www.google.com/maps/vt?pb=!1m5!1m4!1i16!2i45990![...]e-maps-embed&token=76758
- www.google.com/maps/vt?pb=!1m5!1m4!1i16!2i45991![...]e-maps-embed&token=83872
- www.google.com/maps/vt?pb=!1m5!1m4!1i16!2i45991![...]-maps-embed&token=124908
- https://khms0.googleapis.com/kh?v=883&hl=en-US&gl=IN&x=5748&y=3553&z=13

ℹ️ **Do all third-party resources deliver the right content?**

If this page loads resources from third parties, you should ensure their integrity.

SubResource Integrity (SRI)

Use SRI to ensure that a third-party resource has not been tampered. Add the `integrity` attribute to `<script>` and `<link>` tags loading this kind of resource. Example:

```
<script src="https://exemple.com/exemple-framework.js"
  integrity="sha384-oqVuAfXRKap7fdgcCY5uykM6+R9GqQ8K/uxy9rx7HNQlGYl1kPzQho1wx4JwY8wC"
  crossorigin="anonymous">
</script>
```
Example

The `integrity` attribute value is equal to the base64-encoded hash (SHA) of the resource. The browser compares this hash with the downloaded content in order to determine if the resource matches the expected content.

You can create the SHA thanks to several tool. In command line, you can use openssl. You can also test some online tools, as srihash.org or report-uri.io. Learn more about SubResource Integrity.

Well done, these best practices are respected

100/100

#2397

✅ **This page only uses secure content**

This page was transmitted over the HTTPS protocol, and all resources are fetched using the HTTPS protocol. So there is no Mixed Content vulnerability.

100/100

#2509

✅ **The secure version is used systematically**

Users using the HTTP version of the page must be redirected to the secure version.

HTTPs redirect

Even if a page is secure, visitors may still use the HTTP version (via an external link, or because they have bookmarked the HTTP version of the page, for example). Always set up a redirection so that users accessing the page using the HTTP protocol are redirected to a secure version.

🖌 **An Apache server can be configured to rewrite HTTP URLs.**

Your users are automatically redirected to the secure version of the page.

Well done, these best practices are respected

100/100

#74

✅ **Your Apache server version is not exposed**

You are using Apache, but we are not able to detect the version. It is more difficult for a hacker to attack your website, because he does not know the version you use. This is a good practice.

There are two values to check if you want to hide your server version: the ServerSignature and the ServerTokens (*/etc/apache2/conf.d/security* file on a Linux server).

```
# Hide the version from the 'Server' HTTP Header.
# (e.g.): display only "Server: Apache"
ServerTokens Prod
```
Example

```
# Don't add a trailing footer line under server-generated document,
# containing the server name and its version.
ServerSignature Off
```
Example

However, keep in mind that the best way to protect your system from attacks is to regularly update your Apache server.

Well done, these best practices are respected

100/100

#2573

✅ **Your PHP version is properly hidden**

Hide your version of PHP to limit your exposure to attacks and increase your server's security.

You are using PHP, but we are not able to detect which version.

Why hide the PHP version?

Knowing the version of PHP used on a server helps an attacker to target security holes and exploit vulnerabilities.

How to hide the PHP version

To hide your PHP version, you need to find your php.ini configuration file and locate the keyword `expose_php` to set its value to `off` .

```
# php.ini                                                          Example
# expose_php = off
```

Once the php.ini file modified, you will need to restart your web server.

Did you know?

#2303   ℹ️ **Additional information about your jQuery performance**

jQuery is the most used JavaScript library. Upgrade your website performance respecting the jQuery best practices. We recommend that you learn the basics of the jQuery performance, reading the following link: http://learn.jquery.com/performance/.

Well done, these best practices are respected

100/100

#2299

✅ **No DOM manipulation inside loops detected**

Working directly with the DOM has a cost. If you have to add elements to a node, you should prefer to append them once rather than one by one.

Well done, none of your loop contains a jQuery `append` method.

100/100

#2302

✅ **You do not use excessive specificity on selectors**

Using too specific jQuery selectors impacts performance: see more information. Here is an example of a good use of the library:

```
$( ".data table.firstClass td.secondClass" );
// Better: Drop the middle if possible
$( ".data td.secondClass" );
```
Example

This page does not contain too specific selectors, this is a good practice.