

Problem - 2:

You are required to explore on the given questions and summarize your findings.

- 1. What are the ways of optimizing the web application loading time?**
 - 2. Define and explain various components of an URL?**
-

1. Ways of optimizing the web application loading time

1. Minimize HTTP requests

The first step to minimizing your requests is to figure out how many your site currently makes, to use as a benchmark. Reducing this number of requests will speed up your site, look through your files and see if any are unnecessary.

2. Minify and combine files

The fewer elements on a page, the fewer HTTP requests a browser will need to make the page render — and the faster it will load.

3. Use asynchronous loading for CSS and JavaScript files

If your scripts load synchronously, they load one at a time, in the order they appear on the page. If your scripts load asynchronously, on the other hand, some of them will load simultaneously. Loading files asynchronously can speed up your pages because when a browser loads a page, it moves from top to bottom.

4. Defer JavaScript loading

Deferring a file means preventing it from loading until after other elements have loaded. If you defer larger files, like JavaScript, you ensure that the rest of your content can load without a delay.

5. Minimize time to first byte(TTFB)

In general, most issues with slow TTFB are caused by either network issues, dynamic content creation, web server configuration, and traffic.

Unlike a lot of the front-end performance factors most site owners focus on, this is a server-side concern.

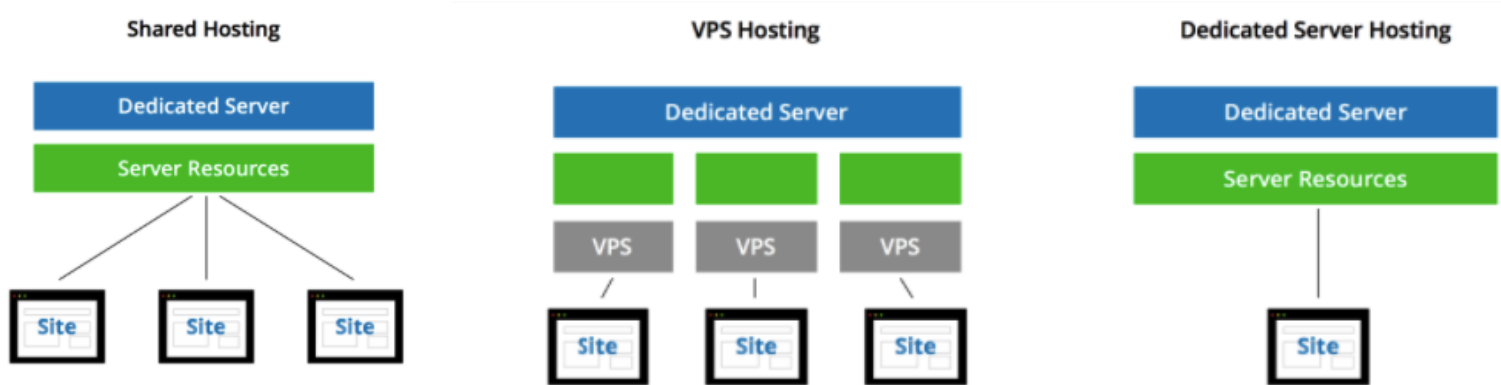
There are three steps that need to happen between that initial request and the first byte of data:

- DNS lookup
- Server processing
- Response

You can reduce the time that this takes by enabling caching

6. Reduce server response time

7. Choose the right hosting option for your needs



If you're at a point where your traffic levels are slowing down your server response times, it may be time to switch from shared hosting to a VPS, or from a VPS to a dedicated server.

8. Run a compression audit

To get a sense of how compression could speed up your site, you can use [GIDNetwork](#) to run a compression audit.

9. Compress your files and images

Compressing files is one of the easiest ways to reduce load times, and today, enabling compression with [Gzip](#) is considered standard practice.

10. Enable browser caching

11. Reduce image sizes

12. Use a CDN

With a CDN, you cache your site on a global network of servers. When a user's browser requests files from your site, that request is routed to the closest server.



13. Use external hosting platforms

Host your video on a third-party service like YouTube, Vimeo, or Wistia

14. Optimize CSS delivery

Use only one CSS stylesheet and no inline CSS

15. Prioritize above-the-fold content (lazy loading)

Load the content visible in first view, and keep loading the beneath data. Lazy Load is an extremely simple plugin that uses jQuery.sonar to only load images when they are visible in a user's browser.

16. Reduce the number of plugins you use on your site

Unfortunately, having too many plugins installed can cause some issues. They can slow your site, create security issues, and even cause crashes and other technical difficulties. Deactivating and deleting any that are unnecessary can both improve your overall speed and make maintenance easier in the long run.

17. Reduce redirects

You can start by using [Screaming Frog](#) to quickly identify all of the redirects currently on your site. Remove one's possible.

18. Reduce external scripts

19. Monitor your speed over time

20. Monitor mobile page speed

21. Install [Google PageSpeed](#)

The PageSpeed modules are open-source server modules that optimize your site automatically.

22. While working with Liferay portal which had portlets as servlet application deployed on the portal, a software plugin named [probe](#) could be deployed on tomcat server to view applications that are using up memory or choking up the session timeouts, and the same could be used to just restart that particular application thus avoiding server restart. Helped performing live monitoring and avoid crashes.

2. Define and explain various components of an URL?

Also known as a web address, a URL (Uniform Resource Locator) is a form of URI and a standardized naming convention for addressing documents accessible over the Internet and Intranet.

URLs have the following format:

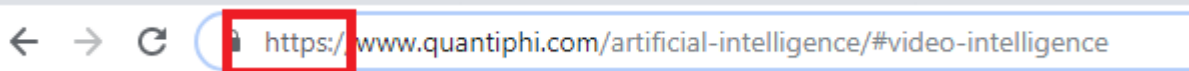
protocol://hostname/other_information

An example of a URL is

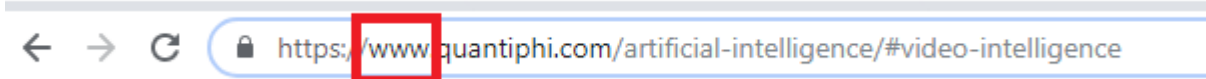
<https://www.quantiphi.com> , which is the URL for the QUANTIPHI website.

Below is additional information about each of the sections of the Quantiphi website URL

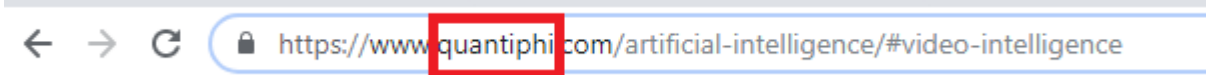
Scheme



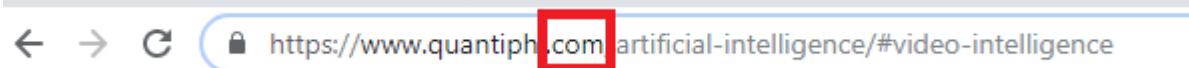
Subdomain



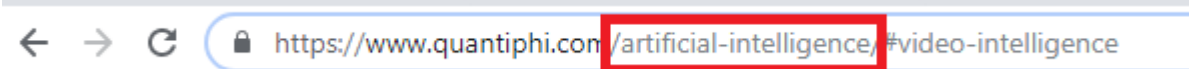
Second-level Domain



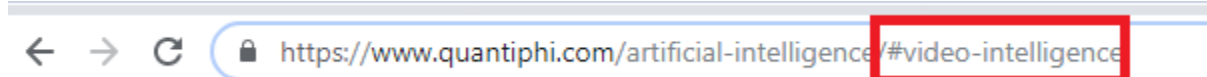
Top-level Domain



Subdirectory



Reference to Particular Id in the page



While all website URLs begin with "http," several other prefixes exist. Below is a list of various URL prefixes:

http – a webpage, website directory, or other file available over HTTP

ftp – a file or directory of files available to download from an FTP server

news – a discussion located within a specific newsgroup

telnet – a Unix-based computer system that supports remote client connections

gopher – a document or menu located on a gopher server

waits - a document or search results from a WAIS database

mailto - an email address (often used to redirect browsers to an email client)

file - a file located on a local storage device (though not technically a URL because it does not refer to an Internet-based location)

http:// or https://

The "http" stands for Hypertext Transfer Protocol and is what enables the browser to know what protocol it is going to use to access the information specified in the domain. An "https" protocol is short for "Hypertext Transfer Protocol Secure" and indicates that information transmitted over HTTP is encrypted and secure. After the http or https is the colon (:) and two forward slashes (//) that separate the protocol from the remainder of the URL.

url.htm

Finally, url.htm is the actual web page on the domain you're viewing. The trailing .htm is the file extension of the web page that indicates the file is an HTML file. Other common file extensions on the Internet include .html, .php, .asp, .cgi, .xml, .jpg, and .gif. Each of these file extensions performs a different function, like all the different types of files on your computer.

What characters are not allowed in a URL?

Most people realize that a space is not allowed in a URL. However, it is also important to realize, as documented in RFC 1738, the URL string can only contain alphanumeric characters and the !\$%&'()* characters. Any other characters that are needed in the URL must be encoded.

Below is additional information about a URL that points to the Computer Hope search page, with the search query of "example search".

<https://www.quantiphi.com/?s=Jobs>