IEEE
Benha university
Benha Student Branch

WEB FUNDAMENTALS REPORT

Submitted By:
Adham Mohammed Nabil

Submitted To:
Omar Ayoub

Table of Contents

Domain & IP

IP Address

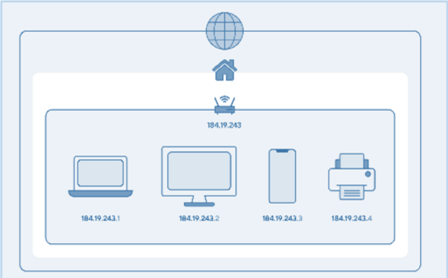
What is an IP Address?

An IP address (Internet Protocol address) is like a digital home address for any device connected to the internet or a local network.

Just like your street address you tell your friends so they can find you, an IP address tells the network where to send and receive data. Every phone, computer, router, or smart toaster on the internet has one.

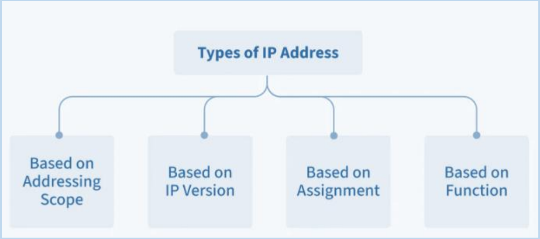
- ***IP (Internet Protocol)***: The set of rules governing how data travels across networks.
- ***Address***: The unique identifier assigned to a device so it can communicate.

> *So, when you open a website, your device says,*
>
> *“Hey, I’m 192.168.1.5 — send that page to me.”*
>
> *and the server replies,*
>
> *“Got it. Here’s your content, Mr. 192.168.1.5.”*



Types of IP Address

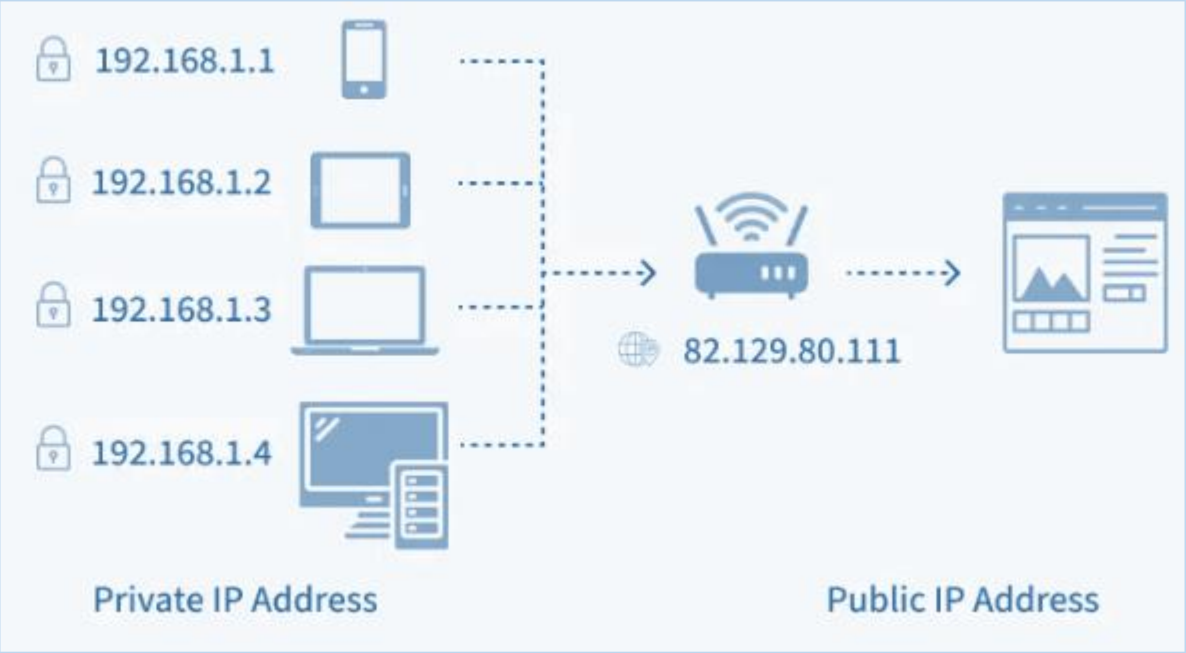
IP addresses can be classified in several ways based on their structure, purpose, and the type of network they are used in.



- ***Based on Addressing Scope:*** Our focus, where we classify them by where they’re used.
- ***Based on IP Version:*** As we learned in our session, we classify them by which internet protocol they follow whether it is IPv4 or IPv6.
- ***Based on Assignment:*** We classify them by how they’re given, either fixed manually (static) or automatically (dynamic).
- ***Based on Function:*** We classify them by what they do, whether they send data to one device, everyone, or a group.

Public VS Private IP Address

Every device on the Internet needs an address so data knows where to go.



Inside our home for example devices like phones or laptops get ***Private IP addresses***. These addresses only work inside your home. They can't be reached directly from the internet, the router manages this mini-network and decides which device gets which address.

We can think of ***Private IP*** as our college hall numbers "B2" these numbers only make sense inside our college.

Now our **Router** has both a ***Private IP*** inside our home network and a ***Public IP*** on the Internet, which is everyone outside sees.

When we visit a website our ***router*** sends and receive information using its ***Public IP***, then passes the data to your device through its ***Private IP***, so our Private IP is like the Street address our college located in, but we have a good Question:

Which IP Address will our client's website use so people can access it?

Short answer is that a client's website must use a ***public IP address***, it is obvious since we want anyone on the internet to see our site not only the people on the private network.

Basically the web server (Hosting provider), which we will talk about later assigns the site a ***Public IP address***, that IP is then linked to ***Domain name*** like *"example.com"* so when someone types it their browser looks up the ***Public IP*** through ***DNS (Domain Name System)*** and connects to that server.

But What is a Domain and DNS?

Domain

What is Domain Name?

We knew that every website has a public IP address, now imagine having to remember every one of those when you want to search your favorite websites – a messy mix of digits like *"142.250.190.78"*, you'll need a brain like a data center just to check your Email, That's where ***Domain*** come in to cover the chaos.

A domain name is a unique, easy-to-remember address used to access websites, such as 'google.com', and 'facebook.com'. Users can connect to websites using domain names thanks to the DNS system.

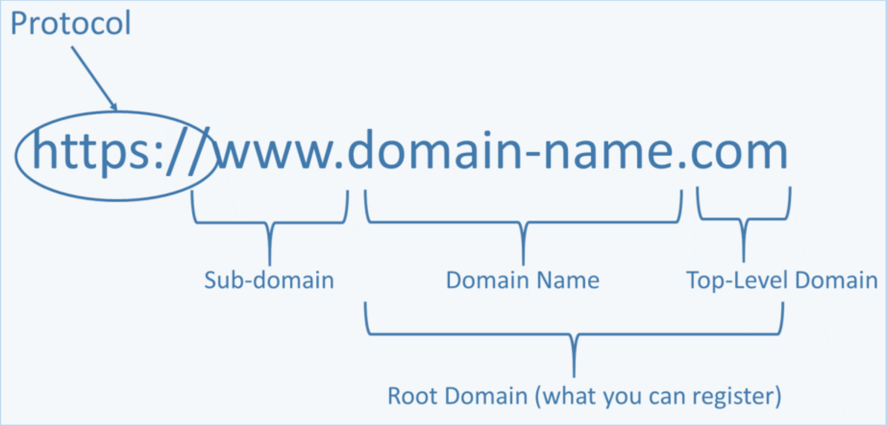
We can think of it like the Public IP is a phone number and the Domain name is the contact name you save for that number.

DNS (Domain Name System)

In order to translate the Domain name into the website real IP address we need ***DNS (Domain Name System).***

So when you type a domain in your browser for example *"google.com"* your device asks the DNS what is the IP for "*google.com*", DNS looks it up and replies with the correct IP so that your browser connects to that IP and load the site.

For web developers, understanding DNS is crucial, it's what makes your site actually reachable on the internet. Knowing how domains, DNS records, and hosting work together can save hours of debugging when something "mysteriously" won't load.


Protocol
https://www.domain-name.com
Sub-domain
Domain Name
Top-Level Domain
Root Domain (what you can register)

TLD (Top Level Domain)

The last part of a domain, like .com or .org, is called the Top-Level Domain (TLD) — it’s like the ‘surname’ of a web address, telling us a bit about its purpose or region.

TLDs play an important role in the DNS lookup process. For all uncached requests, when a user enters a domain name like ‘google.com’ into their browser window, another purpose of TLDs is to help classify and communicate the purpose of domain names. Every TLD will tell you something about the domain that precedes it, the 3 most common TLDs according to Google index analysis is:

- ***.com (Commercial):*** Originally meant for businesses, but it is the default choice for basically everything.
- ***.net (Network):*** initially used by networking and tech companies, still popular among tech-focused or infrastructure-related sites.
- ***.org (Organization):*** Often used by non-profits, communities, or open-source project, it tells people that the site’s purpose is more about service than sales, one of the closest examples is US ***https://www.ieee-bub.org/*** .

Types of TLDs

There are different types of TLDs but the 3 main and popular types are:

- ***Generic TLDs (gTLDs):*** These are the common ones you see everywhere like .com, .net, and .org. ICANN used to limit new ones, but since 2010, it opened the floodgates now we’ve got hundreds like .xyz, .top, and .tech.
- ***Country-code TLDs (ccTLDs):*** Each country gets its own two-letter domain ending. Examples include .uk (United Kingdom), .jp (Japan), and .eg (Egypt EG). These are managed locally but coordinated globally by ICANN’s IANA division.
- ***Sponsored TLDs (sTLDs):*** These are special-purpose domains run by specific communities or organizations. For instance, .app (sponsored by Google) is for developers, and .gov is reserved for U.S. government use.

Also there are some Specialized TLDs like *“.tech”* instantly signals what your site is about. For startups or developers, it creates a brand identity and improves memorability. Compare ***“innovate.tech”*** to “***innovatetechnologies.com”*** shorter, cleaner, and sharper.

Hosting

What is Hosting?

Web hosting is essentially renting space on a server to store your website's files and make them accessible online. Different hosting types—Shared, VPS, and Cloud—offer varying levels of performance, cost, and control, depending on the user's needs.

Types of Hosting

Hosting Type	Shared Hosting	VPS (Virtual Private Server)	Cloud Hosting
Cost	$2-10/month	$10-100/month	$10-200/month
Performane	Low – resources shared with many sites	Medium – dedicated resources on a shared server	High – scalable and distributed resources
Recommended For	Beginners, personal blogs, and small websites with low traffic	Growing websites, developers, and online stores that need more control and resources than shared hosting	Businesses with fluctuating traffic or high-resource demands that need scalable solutions

Bandwidth

Bandwidth in hosting means the amount of data your website can transfer to visitors within a specific time (usually per month).it is like your internet data plan, the more traffic and heavier your pages (images, videos, etc.), the more bandwidth you use, if a website exceeds its bandwidth limit, it may slow down, become temporarily inaccessible, or display a "Bandwidth Limit Exceeded" message until the next billing cycle or upgrade. In business terms, that's lost traffic, lost trust, and potential revenue loss.

Security

SSL and TLS

Difference between SSL and TLS

Secure Sockets Layer (SSL) is an older security protocol designed to establish an encrypted connection between two devices or applications over a network. It was once the standard for securing online communications, allowing browsers and servers to exchange data privately and verify each other's authenticity. However, SSL contained several vulnerabilities and has since been replaced by its successor, ***Transport Layer Security (TLS)***. TLS improves upon SSL by offering stronger encryption, better authentication, and overall enhanced security performance. Today, TLS is the modern standard used to secure internet communications, even though many people still casually refer to it as "SSL."

Similarities between SSL and TLS

Both ***SSL*** and ***TLS*** serve the same fundamental purpose: ensuring secure, encrypted data transfer between servers, applications, and users. These protocols authenticate both parties in a network connection to confirm their identity before any data exchange occurs.

So, Long story short, TLS is the direct successor to SSL, and all SSL versions are now officially deprecated. However, the term "SSL" remains widely used in marketing and technical documentation to describe TLS-secured connections or certificates. In modern web infrastructure, when you see "SSL" or "SSL/TLS," it almost always refers to TLS technology.

HTTPS and Its Importance for SEO

What is HTTPS?

HTTPS (HyperText Transfer Protocol Secure) is the secure version of HTTP used to safely transmit data between a user and a web server. It encrypts information such as passwords and payment details, preventing hackers from intercepting or tampering with it. HTTPS is essential for secure browsing, especially on public Wi-Fi and for sensitive operations like online banking or shopping.

HTTP vs HTTPS

Both ***HTTP*** and ***HTTPS*** define how data is transferred online, but ***HTTP*** sends data in plain text, making it vulnerable to interception. ***HTTPS*** adds ***SSL/TLS*** encryption, securing communication between the browser and the server.

Importance of HTTPS for SEO

Beyond securing data, HTTPS directly impacts a site's search ranking and performance. Google favors HTTPS-enabled sites because they:

- ***Build User Trust:*** Secure sites keep visitors engaged and confident, reducing bounce rates.
- ***Preserve Referral Data:*** HTTPS keeps analytics accurate by maintaining referral source information.
- ***Boost Performance:*** HTTPS enables HTTP/2, improving site speed which is a key ranking factor.
- ***Gain SEO Advantage:*** Google treats HTTPS as a ranking signal, giving secure sites a small boost.

Search Engine Optimization

What is SEO?

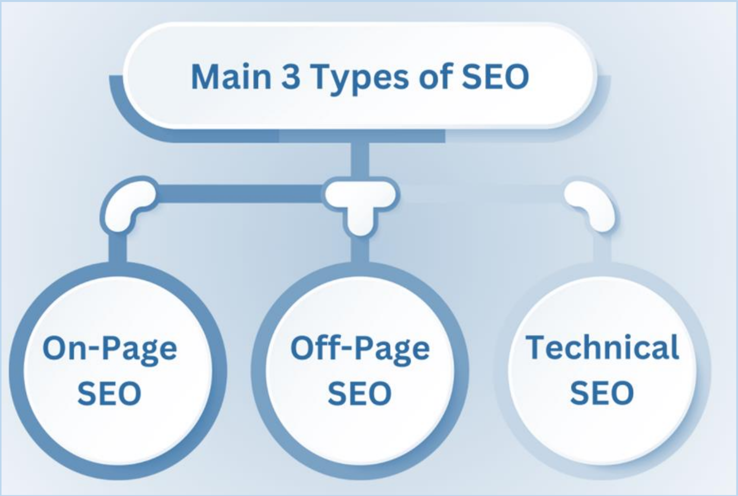
SEO (Search Engine Optimization) is about making your website easier for search engines to understand and rank. Its goal is to appear on the first page of results for relevant keywords, driving real, interested visitors to your site. As technology and search behavior evolve, from voice searches to AI summaries ***SEO*** stays key to keeping your website visible and competitive.

Types of SEO

Imagine SEO as a sports team. To win, you need a strong offense and defense. But you also need fans (an audience).

Think of technical optimization as your defense, content optimization as your offense, and off-site optimization as ways to attract, engage and retain a loyal fan base:

- ***Technical SEO:*** Optimizing the technical aspects of a website.
- ***On-page SEO:*** Optimizing the content on a website for users and search engines.
- ***Off-page SEO:*** Creating brand assets (e.g., people, marks, values, vision, slogans, catchphrases, colors) and doing things that will ultimately enhance brand awareness and recognition (i.e., demonstrating and growing its expertise, authority and trustworthiness) and demand generation.



On-Page SEO

On-Page SEO focuses on optimizing the elements within a website to improve its visibility and ranking on search engines like Google and Bing. It involves enhancing both the content and the technical structure of web pages to make them easier for search engines to understand and more useful for visitors.

Key Elements of On-Page SEO:

- ***Content Quality:*** Create original, engaging content that includes relevant keywords naturally.
- ***Keyword Optimization:*** Use main and secondary keywords in headings, paragraphs.
- ***Meta Tags:*** Write clear, keyword-rich titles and descriptions.
- ***URL Structure:*** Keep URLs short, descriptive, and relevant to the page content.
- ***Internal Linking:*** Link related pages to improve navigation and SEO value.
- ***User Experience (UX):*** Ensure fast loading, mobile compatibility, and easy navigation.

Off-Page SEO

Off-Page SEO refers to all optimization activities performed outside a website to improve its visibility, authority, and ranking in search engine results pages (SERPs). While many associate it primarily with link building, Off-Page SEO extends far beyond that. It focuses on enhancing a site’s credibility and trustworthiness through external factors such as backlinks, social engagement, and brand reputation.

Key Elements of Off-Page SEO

- ***Backlinks:*** Acquire high-quality links from reputable websites to build domain authority and improve rankings.
- ***Social Media Marketing:*** Promote and share content across social platforms to increase reach and brand awareness.
- ***Reputation Management:*** Monitor and strengthen how the brand is perceived online to build trust and reliability.
- ***Influencer Marketing:*** Partner with influencers or industry leaders to expand exposure and credibility.
- ***Brand Mentions:*** Earn mentions on trusted platforms, even without direct links, to enhance visibility and authority.
- ***Guest Blogging:*** Contribute to other websites to gain backlinks, attract new audiences, and build reputation.

Meta Description

What is Meta Description?

A ***meta description*** is a short snippet of text that summarizes a webpage's content. It appears below the title tag on a search engine results page (SERP) and helps users understand what the page is about before clicking. Though it doesn't directly affect rankings, it plays a key role in attracting clicks and improving engagement. A strong meta description should be concise (around 150–160 characters), relevant, and persuasive, clearly telling users what to expect while sparking curiosity. In short, it serves as a mini advertisement for your webpage, influencing user decisions and **indirectly boosting SEO performance through higher click-through rates (CTR)**.

Benefits of a Meta Description

- ***Improves Click-Through Rates (CTR):*** A well-written meta description encourages users to click on your result over others.
- ***Enhances User Experience:*** Clear, informative descriptions align user expectations with your page content.
- ***Boosts Visibility:*** When users engage more with your listing, it signals quality and relevance to search engines.
- ***Supports Brand Credibility:*** Consistent and trustworthy descriptions strengthen your site's professional image.

Best Practices for Writing Meta Descriptions

- ***Keep it concise:*** Stay within 150–160 characters to avoid truncation in search results.
- ***Match user intent:*** Write with the searcher's goal in mind — solve a problem, answer a question, or offer value.
- ***Include relevant keywords naturally:*** Keywords don't impact rankings directly but appear bolded when matching a search query, increasing visibility.
- ***Add a soft call to action:*** Use phrases like "Learn more," "Discover," or "Get started today" to encourage clicks.
- ***Reflect the page content accurately:*** Avoid vague or misleading text that can drive users away.
- ***Update regularly:*** Review and adjust descriptions as your content evolves to maintain accuracy and relevance.

Accessibility (A11y)

Screen Reader

What is screen reader?

A screen reader is an assistive technology that helps visually impaired users access digital content by converting on-screen elements—like text, buttons, and images—into speech or braille. Instead of reading the page's visual layout, it interprets the HTML structure (DOM) and semantic tags such as headings, links, and alt text, allowing users to navigate and understand a webpage through audio or tactile feedback.

How do screen readers work?

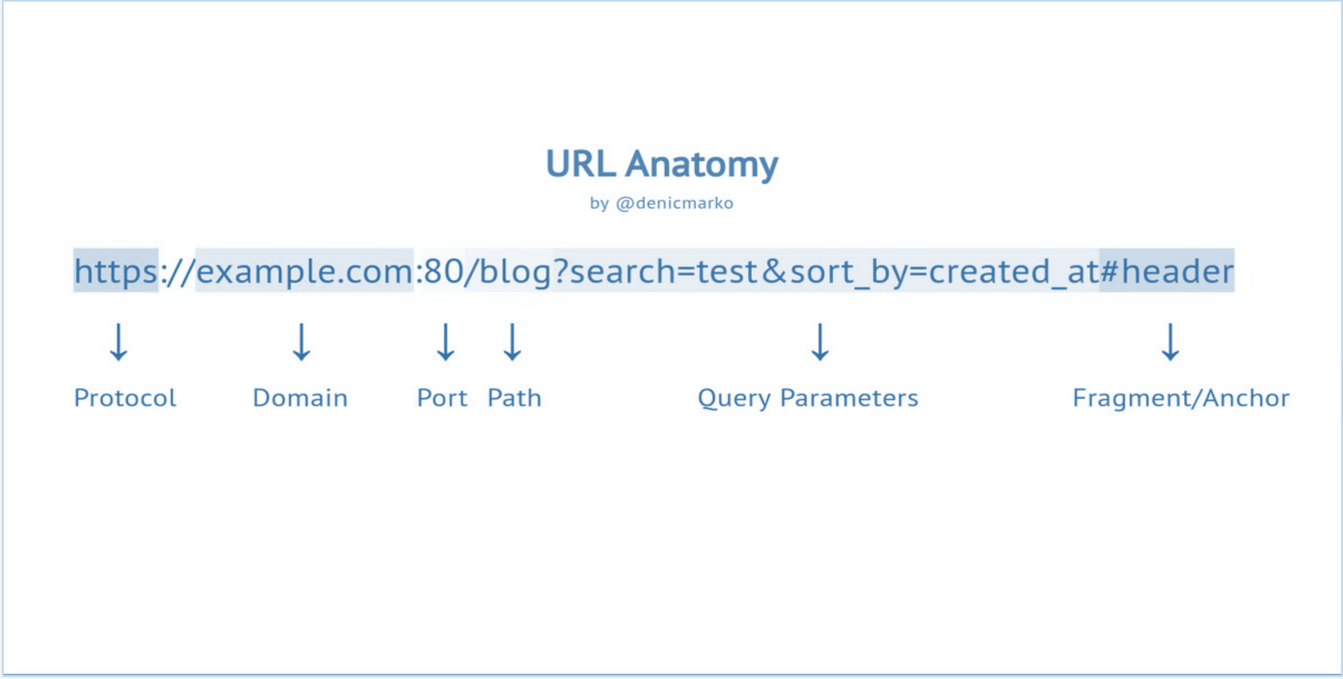
- ***On desktop systems:*** users commonly navigate via the keyboard, jumping through items such as headings, forms, links and landmarks (for example using the "h" key to skip between headings in some readers).
- ***On mobile devices:*** navigation is often via touch or swipe gestures, dragging a finger across the screen to hear what's under focus, or swiping left/right to move to the next or previous item.

Common accessibility issues

- ***Insufficient color contrast:*** Low contrast between text and background makes content unreadable for users with low vision or colour-blindness.
- ***Poor or missing alt text for images:*** When images lack descriptive alt text (or have none at all), screen reader users lose context and meaning.
- ***Incorrect use of headings:*** Headings not structured hierarchically (e.g., skipping from <h1> to <h4>) or used purely for style break navigation for assistive users.
- ***Lack of keyboard accessibility:*** When interactive elements (menus, forms, buttons) don't support keyboard navigation (Tab, Enter, arrow keys), users who can't use a mouse are excluded.
- ***Inaccessible forms:*** Forms with unlabeled fields, ambiguous instructions, or non-accessible error feedback create significant barriers for users relying on assistive tech.
- ***Non-descriptive link text:*** Links using vague labels like "click here" or "read more" fail to provide context to screen reader users about the destination or action.

The URL

The Anatomy of a URL



URL Anatomy

by @denicmarko

https://example.com:80/blog?search=test&sort_by=created_at#header

As we go on with this report, we've already explored the protocol — the rule set that defines how data is transferred — and the domain, which identifies the website's main address. A typical URL also includes a port number, which specifies the communication gateway (though it's often hidden when using default ports like 80 for HTTP or 443 for HTTPS), and a path, which directs the browser to a specific page or resource within the website's structure.

But beyond these familiar parts, URLs can get more dynamic. That's where query parameters and fragments come in — two small but powerful components that help websites deliver more specific and interactive user experiences.

Query String

A query string is the portion of a URL that begins with a question mark (?) and carries data to the web application via key-value pairs separated by ampersands (&). It enables the server (or application) to act based on specific parameters you're passing in the request—such as filtering results, tracking user behavior, managing sessions, or customizing content.

> *In an e-commerce context, the query string might look like:*
>
> *"?category=electronics&sort=price_desc&brand=Acme"*
>
> *Here, "category", "sort", "brand" are keys, and they instruct the website to show only "Electronics" items, sorted by descending price, and limited to the "Acme" brand.*
>
> *You'll want to explain how this lets the front-end send instructions and how the back-end uses them, Also mention security implications: because the query string is visible in the URL, data passed can be exposed or manipulated, so encoding, validation, and sanitization are required.*

Fragment

A ***fragment*** is the portion of a URL that starts with the hash symbol (#). According to MDN, the fragment "is the last part of the URI, starting with the # character. It is used to identify a specific part of the resource, such as a section of a document or a position in a video."

Crucially: the fragment is not sent to the server when you make the HTTP request. It lives entirely on the client-side (the browser), which uses it to scroll to or highlight a part of the page after the page loads.

Example:

> *https://example.com/faq#returns-policy*
>
> *In this URL, the #returns-policy fragment tells the browser to scroll directly to the "Returns Policy" section on the FAQ page.*

Honourable Mentions: Because the Web Doesn’t Stop There

Browser Rendering

Browser rendering is the process by which web browsers convert raw files (HTML, CSS, JavaScript) into the actual visual page that users interact with. The browser engine builds a ***DOM (Document Object Model)*** and ***CSSOM (CSS Object Model)***, merges them into a render tree, calculates layouts, and finally paints pixels on screen.

Rendering is critical because it directly impacts how quickly and effectively users see and interact with content. A page that takes too long to render hampers user experience (leading to higher bounce rates) and can also hurt ***SEO ranking***. Developers must therefore optimize rendering workflows (reduce heavy JavaScript, streamline CSS, minimize layout recalculations) to deliver fast-loading, interactive web applications.

Cookies

Cookies are small files of information that a web server sends to a user’s browser, which the browser then stores and sends back with subsequent requests.

Essentially, when you visit a website, the server can drop a cookie into your browser so that it can ***“remember”*** something about you — such as your login status, items in your shopping cart, or preferences.

Cookies are critical because they enable persistent ***user-experience*** and ***state management*** across web sessions: without them, every page visit would treat you as a brand-new user. They support things like ***user authentication*** (so users don’t need to log in at every step), ***personalization*** (so the site remembers your choices), and ***analytics/tracking*** (so developers understand user behavior). At the same time, cookies introduce privacy and security concerns (tracking, third-party cookies, consent) which developers must handle responsibly.

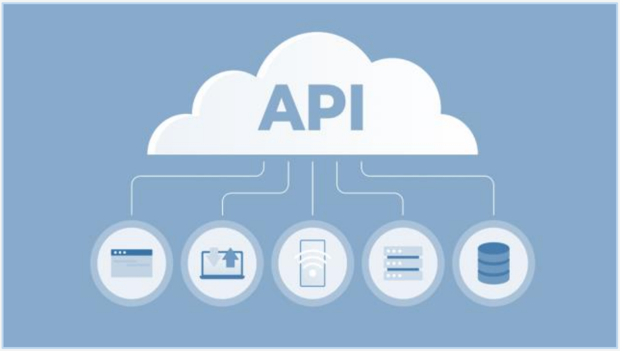
APIs (Application Programming Interface)

An ***API*** is a defined set of rules and protocols that enable different software systems to communicate with one another — essentially acting as a “bridge” between a client and a server.

For example, when a web application requests data (say, a list of users) from a back-end service, the ***API*** governs how that request should be formatted, transmitted, processed and responded to, abstracting away the internal complexities of the server.

APIs are fundamental to modern web development because they allow developers to leverage existing services (data retrieval, authentication, payment processing, etc.) instead of building everything from scratch.

They promote modularity, scalability and interoperability — meaning a front-end built in React (for instance) can request data from a back-end built in whatever stack, as long as the API contract is honoured — which accelerates development and streamlines maintenance.



CDN (Content Delivery Network)

A ***Content Delivery Network (CDN)*** is a globally distributed network of servers—also called “edge” servers—that cache and deliver website content from the server location closest to the end user. When a user requests a page, the ***CDN*** routes the request to the nearest edge server rather than a single origin server, reducing latency and speeding up delivery of static resources like images, JavaScript files and stylesheets.

For modern web development—especially when targeting users in multiple geographic regions—a ***CDN*** is a key performance and reliability enhancer. It decreases load times (thus improving user experience and reducing bounce rates), lowers the strain on your origin server (improving scalability), and strengthens resilience against traffic spikes or distributed attacks. By integrating a ***CDN*** into your front-end strategy (for instance when building your React + Tailwind site), you ensure that users everywhere enjoy faster access and that your architecture remains robust under real-world conditions.

References

- https://www.geeksforgeeks.org/computer-science-fundamentals/what-is-an-ip-address/
- https://www.cloudflare.com/learning/dns/what-is-dns/
- https://www.cloudflare.com/learning/dns/top-level-domain/
- https://www.scoutdns.com/100-most-popular-tlds-by-google-index/
- https://www.hostinger.com/tutorials/what-is-web-hosting#How_Much_Does_It_Cost_to_Host_a_Website
- https://www.netlify.com/blog/what-is-website-bandwidth/
- https://aws.amazon.com/compare/the-difference-between-ssl-and-tls/
- https://www.marketingminer.com/en/blog/what-is-https
- https://searchengineland.com/guide/what-is-seo#h-types-of-seo-and-specializations
- https://www.geeksforgeeks.org/techtips/difference-between-on-page-and-off-page-seo/
- https://searchengineland.com/seo-meta-descriptions-everything-to-know-447910
- https://axesslab.com/what-is-a-screen-reader/
- https://www.a11y-collective.com/blog/how-to-overcome-avoid-the-7-common-digital-accessibility-issues/
- https://developer.mozilla.org/en-US/docs/Learn_web_development/Howto/Web_mechanics/What_is_a_URL
- https://www.twingate.com/blog/glossary/query-string
- https://www.seobility.net/en/wiki/Rendering
- https://www.cloudflare.com/learning/privacy/what-are-cookies/
- https://www.cloudflare.com/learning/privacy/what-are-cookies/
- https://developer.mozilla.org/en-US/docs/Web/URI/Reference/Fragment