



Introduction to Web and Web Design

ABSTRACT

This unit introduces the Internet, World Wide Web (WWW), and the basics of how they work through browsers, HTTP/HTTPS, and client-server architecture. It explains URLs, request/response messages, and common status codes. The unit also covers key web design principles like simplicity, readability, responsiveness, navigation, and load speed. Common design problems and solutions are highlighted, along with page layouts (flexible and fixed) to ensure websites are user-friendly and visually appealing.

❖ Internet

- Internet is a global communication system that links together thousands of individual networks. It allows exchange of information between two or more computers on a network.
- Internet helps in transfer of messages through mail, chat, video & audio conference, etc.
- It has become mandatory for day-to-day activities: bills payment, online shopping and surfing, tutoring, working, communicating with peers, etc.

ARPANET (Advanced Research Projects Agency Network) was the first computer network that used packet switching to connect geographically dispersed computers and laid the foundation for the development of the internet. It was developed by the United States Department of Defense's Advanced Research Projects Agency (ARPA) in the late 1960s and early 1970s to connect computers and researchers at various universities and research institutions.

ARPANET was also the birthplace of many of the technologies that are now essential to the modern internet. For example, the **first email message was sent over ARPANET in 1971**, and the **first online chat system was developed on the network in 1973**. In addition, **ARPANET was the first network to use the TCP/IP protocol suite**.

❖ World Wide Web (WWW)

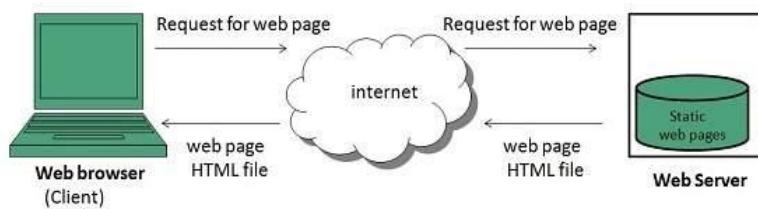
- ✓ The **World Wide Web** also known as the **Web** and the Internet are fundamentally dissimilar from one another because the **Web is a collection of information that can be accessed using the Internet, whereas the Internet is a global network of networks that offers access to almost all types of information.**
- ✓ A program that utilizes the internet is called the WWW. It refers to the extensive network of linked online pages.
- ✓ Hyperlinks are used to connect these sites. As a result, the user may navigate effortlessly between pages to find the information they need.
- ✓ A website is a collection of connected web pages. Accessing webpages or websites on the World Wide Web requires a web browser and the HTTP protocol.
- ✓ The HTTP protocol is a collection of guidelines for transporting data on the World Wide Web, including text, photos, audio, video, and other multimedia assets.

- In 1989, **Tim Berners Lee** began to develop a technology for sharing information via hyperlinked text documents.
- In **October-1994 Tim Berners Lee** founded an organization called **W3C (World Wide Web Consortium)** devoted to developing technologies for www. His invention called **HTML (Hypertext Markup Language)** and the communication protocol **HTTP (Hypertext Transfer Protocol)** used to send information over the web. (Year-2014-HTML5

❖ Difference between the World Wide Web and the Internet

Internet	WWW
The Internet allows you to link your computer to any other computer on the planet.	The World Wide Web is a collection of information accessible through the Internet.
The Internet is a worldwide network of interconnected computer networks that connect devices using the TCP/IP protocol.	The World Wide Web refers to HTML-formatted online material that may be accessed using the HTTP protocol.
The Internet can be compared to a large bookstore.	The web can be considered as a store with a collection of books.
Internet is superset of WWW.	The World Wide Web is a subset of the Internet.
It first appeared in the late 1960s.	Tim Berners-Lee, an English scientist, created the World Wide Web in 1989.
The Internet is mostly based on hardware.	In comparison to the Internet, the WWW is more software-oriented.

❖ Web Browser



- ✓ The web browser is an application software to explore www.
- ✓ It provides an interface between the server and the client and requests to the server for web documents and services.
- ✓ It works as a compiler to render HTML which is used to design a webpage.
- ✓ Whenever we search for anything on the internet, the browser loads a web page written in HTML, including text, links, images, and other items such as style sheets and JavaScript functions.
- ✓ Google Chrome, Microsoft Edge, Mozilla Firefox, Safari are examples of web browsers.
- ✓ A web browser is used to surf the internet or do online browsing.
- ✓ A certain Web site's look may vary slightly depending on the browser being used.

Hyper Text Transfer Protocol(HTTP)

- It is a protocol used to access the data on the World Wide Web (www).
- The HTTP protocol can be used to transfer the data in the form of plain text, hypertext, audio, video, and so on.
- This protocol is known as Hyper Text Transfer Protocol because of its efficiency that allows us to use in a hypertext environment where there are rapid jumps from one document to another document.
- HTTP is used to carry the data in the form of MIME-like format.

MIME (Multipurpose Internet Mail Extensions) is an extension lets users exchange different kinds of data files, including audio, video, images and application programs.

Features of HTTP:

- **Connectionless protocol:** HTTP is a connectionless protocol. HTTP client initiates a request and waits for a response from the server. When the server receives the request, the server processes the request and sends back the response to the HTTP

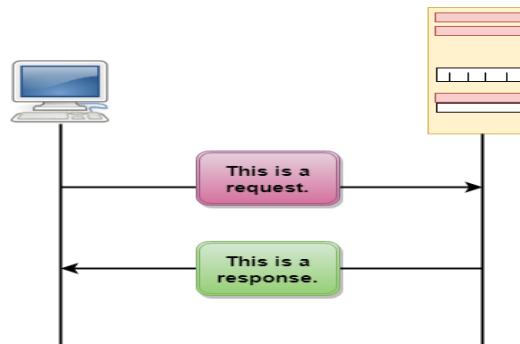
client after which the client disconnects the connection. The connection between client and server exist only during the current request and response time only.

- **Media independent:** HTTP protocol is a media independent as data can be sent as long as both the client and server know how to handle the data content. It is required for both the client and server to specify the content type in MIME-type header.

MIME stands for **Multipurpose Internet Mail Extensions**. It is an internet standard that extends the format of email messages to support text in character sets other than ASCII, as well as attachments like images, audio, video, and other types of files. MIME is widely used in email, but also applies to other Internet protocols, such as HTTP, where it helps define the format of content.

- **Stateless:** HTTP is a stateless protocol as both the client and server know each other only during the current request. Due to this nature of the protocol, both the client and server do not retain the information between various requests of the web pages.

The figure shows the HTTP transaction between client and server. The client initiates a transaction by sending a request message to the server. The server replies to the request message by sending a response message.



❖ HTTP and HTTPS

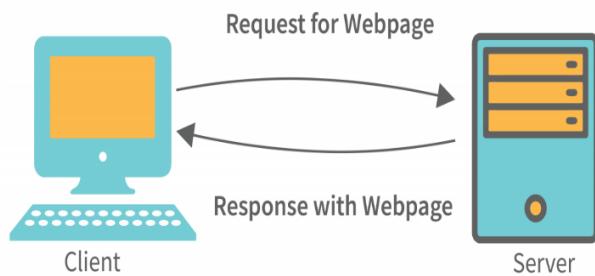
HTTP vs HTTPS		HTTP	HTTPS
Feature			
Full Form	HyperText Transfer Protocol	HyperText Transfer Protocol Secure	
Security	Not secure – data sent in plain text	Secure – data encrypted with SSL/TLS	
URL Prefix	http://	https://	
Browser Indication	Shows “Not Secure” !	Shows lock icon 🔒	
Data Transfer	Anyone can read data (like a postcard)	Data is encrypted (like a sealed envelope)	
SEO Ranking	Lower ranking (Google discourages it)	Higher ranking (Google prefers HTTPS)	
Use Cases	Old websites, testing environments	Banking, shopping, login systems, all modern sites	
Real-Time Example	http://neverssl.com → works but marked “Not Secure”	https://github.com or https://wikipedia.org → secure with lock	

❖ What is Client-Server Architecture?

The client-server model is a way of organizing how computers or programs talk to each other.

Client – The client is like the user side. It asks for something. For example, when you open a website in your browser, the browser is the client.

Server – The server is like the provider side. It gives the client what it asks for. In the website example, the server stores the web pages and sends them to your browser when requested.



How it works:

1. The client sends a request (e.g., "Give me this web page").
2. The server processes the request and sends a response (e.g., "Here is the page").
3. Clients and servers can be on the same computer or connected through a network (like the internet).

Example:

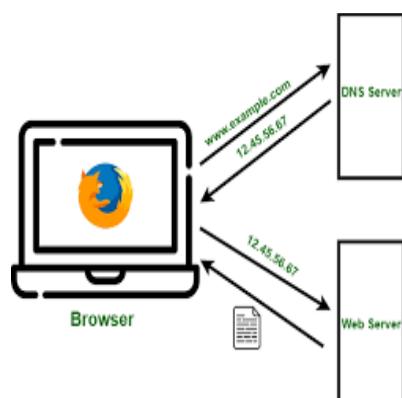
When you use Google:

Your phone/computer (client) asks Google for search results.

Google's computers (servers) find the results and send them back to your device.

Client = asks, Server = provides.

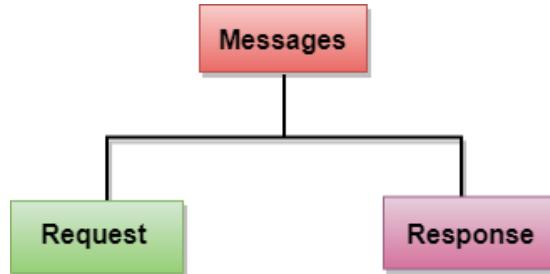
❖ How the browser interacts with the servers?



- ✓ User opens browser and enters the URL of the website. The Browser then requests the DNS(DOMAIN NAME SYSTEM) Server.
- ✓ DNS Server lookup for the IP address of the WEB Server.
- ✓ DNS Server responds with the IP address of the WEB Server.
- ✓ Browser sends over an HTTP/HTTPS request to WEB Server's IP (provided by DNS server).
- ✓ Server sends over the necessary pages of the website.
- ✓ Browser then renders the pages and the website is displayed. This rendering is done with the help of DOM (Document Object Model) interpreter, CSS interpreter and JS Engine collectively known as the JIT or (Just in Time) Compilers.

HTTP Message

HTTP messages are of two types: request and response. Both the message types follow the same message format.



 **Request Message:** The request message is sent by the client that consists of a **request line**, **headers**, and sometimes a **body**.

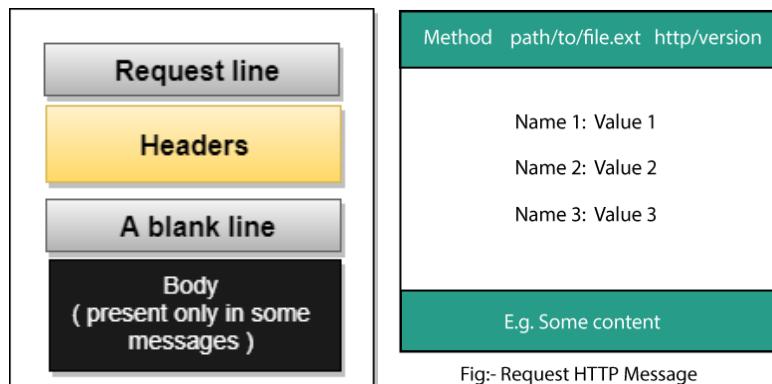


Fig:- Request HTTP Message

➤ Initial Request Line

The initial line is different for the request and for the response. A request-line consists of three parts: a **method name**, **requested resource's local path**, and the **HTTP version** being used. All these parts are separated by spaces.

Syntax:

GET /path/to/file/index.html HTTP/1.0

- **GET** is the most common HTTP method.
- The **path** shows the part of the URL after the host name. It is also called a request **URI**.
- The **version** of HTTP always takes the form "**HTTP/x.x**", uppercase.

In the request line message, the request line is first line. It contains:

- a) Request Method: An HTTP method, a verb (like GET, PUT or POST) or a noun (like HEAD or OPTIONS), that describes the action to be performed.
- b) Request URL: the absolute path of the protocol, port, and domain are usually characterized by the request context.
- c) HTTP Version: which defines the structure of the remaining message, acting as an indicator of the expected version to use for the response.

➤ Message Headers

The Message header provides information about the request and response. It also provides information about the object which is sent in the message body. Message Headers are of four types:

- General Header: It has general applicability for both request messages and response messages.
- Request Header: It has applicability only for the request messages.
- Response Header: It has applicability only for the response messages.
- Entity Header: It defines meta-information about the entity-body, and about the resource identified by request.

➤ Message Body

The message body of an HTTP message is used to carry the entire body associated with the request and response. The message-body differs from the entire-body only when a transfer-coding has been applied, as indicated by the Transfer-Encoding header field.

 **Response Message:** The response message is sent by the server to the client that consists of a **status line**, **headers**, and sometimes a **body**.

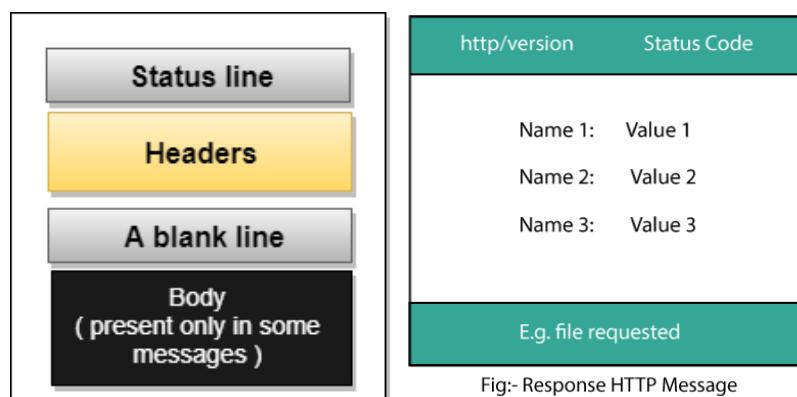


Fig:- Response HTTP Message

➤ Initial Response Line

The **initial Response line** is also known as the **status line**. It also has three parts: the HTTP version, a response status code that gives the result of the request, and the English reason phrase describing the status code.

In the response message, the status line is the first line. The status line contains three items:

a) HTTP Version Number b) Status Code c) Reason Phrase

Example:

HTTP/**1.0 200 OK**

or

HTTP/**1.0 404 Not Found**

Here,

The HTTP version of the response line and request line are the same as "HTTP/x.x"

HTTP/1.1 200 OK

HTTP version	Status Code	Reason Phrase
-----------------	----------------	------------------

HTTP status codes

HTTP defines these standard status codes that can be used to convey the results of a client's request. The status codes are divided into five categories.

- 1xx: Informational – Communicates transfer protocol-level information.
- 2xx: Success – Indicates that the client's request was accepted successfully.
- 3xx: Redirection – Indicates that the client must take some additional action in order to
- complete their request.
- 4xx: Client Error – This category of error status codes points the finger at clients.
- 5xx: Server Error – The server takes responsibility for these error status codes.

100 Continue

The client SHOULD continue with its request. This interim response is used to inform the client that the initial part of the request has been received and has not yet been rejected by the server.

200 OK

The request succeeded. The result meaning of "success" depends on the HTTP method

301 Moved Permanently

The URL of the requested resource has been changed permanently. The new URL is given in the response.

400 Bad Request

The server cannot or will not process the request due to something that is perceived to be a client error (e.g., malformed request syntax, invalid request message framing, or deceptive request routing).

401 Unauthorized

Although the HTTP standard specifies "unauthorized", semantically this response means "unauthenticated". That is, the client must authenticate itself to get the requested response.

402 Payment Required

This response code is reserved for future use. The initial aim for creating this code was using it for digital payment systems, however this status code is used very rarely and no standard convention exists.

403 Forbidden

The client does not have access rights to the content; that is, it is unauthorized, so the server is refusing to give the requested resource. Unlike 401 Unauthorized, the client's identity is known to the server.

404 Not Found

The server cannot find the requested resource. In the browser, this means the URL is not recognized. In an API, this can also mean that the endpoint is valid but the resource itself does not exist. Servers may also send this response instead of 403 Forbidden to hide the existence of a resource from an unauthorized client. This response code is probably the most well known due to its frequent occurrence on the web.

502 Bad Gateway

This error response means that the server, while working as a gateway to get a response needed to handle the request, got an invalid response.

503 Service Unavailable

The server is not ready to handle the request. Common causes are a server that is down for maintenance or that is overloaded.

❖ Uniform Resource Locator (URL)

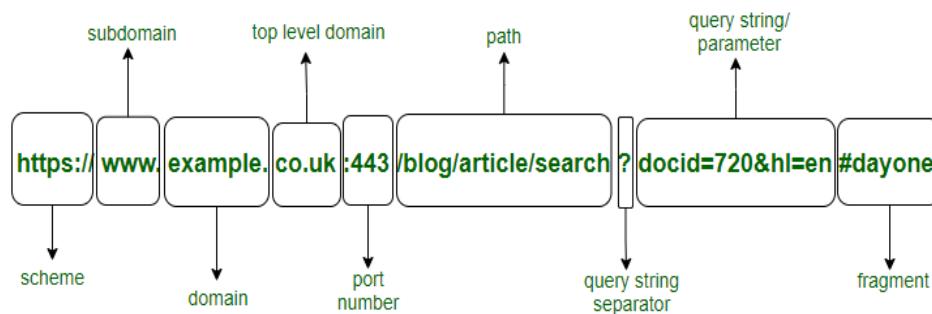
- A client that wants to access the document in an internet needs an address and to facilitate the access of documents, the HTTP uses the concept of Uniform Resource Locator (URL).
- The Uniform Resource Locator (URL) is a standard way of specifying any kind of information on the internet.
- The URL defines four parts: method, host computer, port, and path.



- **Method:** The method is the protocol used to retrieve the document from a server. For example, HTTP.
- **Host:** The host is the computer where the information is stored, and the computer is given an alias name.
- Web pages are mainly stored in the computers and the computers are given an alias name that begins with the characters "www". This field is not mandatory.
- **Port:** The URL can also contain the port number of the server, but it's an optional field. If the port number is included, then it must come between the host and path and it should be separated from the host by a colon.
- **Path:** Path is the pathname of the file where the information is stored. The path itself contains slashes that separate the directories from the subdirectories and files.

Parts of a URL

URL : `https://www.example.co.uk:443/blog/article/search?docid=720&hl=en#dayone`



I. Scheme/Protocol :

The protocol or scheme part of the URL and indicates the **set of rules that will decide the transmission and exchange of data**.

- ✓ **Hypertext transfer protocol (HTTP)** – allows users to access data from web pages using hypertext links.
- ✓ **Hypertext transfer protocol secure (HTTPS)** – protects data transmission between the user's browser and site.
- ✓ **File transfer protocol (FTP)** – used for a file transfer between different devices via a network.

In example **https** is used. HTTPS stands for Hyper Text Transfer Protocol Secure tells the browser to **display the page in Hyper Text (HTML) format** as well as **encrypt any information** that the user enters in the page.

II. Subdomain :

`https://www.`

The subdomain is used to separate different sections of the website as it **specifies the type of resource** to be delivered to the client. Here the subdomain used 'www' is a general symbol for any resource on the web. Subdomains like 'blog' direct to a blog page, 'audio' indicates the resource type as audio.

III. Domain Name :

`https://www.example.`

Domain name **specifies the organization or entity** that the URL belongs to. Like in `www.facebook.com` the domain name 'facebook' indicates the organization that owns the site.

IV. Top-level Domain :

<https://www.example.co.in>

The TLD (top-level domain) **indicates the type of organization the website is registered to.** Like the **.com** in www.facebook.com indicates a **commercial entity**. Similarly, **.org** indicates organization, **.co.in** a **commercial entity in the India.**

V. Port Number :

<https://www.example.co.in:443>

A port number specifies **the type of service** that is requested by the client since **servers often deliver multiple services**. Some default port numbers include 80 for HTTP and 443 for HTTPS servers.

VI. Path :

[https://www.example.co.in:443/blog/article/search](https://www.example.co.in:443/<u>blog/article/search</u>)

Path specifies the exact location of the web page, file, or any **resource that the user wants access to**. Like here the path indicates a specific article in the blog webpage.

VII. Query String Separator :

<https://www.example.co.in:443/blog/article/search?>

The **query string** which contains specific parameters of the search is **preceded by a question mark (?)**. The question mark tells the browser that **a specific query is being performed**.

VIII. Query String :

<https://www.example.co.in:443/blog/article/search?docid=720&hl=en>

The query string **specifies the parameters of the data that is being queried from a website's database**. Each query string is **made up of a parameter and a value** joined by the equals (=) sign. In case of multiple parameters, query strings are joined using the ampersand (&) sign. The parameter can be a number, string, encrypted value, or any other form of data on the database.

IX. Fragment :

<https://www.example.co.in:443/blog/article/search?docid=720&hl=en#dayone>

The fragment identifier of a URL is **optional**, usually appears at the end, and begins with a hash (#). It indicates a **specific location within a page such as the 'id' attribute for an HTML element**.

Concepts of effective Web Design:

❖ Have a Purpose and Plan

- ✓ Your website and each page of it should have a purpose and clear specification that what it does. Before you jump to the code figure out what's your user's expectation is, what pages they are going to view, what content they are going to read and according to that find out where you want elements to go.

❖ Simplicity

- ✓ Complexity is not good in web designing. ***less-is-more*** approach work more effectively in web design. Make your website simple and try to display one detail at a time for getting the individual attention of each part.
- ✓ Here are some tips for designing a clean and simple website.

- Don't overload the information at home page on your website. If there is too much content or information, keep them on inner pages.
- Also, make your website consistent in terms of font-family, font-size, colors, images, headings and other aspects.
- Include white spaces in your design. Give enough padding, margin, space between sections, paragraphs, and boxes. It highlights the content and makes your website easy to use.
- Limit the pull-out menus (drop-downs etc.) and avoid using sidebars, sliders, accordions, tabs, and carousels which most of the web designers prefer to include on the website. The user gets distracted and ignores the rest of the useful content on the website

❖ Readability

- ✓ Users should not face difficulty in reading the text on your website so you need to pay attention to the contrast between your text and its background.
- ✓ A lot of people use a background image with text over it where the image doesn't have any overlay and the text gets blended with the image.

❖ Responsiveness

- ✓ People use mobile devices for most of their browsing and 50% of web traffic comes from mobile devices.
- ✓ These all are the reasons that your application should be viewable on different devices.
Text, layout, images all the elements of your website should be viewable and accessible on different devices for better user experience.

❖ Simple Navigation

- ✓ Visitors will definitely move away from your website if they won't be able to find out the things they are looking.
- ✓ Navigation organizes your complete website and guides a user to move around your application.
- ✓ The navigation menu should be on the top and sticky on your website so if a user scrolls down the webpage they can still find and access the menu on your website

❖ Call To Action (CTA)

- ✓ It's very important in your website to have a clear call to action button or form that indicates the next step or action user should take on a page to accomplish the task. For example, buy now, sign up, contact form, subscription, registration form, social media button, etc.
- ✓ It's good if you keep your CTA above the fold so the visitor can find it right in front of them.

❖ Load Speed

- ✓ People are very impatient and they are not going to wait for too long if your website speed is slow.
- ✓ Most of the user moves away from the website within just 5 seconds if they id doesn't load fast.
- ✓ Size of images or videos makes a big impact on website speed so try to compress the image before you add it there.
- ✓ Also, combine code into a central CSS or JavaScript file to reduce the HTTP requests.
- ✓ Minify HTML, CSS, JavaScript (compressed to speed up their load time).

❖ Pick Up The Right Images

- ✓ A picture says a lot about a website and choosing the right image for your website can get a lot of visitors.
- ✓ It is recommended to use high-quality optimized images which doesn't affect your website speed.

❖ Color Palette

- ✓ The color of your website should match the brand of your website.
- ✓ When you are choosing colors for your website it should go well with each other also your text should be clear and readable.

- ✓ Use a single color for the main element (primary), highlights (secondary) and other less-important elements (background). You also need to keep in mind that your primary, secondary, and background colors should be consistent throughout your entire site.
- ✓ Vibrant colors create emotion so it should be used sparingly (e.g. for buttons and call to actions).

Key Points

- + Have a Purpose and Plan
- + Design a clean and simple website. (*less-is-more* approach)
- + Users should not face difficulty in reading the text on your website (Readability)
- + Your application should be viewable on different devices (Responsiveness)
- + Simple Navigation
- + Call To Action (CTA)
- + Load Speed
- + Pick Up The Right Images
- + The color of your website should match the brand of your website.
- + Web design should be compatible with all major browsers.

Symptoms of bad design:

- 1. Confusing or cluttered layout:** Users are unable to find the information they need in your website. Thus, messy layouts can result in a frustrating experience and ultimately lead to users leaving the site or app.
- 2. Poor readability or legibility of text:** Text that is too small, poorly spaced, or hard to read can be a significant obstacle for users. It can make the design feel unprofessional and may even discourage users from reading important content.
- 3. Lack of functionality or user-friendliness:** Lack of functionality can include broken links, slow loading times, or confusing navigation.
- 4. Unattractive or outdated appearance:** A design that looks unattractive or outdated can give users a negative impression of the brand.
- 5. Fails to meet user needs or expectations:** A design that fails to meet user needs or expectations can lead to dissatisfaction and disappointment.
- 6. Horizontal scrollbars, Use of sidebars, sliders, accordions, tabs, and carousels on the website:** The user gets distracted and ignores the rest of the useful content on the website.

Web design issues:

When designing a website, it's crucial to account for several important aspects to ensure a smooth, user-friendly experience. Below are the main web design issues and their explanations, with some examples where applicable.

1. Simplicity

- **Issue:** Overloading a webpage with too many animations, a large amount of information, or other distracting elements can overwhelm users.
- **Example:** A landing page for a blog should not have too many pop-ups, excessive animations, or too much scrolling information. A simple, clear, and minimalist design helps users focus on the content.
- **Solution:** Minimize animations and present concise, well-organized information. Make the design clean and easy to navigate.

2. Identity

- **Issue:** The design of a website should align with the nature of the application or brand.
- **Example:** A corporate website like a law firm should use professional and formal fonts, layouts, and color schemes. In contrast, a gaming website can use bold colors, interactive elements, and dynamic graphics.
- **Solution:** Understand the nature of your website and choose design elements that reinforce the brand's identity.

3. Consistency

- **Issue:** Inconsistent fonts, colors, or layouts across web pages can confuse users and make the site look unprofessional.
- **Example:** A website where one page has a red background and a different font size while another page uses blue with a completely different font can disrupt the user experience.
- **Solution:** Ensure uniformity in design elements such as typography, button styles, and color schemes throughout the website.

4. Robustness

- **Issue:** Missing content such as broken links, missing images, or incomplete functionality reduces the website's effectiveness and credibility.
- **Example:** A news website might have broken links to certain articles, resulting in a 404 "Page Not Found" error.
- **Solution:** Regularly test for broken links and missing content. Tools like W3C Link Checker can help identify and fix broken links.

5. Compatibility

- **Issue:** Websites need to work seamlessly across various browsers and operating systems.
- **Example:** A website that works perfectly in Chrome may look distorted or fail to load correctly in Firefox or Edge.
- **Solution:**
 - **Follow W3C standards:** Ensure your HTML and CSS follow web standards to prevent rendering issues across different browsers.
 - **Validate the code:** Use tools like the W3C Markup Validation Service to ensure your code is error-free.
 - **Test on multiple browsers and devices:** Test your website on various browsers like Chrome, Firefox, Edge, and Safari, as well as on different devices.

1) Browser Compatibility Issues

Websites should render properly across different browsers (Chrome, Firefox, Edge, Safari). Failure to do so can result in distorted layouts, missing functionality, or pages that don't load.

How to Solve Browser Compatibility Issues:

- **Follow W3C standards:** Ensures consistent rendering.
- **Validate the code:** Detect errors that might lead to browser compatibility problems.

- **Know your audience:** If most users use a specific browser, prioritize compatibility for that browser.
- **Test your website:** Regularly check on different browsers and operating systems.

Example:

A CSS property like flexbox works well in modern browsers like Chrome and Firefox but might render differently in older versions of Internet Explorer. Using standardized code and browser-specific fallbacks (such as using @supports in CSS) can ensure cross-browser compatibility.

2) Bandwidth Issue

- **Issue:** Users with slower internet connections (low bandwidth) will experience longer load times if a webpage has many images, videos, or other large files.
 - **Example:** A news site with many high-resolution images or videos will load slowly on a mobile network or slow broadband connection.
 - **Solution:** Optimize images and compress resources. Tools like TinyPNG or image compression via CSS/JS can reduce image sizes. Use lazy loading for images and other large media files to only load them when needed.
-

3) Cache

Cache refers to a browser's temporary storage for web pages and images. When a user visits a website, the browser stores some of the site's data in its cache. On the next visit, the browser tries to load the page from the cache rather than downloading everything again, which makes the page load faster.

- **Issue:** Sometimes changes made to a website are not immediately reflected for users because their browser loads the cached version.
 - **Example:** If a website's logo is changed, but users still see the old logo due to cached data.
 - **Solution:** Set cache headers appropriately or provide users with the option to refresh the cache by adding versioning to resources (e.g., logo-v2.png).
-

4) Display Resolution

Different devices have different display resolutions, and web pages must be designed to scale well across various screen sizes.

- **Issue:** A site designed for a 1280x720 resolution may look too small or too large on other screen resolutions, leading to awkward layout issues.
- **Example:** On a mobile device, a desktop-designed website might require zooming in and out, making navigation difficult.
- **Solution:** Use **responsive design** with CSS media queries to make the website adaptable to different screen sizes. Frameworks like **Bootstrap** help create mobile-friendly, responsive websites that adjust to various screen resolutions.
 - 1366x768 pixels for desktops
 - 375x667 pixels for mobile screens
 - 768x1024 pixels for tablets

Designing Effective navigation

1. Don't Make your Users Guess!

It is sometimes necessary to spoon feed your visitors with information. Make careful decisions as to where your navigation is placed, and make it stand out.

2. Keep it high

Newspaper editors place their most important information — latest headlines, significant content, etc — ‘above the fold’, meaning in the top half of the newspaper itself.

Consider your pages from the same perspective, and keep the significant information, including the navigation, as high up as possible.

3. Below the Banners

Users naturally ignore content placed above any banner ads on your site — they simply consider this space for further advertisement. So don't place any navigation elements above your banner ads.

4. Consistency is Key

All the linked pages of your website locate the navigation in the same place, with the same styles and font. So, users know exactly where to look for it.

5. Don't be Adventurous

It's always good to see your site stand out from the crowd. But when it comes to navigation, try to make it simpler as other websites. So, users won't need to be taught anything new.

6. Add a 'Home' Button

Your home page is the most important page on your site, so make sure your users always know how to get back to it. Allow people who land on sub-pages within your Website to find your home page link from those subpages.

7. Keep it Fast

It is not necessary that all your users have super-fast Internet connections. Despite this fact users want to see fast loading pages wherever possible. So, optimize your images, your HTML and your stylesheets to ensure everything loads as fast as possible.

8. Quality, not Quantity

Build your navigation so that it gives users more and more specific information. Internet users are a lot happier with a few choices of navigation buttons. Use subsections and subcategories with appropriate navigation to enable users to quickly locate the specific content they want.

9. Browser Compatibility

People use other Web browsers and resolutions to the ones you use. Check site's navigation in all possible browsers and resolutions before you launch.

10. Leave Out the Unimportant Stuff

Link such as 'Privacy Policy' should not be the part of Website's main navigation. The most common place for it to go is at the footer section of the page.

Look and Feel of Website:

- ✓ Decoration of each webpage
- ✓ Formatting should be perfect
- ✓ Attractive designing
- ✓ Viewing multimedia

Page Layout and Linking

Page layout is the part of graphic design that deals with the arrangement of visual elements on a page.

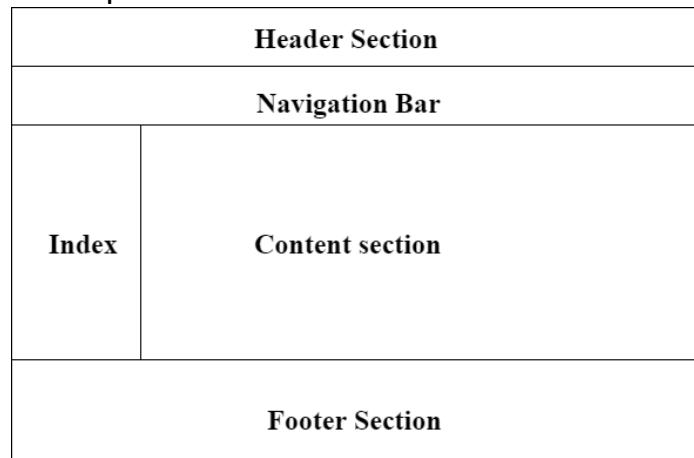
Page layout is used to make the web pages look better. It establishes the overall appearance, relative importance, and relationships between the graphic elements to achieve a smooth flow of information and eye movement for maximum effectiveness or impact.

- **Header:** The part of the front end which is used at the top of the page.
- **Navigation bar:** The navigation bar is the same as the menu list. It is used to display the content information using hyperlinks.
- **Index / Sidebar:** It holds additional information or advertisements and is not always necessary to be added to the page.
- **Content Section:** The content section is the central part where content is displayed.
- **Footer:** The footer section contains the contact information and other query related to web pages. The footer section is always put on the bottom of the web pages

There are two types of page layouts:

1) Flexible page layout

- ✓ Also known as fluid page layout. This kind of page layout works well for text based contents.
- ✓ For ex. Wikipedia is designed with flexible layout.
- ✓ In this layout, majority of the components have percentage widths, thus adjust with screen's resolution.
- ✓ The main drawback is the line of text can be long at wider screen resolution & decrease readability.
- ✓ This makes website responsive to some extent.



2) Fixed page Layout

- ✓ If you want more control over the layout of a page, you may opt to design a web page with a fixed width that stays the same for all users, regardless of monitor resolution or browser window size. This page has consistent width and height.
- ✓ Regardless of user's screen resolution, the designed content remains at fixed position.
- ✓ Fixed layouts are aligned in the center of the browser window to consistently present the same page format at multiple screen resolution.

