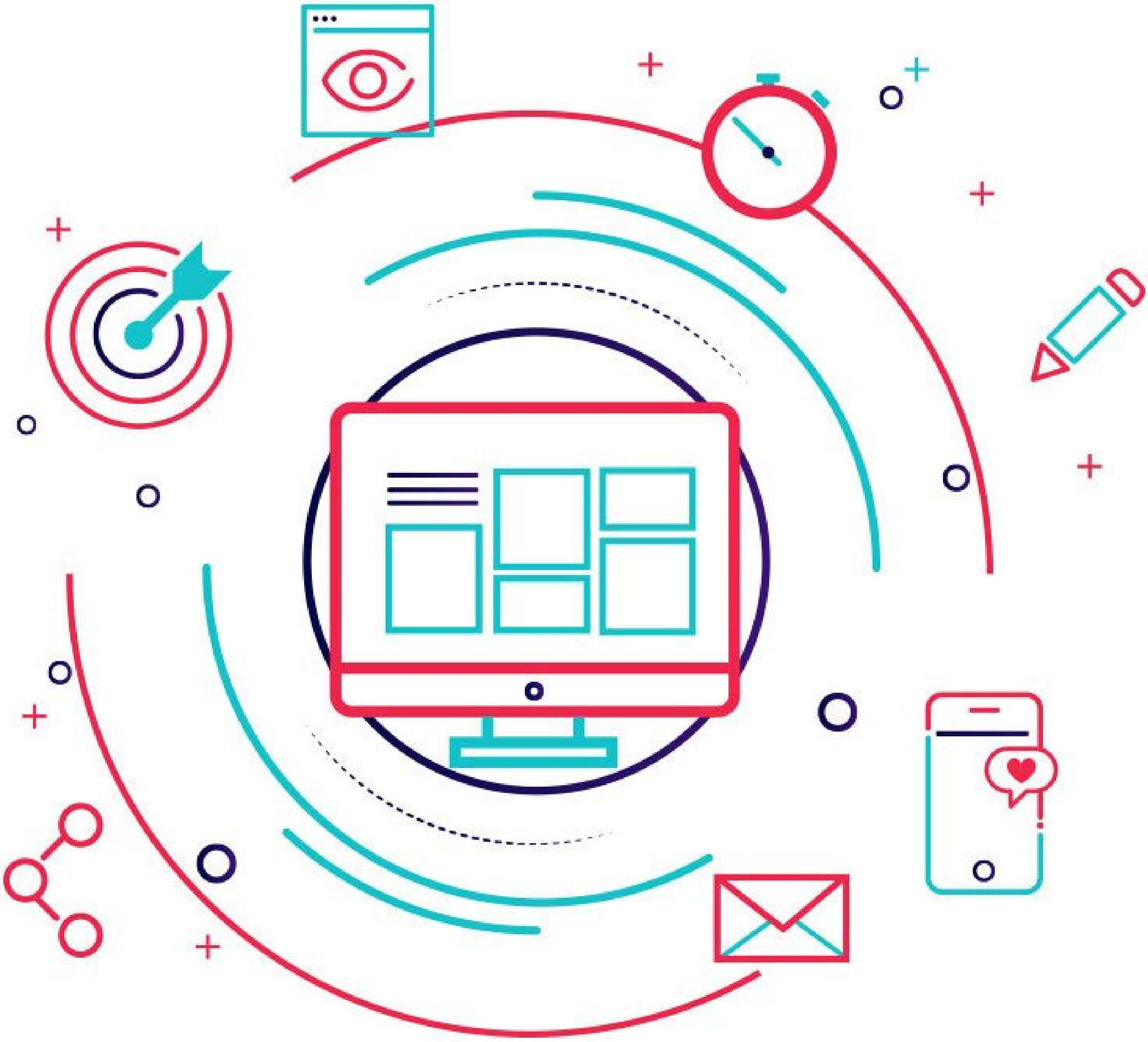
FUNDAMENTALS OF GOOD WEBSITE DESIGN

An effective website design should fulfil its intended function by conveying its particular message whilst simultaneously engaging the visitor. Several factors such as consistency, colours, typography, imagery, simplicity, and functionality contribute to good website design. When designing a website there are many key factors that will contribute to how it is perceived. A well-designed website can help build trust and guide visitors to take action. Creating a great user experience involves making sure your website design is optimised for usability (form and aesthetics) and how easy is it to use (functionality).



1. WEBSITE PURPOSE

Your website needs to accommodate the needs of the user. Having a simple clear intention on all pages will help the user interact with what you have to offer. What is the purpose of your website? Are you imparting practical information like a ‘How to guide’? Is it an entertainment website like sports coverage or are you selling a product to the user? There are many different purposes that websites may have but there are core purposes common to all websites;

1. Describing Expertise

2. Building Your Reputation
3. Generating Leads
4. Sales and After Care

2. SIMPLICITY

Simplicity is the best way to go when considering the user experience and the usability of your website. Below are ways to achieve simplicity through design.

Colour

Colour has the power to communicate messages and evoke emotional responses. Finding a colour palette that fits your brand will allow you to influence your customer's behaviour towards your brand. Keep the colour selection limited to less than 5 colours. Complementary colours work very well. Pleasing colour combinations increase customer engagement and make the user feel good.

Type

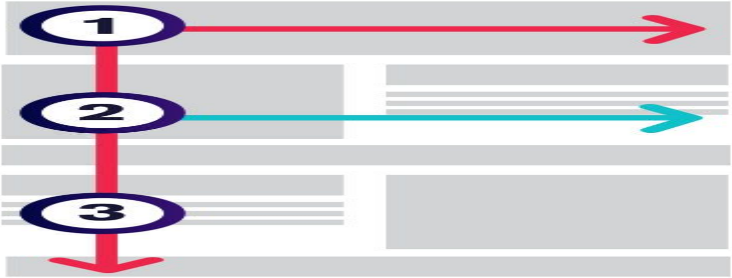
Typography has an important role to play on your website. It commands attention and works as the visual interpretation of the brand voice. Typefaces should be legible and only use a maximum of 3 different fonts on the website.

Imagery

Imagery is every visual aspect used within communications. This includes still photography, illustration, video and all forms of graphics. All imagery should be expressive and capture the spirit of the company and act as the embodiment of their brand personality. Most of the initial information we consume on websites is visual and as a first impression, it is important that high-quality images are used to form an impression of professionalism and credibility in the visitors' minds.

3. NAVIGATION

Navigation is the wayfinding system used on websites where visitors interact and find what they are looking for. Website navigation is key to retaining visitors. If the website navigation is confusing visitors will give up and find what they need elsewhere. Keeping navigation simple, intuitive and consistent on every page is key.



4. F-SHAPED PATTERN READING

The F- based pattern is the most common way visitors scan text on a website. Eye-tracking studies have found that most of what people see is in the top and left areas of the screen. The F shaped layout mimics our natural pattern of reading in the West (left to right and top to bottom). An effectively designed website will work with a reader's natural pattern of scanning the page.

5. VISUAL HIERARCHY

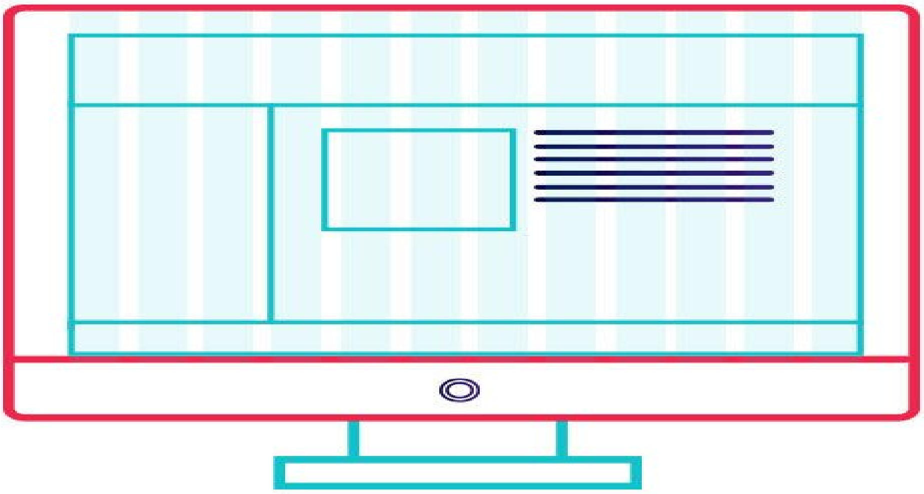
Visual hierarchy is the arrangement of elements in order of importance. This is done either by size, colour, imagery, contrast, typography, whitespace, texture and style. One of the most important functions of visual hierarchy is to establish a focal point; this shows visitors where the most important information is.

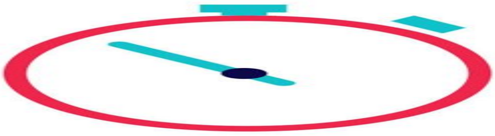
6. CONTENT

An effective website has both great design and great content. Using compelling language great content can attract and influence visitors by converting them into customers.

7. GRID BASED LAYOUT

Grids help to structure your design and keep your content organised. The grid helps to align elements on the page and keep it clean. The grid-based layout arranges content into a clean rigid grid structure with columns, sections that line up and feel balanced and impose order and results in an aesthetically pleasing website.





8. LOAD TIME

Waiting for a website to load will lose visitors. Nearly half of web visitors expect a site to load in 2 seconds or less and they will potentially leave a site that isn't loaded within 3 seconds. Optimising image sizes will help load your site faster.

9. MOBILE FRIENDLY

More people are using their phones or other devices to browse the web. It is important to consider building your website with a responsive layout where your website can adjust to different screens.

Introduction to Web Page & Website, Web Application

1) Web Page

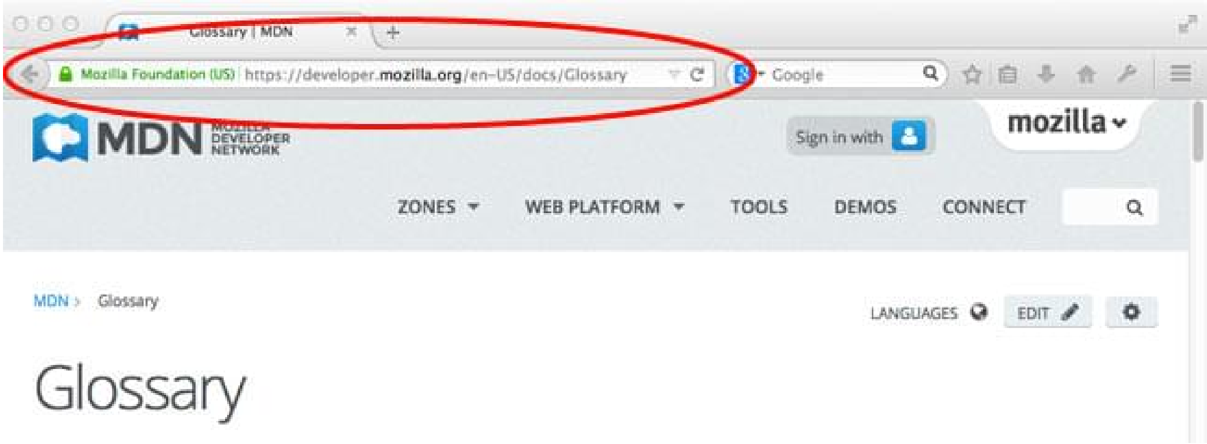
A document which can be displayed in a web browser such as Firefox, Google Chrome, Opera, Microsoft Edge, or Apple Safari. These are also often called just "pages."

A **web page** is a simple document displayable by a browser. Such documents are written in the HTML language. A web page can embed a variety of different types of resources such as:

- *style information* — controlling a page's look-and-feel
- *scripts* — which add interactivity to the page
- *media* — images, sounds, and videos.

Note: Browsers can also display other documents such as PDF files or images, but the term **web page** specifically refers to HTML documents. Otherwise, we only use the term **document**.

All web pages available on the web are reachable through a unique address. To access a page, just type its address in your browser address bar:



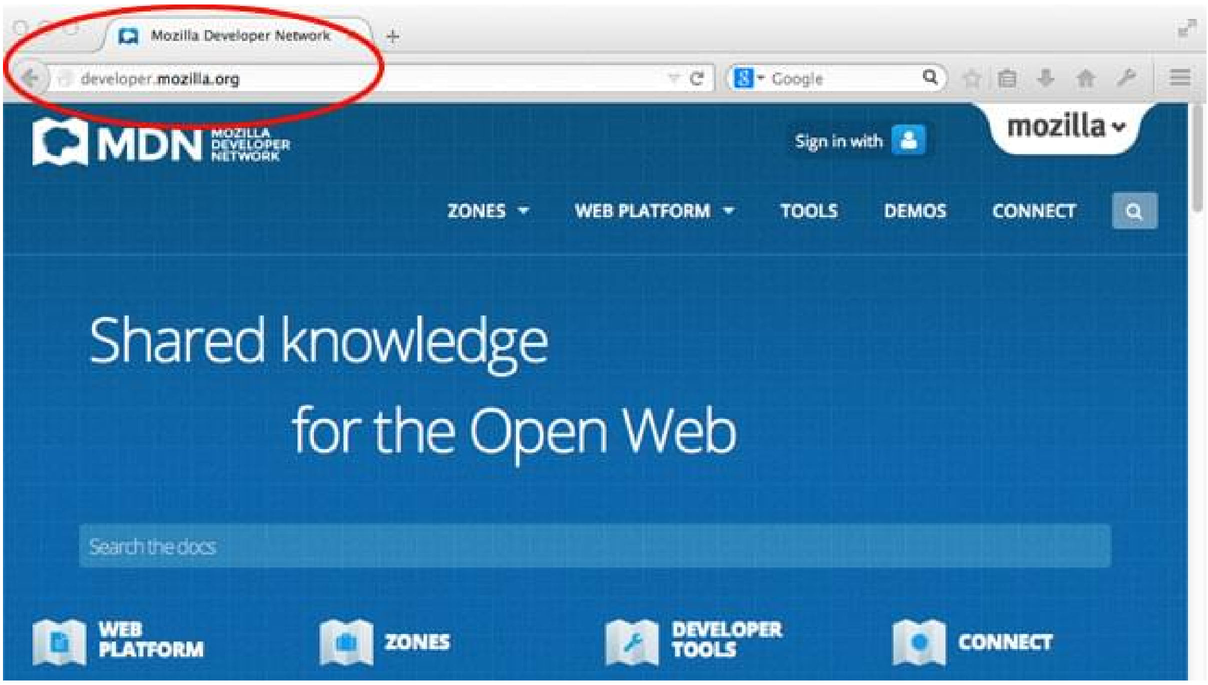
Mozilla Foundation (US) https://developer.mozilla.org/en-US/docs/Glossary

2) Website

A collection of web pages which are grouped together and usually connected together in various ways. Often called a "website" or a "site."

A *website* is a collection of linked web pages (plus their associated resources) that share a unique domain name. Each web page of a given website provides explicit links—most of the time in the form of clickable portions of text—that allow the user to move from one page of the website to another.

To access a website, type its domain name in your browser address bar, and the browser will display the website's main web page, or *homepage* (casually referred as "the home"):



Note that it is also possible to have a *single-page website*: a site that consists of a single web page which is dynamically updated with new content when needed.

3) Web Server

A computer that hosts a website on the Internet.

A *web server* is a computer hosting one or more *websites*. "Hosting" means that all the *web pages* and their supporting files are available on that computer. The *web server* will send any *w eb page* from the *website* it is hosting to any user's browser, per user request.

Don't confuse *websites* and *web servers*. For example, if you hear someone say, "My website is not responding", it actually means that the *web server* is not responding and therefore the *website* is not available. More importantly, since a web server can host multiple websites, the term *web server* is never used to designate a website, as it could cause great confusion. In our previous example, if we said, "My web server is not responding", it means that multiple websites on that web server are not available.

4) Search engine

A web service that helps you find other web pages, such as Google, Bing, Yahoo, or DuckDuckGo. Search engines are normally accessed through a web browser (e.g. you can perform search engine searches directly in the address bar of Firefox, Chrome, etc.) or through a web page (e.g. bing.com or duckduckgo.com).

Example: a public library. This is what you would generally do when visiting a library:

1. Find a search index and look for the title of the book you want.
2. Make a note of the catalog number of the book.

3. Go to the particular section containing the book, find the right catalog number, and get the book.

Compare the library with a web server:

- The library is like a web server. It has several sections, which is similar to a web server hosting multiple websites.
- The different sections (science, math, history, etc.) in the library are like websites. Each section is like a unique website (two sections do not contain the same books).
- The books in each section are like webpages. One website may have several webpages, e.g., the Science section (the website) will have books on heat, sound, thermodynamics, statics, etc. (the webpages). Webpages can each be found at a unique location (URL).
- The search index is like the search engine. Each book has its own unique location in the library (two books cannot be kept at the same place) which is specified by the catalog number.
- Search engines are a common source of confusion on the web. A search engine is a special kind of website that helps users find web pages from *other* websites.
- There are plenty out there: Google, Bing, Yandex, DuckDuckGo, and many more. Some are generic, some are specialized about certain topics. Use whichever you prefer.
- Many beginners on the web confuse search engines and browsers. Let's make it clear: A ***browser*** is a piece of software that retrieves and displays web pages; a ***search engine*** is a website that helps people find web pages from other websites. The confusion arises because, the first time someone launches a browser, the browser displays a search engine's homepage. This makes sense, because, obviously, the first thing you want to do with a browser is to find a web page to display. Don't confuse the infrastructure (e.g., the browser) with the service (e.g., the search engine). The distinction will help you quite a bit, but even some professionals speak loosely, so don't feel anxious about it.

5) Web Application:

A web application (web app) is an application program that is stored on a remote server and delivered over the internet through a browser interface. Web services are web apps by definition and many, although not all, websites contain web apps.

Developers design web applications for a wide variety of uses and users, from an organization to an individual for numerous reasons. Commonly used web applications can include webmail, online calculators or e-commerce shops. While users can only access some web apps by a specific browser, most are available no matter the browser.

How web applications work

Web applications do not need to be downloaded since they are accessed through a network. Users can access a web application through a web browser, such as Google Chrome, Mozilla Firefox or Safari.

For a web app to operate, it needs a web server, application server and database. Web servers manage the requests that come from a client, while the application server completes the requested task. A database stores any necessary information.

Web applications typically have short development cycles and small development teams. Developers write most web apps in JavaScript, HTML5 or CSS. Client-side programming typically utilizes these languages, which help build an application's front-end. Server-side programming creates the scripts a web app will use. Languages such as Python, Java and Ruby are commonly used in server-side programming.

Benefits

Web applications have many benefits. Some common benefits include the following:

- Multiple users can access the same version of an application.
- Users don't need to install the app.
- Users can access the app through various platforms such as a desktop, laptop or mobile.
- Users can access the app through multiple browsers.

Web app vs. native app vs. hybrid app

Within the mobile computing sector, web apps are sometimes contrasted with native apps, which are applications developers build specifically for a particular platform or device and install on that device. Native apps can commonly make use of device-specific hardware, such as a GPS or camera on a mobile native app.

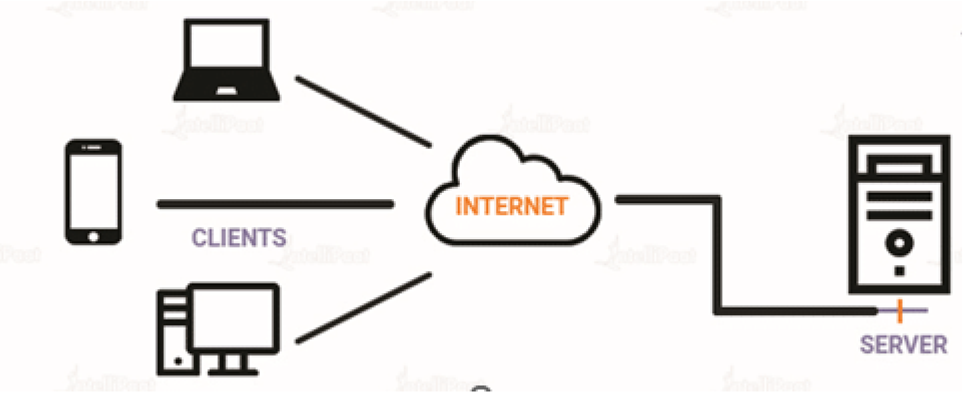
Programs that combine the two approaches are sometimes referred to as hybrid applications. Hybrid apps work similar to web apps but install to the device as a native app would. Hybrid apps can also take advantage of device-specific resources by using internal APIs. Downloaded native apps can sometimes operate offline; however, hybrid apps don't have this functionality.

A hybrid app will typically share similar navigation elements to a web app since they are primarily based on web apps.

Client Server Architecture

Two factors are involved :

- A server is the one who provides requested services.
- Clients are the ones who request services.



Client-server architecture is a computing model in which the server hosts, delivers, and manages most of the resources and services requested by the client. It is also known as the networking computing model or client-server network as all requests and services are delivered over a network. The client-server architecture or model has other systems connected over a network where resources are shared among the different computers.

Typically, client-server architecture is arranged in a way that clients are often situated at workstations or on personal computers, while servers are located elsewhere on the network, usually on more powerful machines. Such a model is especially beneficial when the clients and server perform routine tasks. For example, in hospital data processing, a client computer can be busy running an application program for entering patient information, meanwhile, the server computer can be running another program to fetch and manage the database in which the information is permanently stored.

Client-Server architecture example

Here are some of the client-server model architecture examples from our daily life. Hope it helps you to understand the concept better.

Mail servers

Email servers are used for sending and receiving emails. There are different software that allows email handling.

File servers

File servers act as a centralized location for files. One of the daily life examples to understand this is the files that we store in Google Docs. The cloud services for Microsoft Office and Google Docs can be accessed from your devices; the files that you save from your computer can be accessed from your phone. So, the centrally stored files can be accessed by multiple users.

Web servers

Web servers are high-performance computers that host different websites. The server site data is requested by the client through high-speed internet.

Components of Client-Server architecture:

Essentially, three components are required to make client-server architecture work. The three components are workstations, servers, and networking devices. Let us, now, discuss them in detail:

Workstations

Workstations are also called client computers. Workstations work as subordinates to servers and send them requests to access shared files and databases. A server requests information from the workstation and performs several functions as a central repository of files, programs, databases, and management policies. Workstations are governed by server-defined policies.

Servers

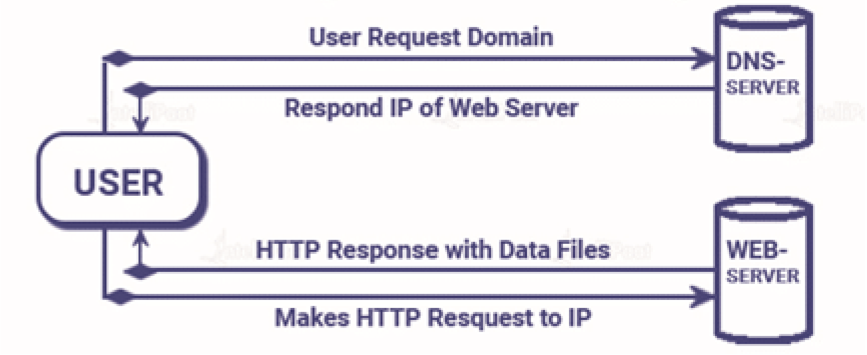
Servers are defined as fast-processing devices that act as centralized repositories of network files, programs, databases, and policies. Servers have huge storage space and robust memory to deal with multiple requests, approaching simultaneously from various workstations. Servers can perform many roles, such as mail server, database server, file server, and domain controller, in client-server architecture at the same time.

Networking devices

Now that we know about the roles that workstations and servers play, let us learn about what connects them, networking devices. Networking devices are a medium that connects workstations and servers in a client-server architecture. Many networking devices are used to perform various operations across the network. For example, a hub is used for connecting a server to various workstations. Repeaters are used to effectively transfer data between two devices. Bridges are used to isolate network segmentation.

How does client-server architecture work?

The client-server architecture is made up of two elements, one that provides services and the other that consumes those services.

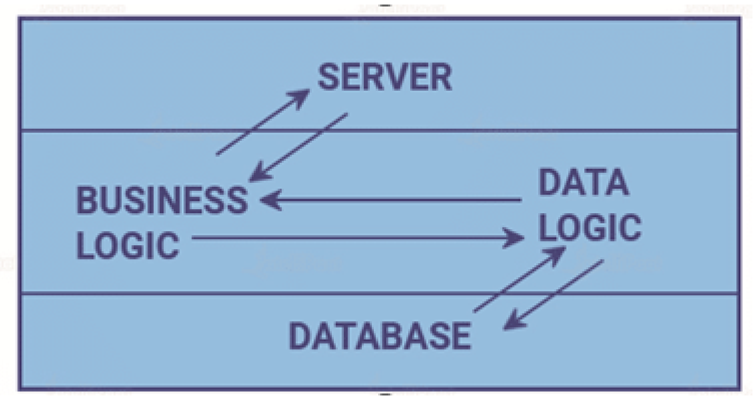


- The user enters the uniform resource locator (URL) of the website or file and the browser sends a request to the domain name system (DNS) server.
- DNS server is responsible for searching and retrieving the IP address associated with a web server and then initiating actions using that IP address.
- After the DNS server responds, the browser sends over an HTTP or HTTPS request to the web server's IP, which was provided by the DNS server.
- Following the request, the server proceeds to transmit the essential website files required.

- Ultimately, the files are processed by the browser and the website is subsequently presented for viewing.

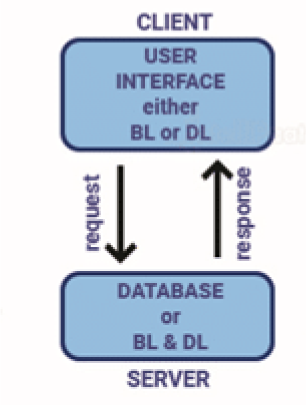
The functionality of Client-Server architecture is in various tiers.

1-tier architecture



The architecture in this specific client-server category incorporates various settings, including configuration settings and marketing logic, within a single device. Although the wide range of services provided by the 1-tier architecture establishes it as a dependable resource, managing such an architecture proves challenging. This difficulty primarily arises from the variability of data, often leading to duplicated efforts. The 1-tier architecture comprises multiple layers, such as the presentation layer, business layer, and data layer, which are unified through a specialized software package. The data residing within this layer is typically stored either in local systems or on a shared drive.

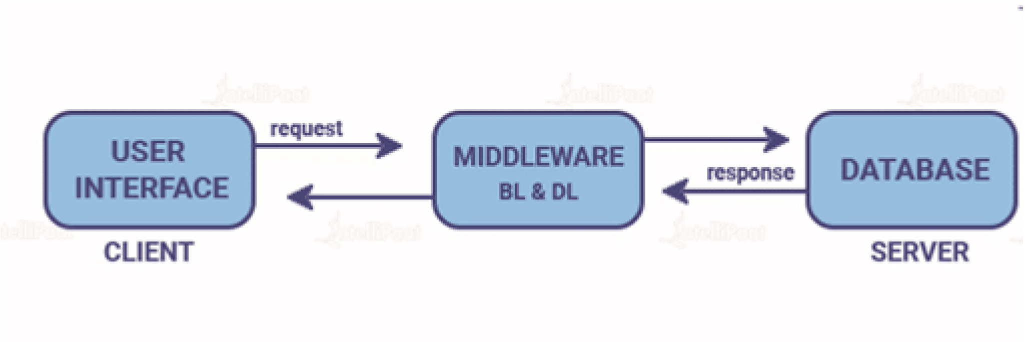
2-tier architecture



The best environment is possessed by this architecture, where the client's side stores the user interface and the server houses the database, while either the client's side or the server's side manages the database logic and business logic.

The 2-tier architecture outpaces the 1-tier architecture due to its absence of intermediaries between the client and server. Its primary application is to eliminate client confusion, and an instance of its popularity lies in the online ticket reservation system.

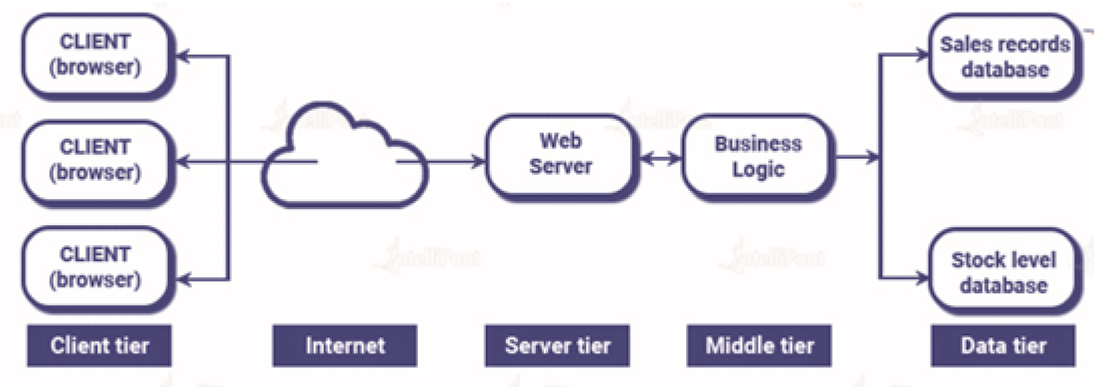
3-tier architecture



Unlike 2-tier architecture that has no intermediary, in 3-tier client-server architecture, middleware lies between the client and the server. If the client places a request to fetch specific information from the server, the request will first be received by the middleware. It will then be dispatched to the server for further action. The same pattern will be followed when the server sends a response to the client. The framework of 3-tier architecture is categorized into three main layers, presentation layer, application layer, and database tier.

All three layers are controlled at different ends. While the presentation layer is controlled at the client's device, the middleware and the server handle the application layer and the database tier respectively. Due to the presence of a third layer that provides data control, 3-tier architecture is more secure, has invisible database structure, and provides data integrity.

N-tier architecture



N-tier architecture is also called multi-tier architecture. It is the scaled form of the other three types of architecture. This architecture has a provision for locating each function as an isolated layer that includes presentation, application processing, and management of data functionalities.

Difference between peer-to-peer network and Client-Server architecture

The following table lists five major differences between peer-to-peer architecture and client-server architecture:

Client server architecture	**Peer-to-peer architecture**
It has specific clients and servers.	There is no differentiation between clients and servers.
Centralized data management is accomplished through it.	It possesses its own data and applications.
The objective is to disseminate knowledge or exchange relevant data.	Its main goal is to encourage peer connectivity and preserve continuous relationships between people.
Data is provided only in response to a request.	In this network, peers have the authority to request as well as provide a service.

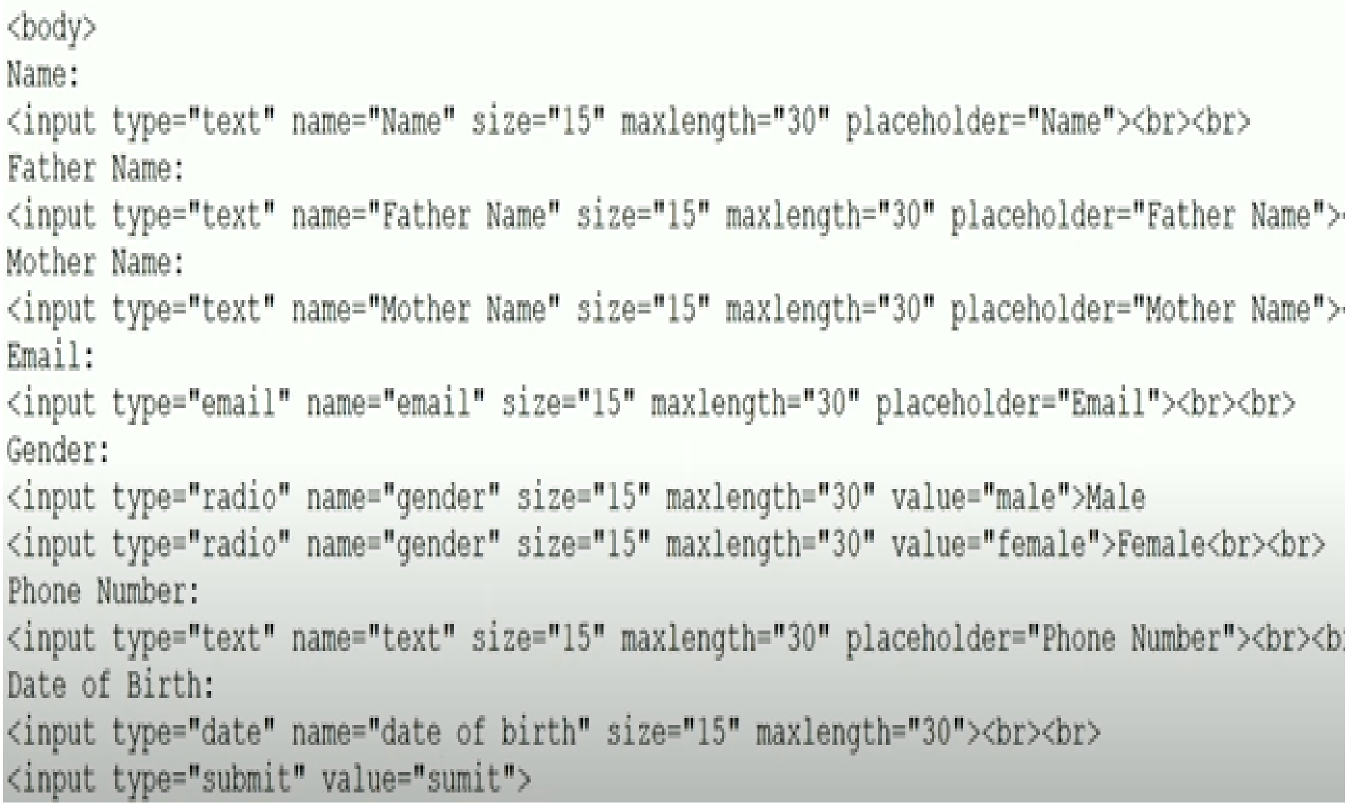
It is suitable for small as well as large networks.	It is appropriate for a limited number of users, specifically fewer than ten devices.

Advantages and disadvantages of client-server architecture

The advantages and disadvantages of client-server architecture are mentioned below:

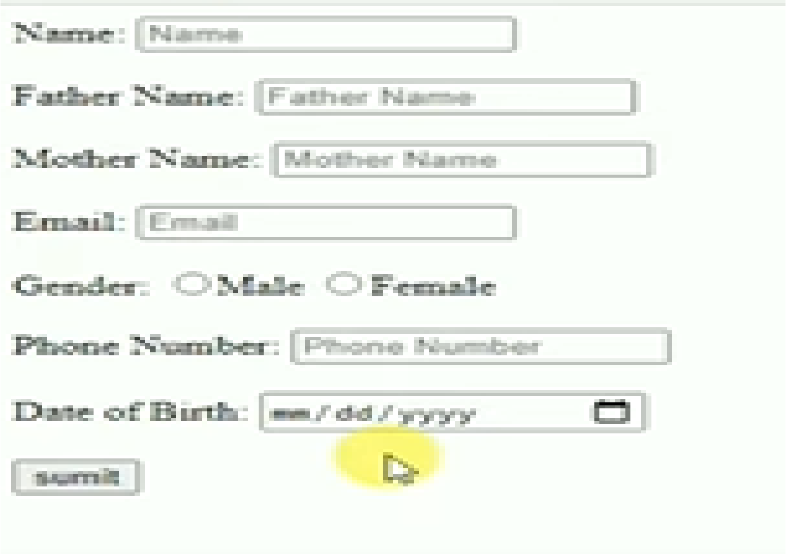
Advantages	Disadvantages
The centralized network has complete leverage to control the processes and activities.	If the primary server goes down, the entire architecture is disrupted.
All devices in the network can be controlled centrally.	It is expensive to operate because of the cost of heavy hardware and software tools.
Users have the authority to access any file, residing in the central storage, at any time.	This architecture requires particular OSs related to networking.
It provides a good user interface, easy file finding procedure, and management system for organizing files.	Too many users at once can cause the problem of traffic congestion.
Easy sharing of resources across various platforms is possible.	It requires highly technical stuff, such as server machines, for maintenance of the network.

Basic Form in HTML



```
<body>
Name:
<input type="text" name="Name" size="15" maxlength="30" placeholder="Name"><br><br>
Father Name:
<input type="text" name="Father Name" size="15" maxlength="30" placeholder="Father Name">
Mother Name:
<input type="text" name="Mother Name" size="15" maxlength="30" placeholder="Mother Name">
Email:
<input type="email" name="email" size="15" maxlength="30" placeholder="Email"><br><br>
Gender:
<input type="radio" name="gender" size="15" maxlength="30" value="male">Male
<input type="radio" name="gender" size="15" maxlength="30" value="female">Female<br><br>
Phone Number:
<input type="text" name="text" size="15" maxlength="30" placeholder="Phone Number"><br><b
Date of Birth:
<input type="date" name="date of birth" size="15" maxlength="30"><br><br>
<input type="submit" value="sumit">
```

Output:



Name: Name

Father Name: Father Name

Mother Name: Mother Name

Email: Email

Gender: ○ Male ○ Female

Phone Number: Phone Number

Date of Birth: mm / dd / yyyy