

Introduction to Web Technologies:

Internet and WWW

Introduction to Internet:- A global computer network providing a variety of information and communication facilities, consisting of interconnected networks using standardized communication protocols. "the guide is also available on the Internet"

The Internet is the global system of interconnected computer networks that use the Internet protocol suite (TCP/IP) to link devices worldwide. It is a network of networks that consists of private, public, academic, business, and government networks of local to global scope, linked by a broad array of electronic, wireless, and optical networking technologies. The Internet carries a vast range of information resources and services.

History of Internet

This marvelous tool has quite a history that holds its roots in the cold war scenario. A need was realized to connect the top universities of the United States so that they can share all the research data without having too much of a time lag. This attempt was a result of Advanced Research Projects Agency (ARPA) which was formed at the end of 1950s just after the Russians had climbed the space era with the launch of Sputnik. After the ARPA got success in 1969, it didn't take the experts long to understand that how much potential can this interconnection tool have. In 1971 Ray Tomlinson made a system to send electronic mail. This was a big step in the making as this opened gateways for remote computer accessing i.e. telnet.

During all this time, rigorous paper work was being done in all the elite research institutions. From giving every computer an address to setting out the rules, everything was getting penned down. 1973 saw the preparations for the vital TCP/IP and Ethernet services. At the end of 1970s, Usenet groups had surfaced up. By the time the 80s had started, IBM came up with its PC based on Intel 8088 processor which was widely used by students and universities for it solved the purpose of easy computing. By 1982, the Defense Agencies made the TCP/IP compulsory and the term —internetll was coined. The domain name services arrived in the year 1984 which is also the time around which various internet based marked their debut. A worm, or a rust the computers, attacked in 1988 and disabled over 10% of the computer systems all over the world. While most of the researchers regarded it as an opportunity to enhance computing as it was still in its juvenile phase, quite a number of computer companies became interested in dissecting the cores of the malware which resulted to the formation Computer Emergency Rescue Team (CERT). Soon after the world got over with the computer worm, World Wide Web came into existence. Discovered by Tim Berners-Lee, World Wide Web was seen as a service to connect documents in websites using hyperlinks.

Web Development Using PHP and MySQL-MMCB311C

World Wide Web

The World Wide Web (abbreviated WWW or the Web) is an information space where documents and other web resources are identified by Uniform Resource Locators (URLs), interlinked by hypertext links, and can be accessed via the Internet. English scientist Tim Berners-Lee invented the World Wide Web in 1989. He wrote the first web browser computer program in 1990 while employed at CERN in Switzerland. The Web browser was released outside CERN in 1991, first to other research institutions starting in January 1991 and to the general public on the Internet in August 1991.

The World Wide Web has been central to the development of the Information Age and is the primary tool billions of people use to interact on the Internet. Web pages are primarily text documents formatted and annotated with Hypertext Markup Language (HTML). In addition to formatted text, web pages may contain images, video, audio, and software components that are rendered in the user's web browser as coherent pages of multimedia content.

Embedded hyperlinks permit users to navigate between web pages. Multiple web pages with a common theme, a common domain name, or both, make up a website. Website content can largely be provided by the publisher, or interactively where users contribute content or the content depends upon the users or their actions. Websites may be mostly informative, primarily for entertainment, or largely for commercial, governmental, or non-governmental organizational purposes



WWW is another example of client/server computing. Each time a link is followed, the client is requesting a document (or graphic or sound file) from a server (also called a Web server) that's part of the World Wide Web that "serves" up the document. The server uses a protocol called HTTP or Hyper Text Transport Protocol. The standard for creating hypertext documents for the WWW is Hyper Text Markup Language or HTML. HTML essentially codes plain text documents so they can be viewed on the Web.

Browsers:

WWW Clients, or "Browser": The program you use to access the WWW is known as a browser because it "browses" the WWW and requests these hypertext documents. Browsers can

Web Development Using PHP and MySQL-MMCB311C

be graphical, allows to see and hear the graphics and audio;

text-only browsers (i.e., those with no sound or graphics capability) are also available. All of these programs understand http and other Internet protocols such as FTP, gopher, mail, and news, making the WWW a kind of "one stop shopping" for Internet users.

Year	List of Web browsers
1991	World Wide Web (Nexus)
1992	Viola WWW, Erwise, MidasWWW, MacWWW (Samba)
1993	Mosaic, Cello,[2] Lynx 2.0, Arena, AMosaic 1.0
1994	IBM WebExplorer, Netscape Navigator, SlipKnot 1.0, MacWeb, IBrowse, Agora (Argo), Minuet
1995	Internet Explorer 1, Internet Explorer 2, Netscape Navigator 2.0, OmniWeb, UdiWWW, Grail
1996	Arachne 1.0, Internet Explorer 3.0, Netscape Navigator 3.0, Opera 2.0, PowerBrowser 1.5,[4] Cyberdog, Amaya 0.9,[5] AWeb, Voyager
1997	Internet Explorer 4.0, Netscape Navigator 4.0, Netscape Communicator 4.0, Opera 3.0,[6] Amaya 1.0[5]
1998	iCab, Mozilla
1999	Amaya 2.0,[5] Mozilla M3, Internet Explorer 5.0
2000	Konqueror, Netscape 6, Opera 4,[7] Opera 5,[8] K-Meleon 0.2, Amaya 3.0,[5] Amaya 4.0[5]
2001	Internet Explorer 6, Galeon 1.0, Opera 6,[9] Amaya 5.0[5]
2002	Netscape 7, Mozilla 1.0, Phoenix 0.1, Links 2.0, Amaya 6.0,[5] Amaya 7.0[5]
2003	Opera 7,[10] Apple Safari 1.0, Epiphany 1.0, Amaya 8.0[5]
2004	Firefox 1.0, Netscape Browser, OmniWeb 5.0
2005	Opera 8,[11] Apple Safari 2.0, Netscape Browser 8.0, Epiphany 1.8, Amaya 9.0,[5] AOL Explorer 1.0, Maxthon 1.0, Shiira 1.0
2006	Mozilla Firefox 2.0, Internet Explorer 7, Opera 9,[12], SeaMonkey 1.0, K-Meleon 1.0, Galeon 2.0, Camino 1.0, Avant11, iCab 3
2007	Apple Safari 3.0, Maxthon 2.0, Netscape Navigator 9, NetSurf 1.0, Flock 1.0, Conkeror
2008	Google Chrome 1, Mozilla Firefox 3, Opera 9.5,[13], Apple Safari 3.1, Konqueror 4, Amaya 10.0,[5] Flock 2, Amaya 11.0[5]
2009	Google Chrome 2–3, Mozilla Firefox 3.5, Internet Explorer 8, Opera 10,[14], Apple Safari 4, SeaMonkey 2, Camino 2,surf, Pale Moon 3.0[15]
2010	Google Chrome 4–8, Mozilla Firefox 3.6, Opera 10.50,[16], Opera 11, Apple Safari 5, K-Meleon 1.5.4,
2011	Google Chrome 9–16, Mozilla Firefox 4-9, Internet Explorer 9, Opera 11.50,Apple Safari 5.1, Maxthon 3.0, SeaMonkey 2.1–2.6

Web Development Using PHP and MySQL-MMCB311C

2012	Google Chrome 17–23, Mozilla Firefox 10–17, Internet Explorer 10, Opera 12, Apple Safari 6, Maxthon 4.0, SeaMonkey 2.7-2.14
2013	Google Chrome 24–31, Mozilla Firefox 18–26, Internet Explorer 11, Opera 15–18, Apple Safari 7, SeaMonkey 2.15-2.23
2014	Google Chrome 32–39, Mozilla Firefox 27–34, Opera 19–26, Apple Safari 8
2015	Google Chrome 40–47, Microsoft Edge, Mozilla Firefox 35–43, Opera 27–34, Vivaldi

Uniform Resource Locators, or URLs: A Uniform Resource Locator, or URL is the address of a document found on the WWW. Browser interprets the information in the URL in order to connect to the proper Internet server and to retrieve your desired document. Each time a click on a hyperlink in a WWW document instructs browser to find the URL that's embedded within the hyperlink.

The elements in a URL: **Protocol://server's address/filename**

Hypertext protocol: <http://www.aucegypt.edu>

File Transfer Protocol: <ftp://ftp.dartmouth.edu>

Telnet Protocol: <telnet://pac.carl.org>

News Protocol: <news:alt.rock-n-roll.stones>

What are Domains? Domains divide World Wide Web sites into categories based on the nature of their owner, and they form part of a site's address, or uniform resource locator (URL). Common top-level domains are:

.com—commercial enterprises	.mil—military site
org—organization site (non-profits, etc.)	int—organizations established by international treaty
.net—network	.biz—commercial and personal
.edu—educational site (universities, schools, etc.)	.info—commercial and personal
.gov—government organizations	.name—personal sites

Additional three- letter, four- letter, and longer top- level domains are frequently added. Each country linked to the Web has a two- letter top- level domain, for example .fr is France, .ie is Ireland.

MIME (Multi-Purpose Internet Mail Extensions): MIME is an extension of the original Internet e- mail protocol that lets people use the protocol to exchange different kinds of data files on the Internet: audio, video, images, application programs, and other kind s, as well as the ASCII text handled in the original protocol, the Simple Mail Transport Protocol (SMTP). In

1991, Nathan Borenstein of Bellcore proposed to the IETF that SMTP be extended so that Internet (but mainly Web) clients and servers could recognize and handle other kinds of data than ASCII text. As a result, new file types were added to "mail" as a supported Internet Protocol file type.

Servers insert the MIME header at the beginning of any Web transmission. Clients use this header to select an appropriate "player" application for the type of data the header indicates. Some of these players are built into the Web client or browser (for example, all browsers come with GIF and JPEG image players as well as the ability to handle HTML files); other players may need to be downloaded.

New MIME data types are registered with the Internet Assigned Numbers Authority (IANA).

MIME is specified in detail in Internet Request for Comments 1521 and 1522, which amend the original mail protocol specification, RFC 821 (the Simple Mail Transport Protocol) and the ASCII messaging header, RFC 822.

Hypertext Transport Protocol:

HTTP means HyperText Transfer Protocol. HTTP is the underlying protocol used by the World Wide Web and this protocol defines how messages are formatted and transmitted, and what actions Web servers and browsers should take in response to various commands.

For example, when you enter a URL in your browser, this actually sends an HTTP command to the Web server directing it to fetch and transmit the requested Web page. The other main standard that controls how the World Wide Web works is HTML, which covers how Web pages are formatted and displayed.

HTTP is called a stateless protocol because each command is executed independently, without any knowledge of the commands that came before it. This is the main reason that it is difficult to implement Web sites that react intelligently to user input.

HTTPS: A similar abbreviation, HTTPS means Hyper Text Transfer Protocol Secure. Basically, it is the secure version of HTTP. Communications between the browser and website are encrypted by Transport Layer Security (TLS), or its predecessor, Secure Sockets Layer (SSL).

The Web Programmer's Toolbox:

- **HTML** - a *markup* language
 - To describe the general form and layout of documents
 - HTML is **not** a programming language - it cannot be used describe **computations**.
 - An HTML document is a mix of **content** and **controls**

- Controls are **tags** and their **attributes**
 - Tags often delimit content and specify something about how the content should be arranged in the document
For example, <p>Write a paragraph here </p> is an *element*.
 - Attributes provide additional information about the content of a tag
For example,
- Plug ins
 - Integrated into tools like word processors, effectively converting them to WYSIWYG HTML editors
- Filters
 - Convert documents in other formats to HTML

Basics of JavaScript and client-side scripting

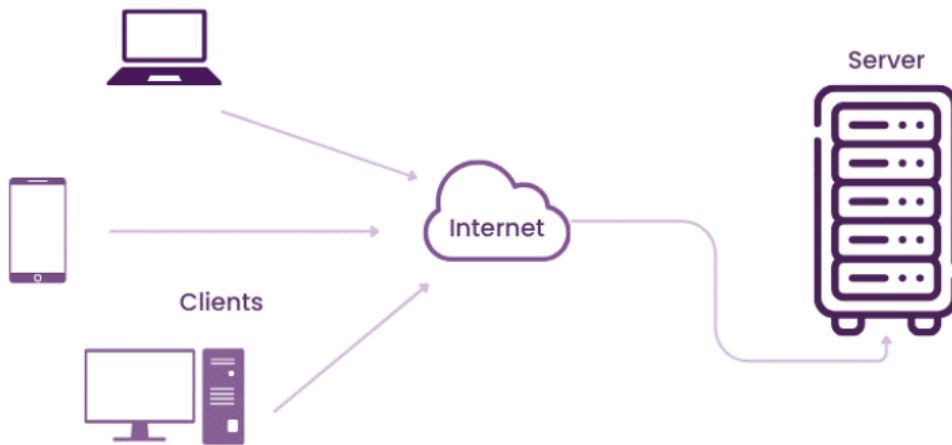
HTTP

Web Servers,

Client-Server Architecture

A Client Server Architecture is a network-based computing structure where responsibilities and operations get distributed between clients and servers. Client-Server Architecture is widely used for network applications such as email, web, online banking, e-commerce, etc. In this blog, we will explore Client-Server Architecture, components, types with examples, and how it works.

What is Client Server Architecture?



Client-Server Architecture is a network model that allows communication and data exchange between different applications over a single or multiple servers.

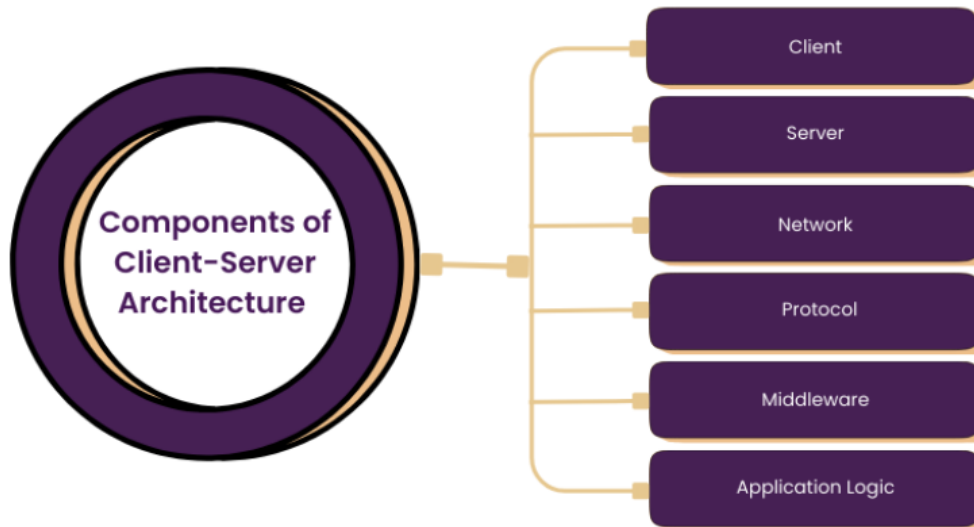
This model divides the system into two parts: the client side and the server side.

a) Client: An application that requests services from the server, such as data retrieval, storage, calculations, or other functions.

b) Server: An application that processes client requests, sends responses, or performs specified actions. The server and client may reside on the same machine or different devices across the network. Effective Communication occurs via predefined protocols like HTTP, FTP, or SMTP.

Client-Server Architecture is widely used in applications such as email, web browsing, online banking, and e-commerce.

Components of Client-Server Architecture



Client-Server Architecture depends on three main components that need to work together for it to function. These components are:

- a) Client:** A client device or software that requests services from a server. Clients are consumer-facing and often include web browsers, mobile applications, or desktop applications that people can interact with. They communicate with the server to retrieve data, make transactions, or perform other tasks by delegating that responsibility to the server.
- b) Server:** A server is a computer or program that offers services or solutions to clients over a network. Servers handle processing of client requests, which includes tasks like file storage, database access, and application hosting, along with backend activities like computations, data management, and business logic, significantly reducing what clients need to handle.
- c) Network:** This serves as the channel through which clients and servers are connected for data transfer between them. Networks range from local area networks (LAN) within a single building to wide area networks (WAN) and the internet, which can span countries. It acts as the intermediary, facilitating the interchange of requests and responses between the clients and servers, which influences the speed and reliability of these interactions.
- d) Protocol:** Protocols are rules that define how data is exchanged between clients and servers, ensuring communication is orderly, secure, and understandable. Common protocols include HTTP or HTTPS for web

services, FTP for file transfers, and SMTP for email. They help bridge communication between different systems, independent of their technology stack.

e) Middleware: Middleware acts as a bridge between client-side and server-side code, enabling them to communicate. It performs tasks such as authentication, load balancing, data translation, and message queuing, simplifying interactions within the Client-Server model by enhancing transaction speed, scalability, and integration.

f) Application Logic: Application logic is the code and processes that determine how a server responds to client requests, involving business rules, Big Data Processing, and workflows on the server side. It ensures the server correctly interprets client requests, performs necessary calculations or data manipulations, and delivers appropriate responses.



Characteristics of Client-Server Architecture

Client-Server architecture has distinct characteristics:

- a) The client and server machines can differ greatly in hardware and software requirements and may come from different vendors.
- b) The network can scale horizontally by adding more client machines or vertically by upgrading to more powerful servers or a multi-server configuration.
- c) A single server can offer multiple services simultaneously, though each service must run its own server program.
- d) Both client and server applications interact directly with a transport layer protocol, establishing communication to send and receive information.
- e) Both client and server computers require a full stack of protocols, where the transport protocol utilises lower-layer protocols to send and receive individual messages.

How Does Client-Server Architecture Work?

The basic steps of how Client-Server Architecture works are:

- 1) In the first step, the client sends a request to the server using the network medium. The request can be a query, a command, or a message.

- 2) In the second step, the server receives the request and processes it according to its logic and data. The server may access its own resources or other servers to fulfil the request.
- 3) In the third step, the server sends a response back to the client using the network medium. The response can be Data, an acknowledgement, or an error message.
- 4) Lastly, the client receives the response and displays it to the user or performs further actions based on it.

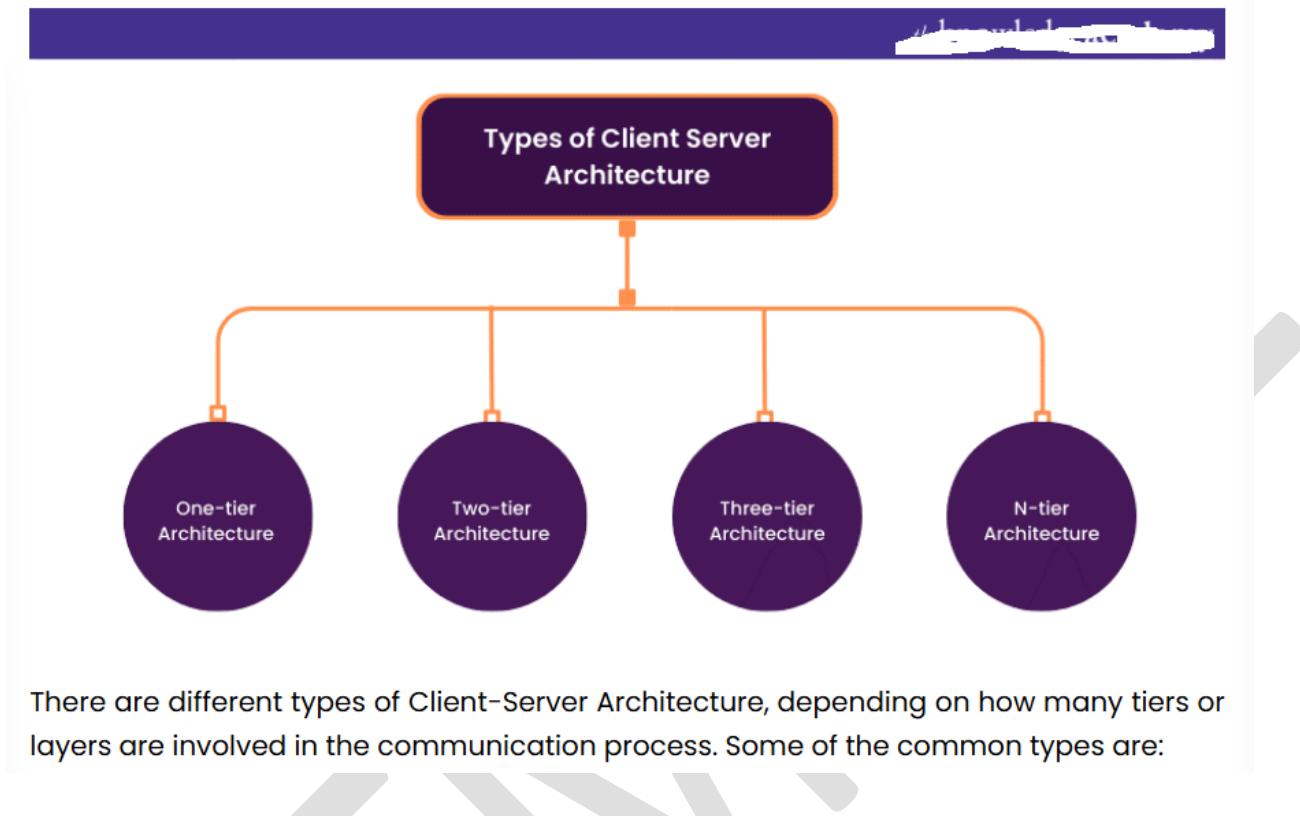
What are Some Examples of Client-Server Architecture?

You might be surprised at how often you encounter Client-Server Architecture in everyday life. Here are three common examples:

- 1) Email Servers:** Email has evolved into the primary communication method for businesses due to its speed and convenience. Various server components work together to deliver emails between users across different mail servers.
- 2) File Servers:** When saving documents on services like Google Docs or Microsoft Office, you're interacting with file servers. These servers store data centrally and allow multiple clients to access it.
- 3) Web Servers:** These high-powered servers host websites, which web clients access through DNS or an IP address. Here's a simplified process:
 - a) A user enters a URL in the browser.
 - b) The browser requests the IP address from the Domain Name System (DNS) .
 - c) The DNS server provides the IP address to the browser.
 - d) The browser sends an HTTPS or HTTP request to the web server.
 - e) The server sends back the requested files.
 - f) The user retrieves the files, and the process continues as needed.

These examples highlight how Client-Server Architecture forms the foundation of many digital services we rely on daily.

Types of Client-Server Architecture



1) One-tier Architecture

A self-contained application on a single platform. In one-tier architecture, the client, server, and database are all on the same machine. The client handles user interaction and business logic, the server provides services like data storage and processing, and the database manages data. While simple and popular for small apps, this architecture is rarely used in production because it doesn't meet most system requirements.

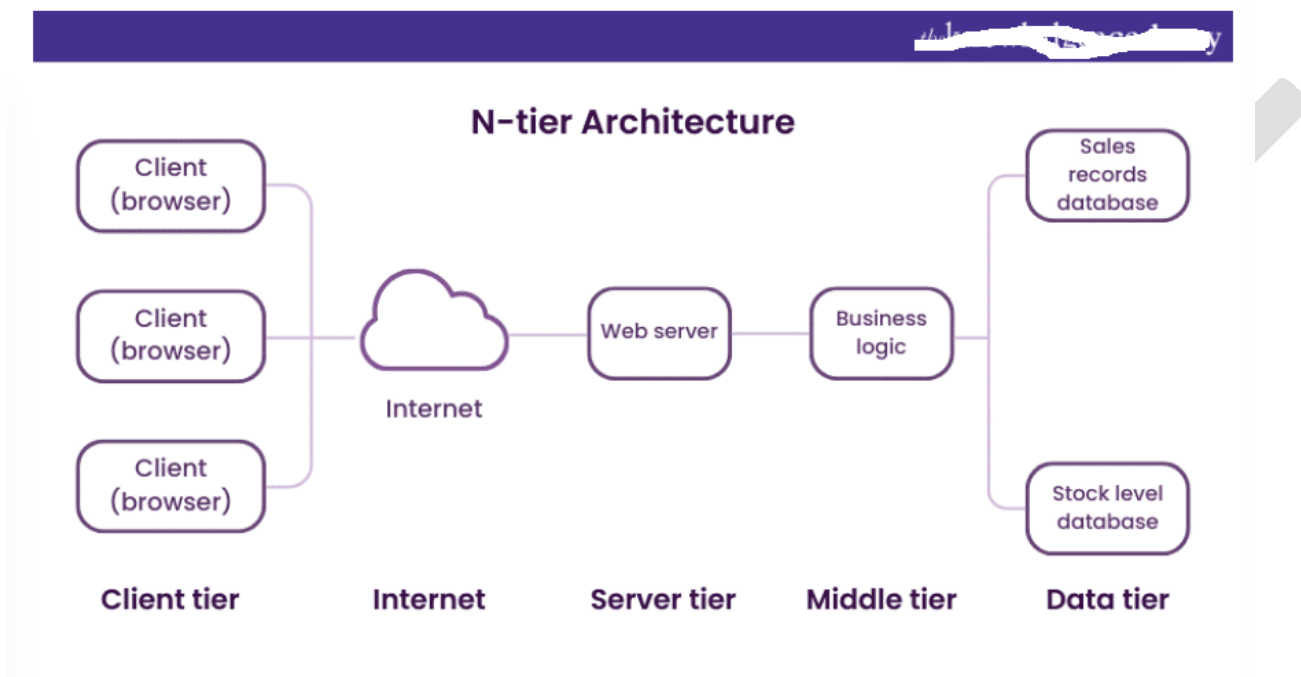
2) Two-tier Architecture

This basic Client-Server Architecture involves direct communication between the client and server without an intermediate layer. The client manages the User Interface (UI) and business logic, while the server handles data storage and processing. An example is a web browser requesting pages from a web server, which responds with HTML files. It's easy to implement but has drawbacks like low scalability, high network traffic, and security risks.

Three-tier Architecture

A more complex client-server setup with an intermediate layer (usually middleware or an application server) that handles business logic, acting as a bridge between the client and server. The client deals with the user interface (UI), while the server manages data storage. An example is an online banking system where the client is a web browser, the middleware checks transactions, and the server stores account data. This architecture improves scalability, performance, and security but increases complexity and cost.

N-tier Architecture



A more flexible Client-Server Architecture with more than three tiers, allowing greater scalability, flexibility, and modularity. Each tier can be distributed across different machines or networks and updated independently. An example is an e-commerce system with a web browser displaying the product catalog, a web server handling HTTP requests, an application server processing business logic, and a database storing product information. While suitable for complex systems, it requires more resources and management.

In addition to these types, there are also some variations or subtypes of Client-Server Architecture, such as:

Peer-to-peer Architecture

This Client-Server Architecture allows each node to act as both a client and a server, with no central authority or hierarchy. Nodes can directly request or provide services to each other. An example is BitTorrent, a file-sharing system where nodes can download or upload files from each other without a central server, enhancing reliability, availability, and resilience. However, it presents challenges like security, coordination, and quality control.

Thin-client Architecture

In this architecture, the client has minimal functionality, relying on the server for processing and data storage. The client handles only the user interface (UI) and sends inputs to the server, which performs all computations and returns the results. An example is Google Docs, where the client is a web browser accessing the document editor while the server stores and updates data in the cloud. This architecture reduces client-side hardware costs, maintenance, and security risks but increases network dependency, bandwidth usage, and server load.

Fat-client Architecture

In a fat-client setup, the client is highly functional, performing most processing and data storage locally, and only occasionally communicates with the server for synchronisation or backup. An example is an offline mobile app like Evernote, where the client runs the app and stores notes locally, syncing with the cloud when connected. This architecture boosts performance, responsiveness, and availability on the client side but increases hardware requirements, software complexity, and risks of data inconsistency.

Unlock the power of OpenStack on Linux – Join our Linux OpenStack Administration Training now and become an OpenStack Administration expert!

Advantages and disadvantages of Client-Server Architecture

Client-Server Architecture has some advantages and disadvantages, depending on the requirements and constraints of the system.

Advantages

Some of the advantages are:

- 1) It is centralised, which means that all data and services are stored and managed in a single place. This makes it easier to maintain, update, and secure the system.
- 2) It is cost-efficient, as it requires less hardware and software resources for the client side. The client only needs a network connection and an application or web browser to access the server.
- 3) It has high performance and low latency, as the server can handle many requests from many clients simultaneously and efficiently.

Disadvantages

Some of the disadvantages are:

Web Development Using PHP and MySQL-MMCB311C

- 1) It has limited scalability, as it depends on the capacity and availability of the server. If the server is overloaded or fails, the system may not function properly or at all.
- 2) It has high network dependency, as it relies on the network connection between the client and the server. If the network is slow or disrupted, the system may experience delays or errors.
- 3) It has complex architecture, as it involves multiple components and layers that need to be designed, implemented, and coordinated. The system may also face challenges such as security, synchronisation, and compatibility.

Unlock the power of Microservices: Join our Microservices Architecture Training today and transform your software development!

HTML Common tags:-

HTML is the building block for web pages. HTML is a format that tells a computer how to display a web page. The documents themselves are plain text files with special "tags" or codes that a web browser uses to interpret and display information on your computer screen.

- HTML stands for Hyper Text Markup Language
- An HTML file is a text file containing small markup tags
- The markup tags tell the Web browser how to display the page
- An HTML file must have an htm or html file extension.

HTML Tags:- HTML tags are used to mark-up HTML elements .HTML tags are surrounded by the two characters < and >. The surrounding characters are called angle brackets. HTML tags normally come in pairs like **and** The first tag in a pair is the start tag, the second tag is the end tag . The text between the start and end tags is the element content . HTML tags are not case sensitive, **means the same as**.

The most important tags in HTML are tags that define headings, paragraphs and line breaks.

Tag	Description
<!DOCTYPE...>	This tag defines the document type and HTML version.
<html>	This tag encloses the complete HTML document and mainly comprises of document header which is represented by <head>...</head> and document body which is represented by <body>...</body> tags.
<head>	This tag represents the document's header which can keep other HTML tags like <title>, <link> etc.
<title>	The <title> tag is used inside the <head> tag to mention the document title.

<body>	This tag represents the document's body which keeps other HTML tags like <h1>, <div>, <p> etc.
<p>	This tag represents a paragraph.
<h1> to <h6>	Defines header 1 to header 6
 	Inserts a single line break
<hr>	Defines a horizontal rule
<!-->	Defines a comment

Headings:-

Headings are defined with the <h1> to <h6> tags. <h1> defines the largest heading while <h6> defines the smallest.

<h1>This is a heading</h1>

<h2>This is a heading</h2>

<h3>This is a heading</h3>

<h4>This is a heading</h4>

<h5>This is a heading</h5>

<h6>This is a heading</h6>

Paragraphs:-

Paragraphs are defined with the <p> tag. Think of a paragraph as a block of text. You can use the align attribute with a paragraph tag as well.

<p align="left">This is a paragraph</p>

<p align="center">this is another paragraph</p>

Note: You must indicate paragraphs with <p> elements. A browser ignores any indentations or blank lines in the source text. Without <p> elements, the document becomes

Line Breaks:-


The
 tag is used when you want to start a new line, but don't want to start a new paragraph. The
 tag forces a line break wherever you place it. It is similar to single spacing in a document.

This Code	output
<p>This is a para graph with line breaks</p>	This is a para graph with line breaks

Web Development Using PHP and MySQL-MMCB311C

Horizontal Rule The element is used for horizontal rules that act as dividers between sections like this:

The horizontal rule does not have a closing tag. It takes attributes such as align and width

Code	Output
<code><hr width="50%" align="center"></code>	

Sample html program

```
<!DOCTYPE html>
<html>
  <head>
    <title>This is document title
    </title>
  </head>
  <body>
    <h1>This is a heading</h1>
    <p>Document content goes here.....</p>
  </body>
</html>
```



- Type the above program in notepad and save with some file name eg:sample.html
- Open the file with browser and the webpage looks like this

Lists:-HTML offers web authors three ways for specifying lists of information.

All lists must contain one or more list elements. Lists are of three types

- 1) Un ordered list
- 2) Ordered List
- 3) Definition list

HTML Unordered Lists:An unordered list is a collection of related items that have no special order or sequence. This list is created by using HTML tag. Each item in the list is marked with a bullet.

Example

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML Unordered List</title>
  </head>
  <body>
    <ul>
      <li>Beetroot</li>
```



- Beetroot
- Ginger
- Potato
- Radish

HTML Ordered Lists:- items are numbered list instead of bulleted, This list is created by using `` tag.

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML Ordered List</title>
  </head>
  <body>
    <ol>
      <li>Beetroot</li>
      <li>Ginger</li>
```

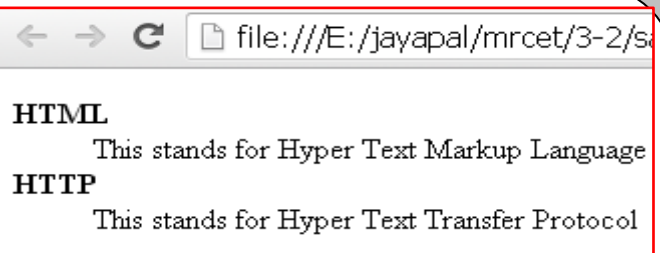


1. Beetroot
2. Ginger
3. Potato
4. Radish

HTML Definition Lists:- HTML and XHTML supports a list style which is called definition lists where entries are listed like in a dictionary or encyclopedia. The definition list is the ideal way to present a glossary, list of terms, or other name/value list. Definition List makes use of following three tags.

- 1). `<dl>` - Defines the start of the list
- 2). `<dt>` - A term
- 3). `<dd>` - Term definition
- 4). `</dl>` - Defines the end of the list

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML Definition List</title>
  </head>
  <body>
    <dl>
```



HTML tables:

The HTML tables allow web authors to arrange data like text, images, links, other tables, etc. into rows and columns of cells. The HTML tables are created using the **<table>** tag in which the **<tr>** tag is used to create table rows and **<td>** tag is used to create data cells.

Example:

```
<!DOCTYPE html>

<html>

<head>

<title>HTML Tables</title>

</head>

<body>
```

```
    <table border="1">
```

```
        <tr>
```

```
            <td>Row 1, Column 1</td> <td>Row 1, Column 2</td>
```

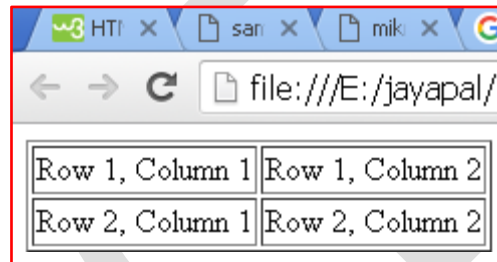


Table Heading: Table heading can be defined using **<th>** tag. This tag will be put to replace

<td> tag, which is used to represent actual data cell. Normally you will put your top row as table heading as shown below, otherwise you can use <th> element in any row.

Tables Backgrounds: set table background using one of the following two ways:

- 1) bgcolor attribute - You can set background color for whole table or just for one cell.
- 2) background attribute - You can set background image for whole table or just for one cell. You can also set border color also using bordercolor attribute.

```
<!DOCTYPE html>

<html>

<head>

<title>HTML Tables</title> </head>

<body>

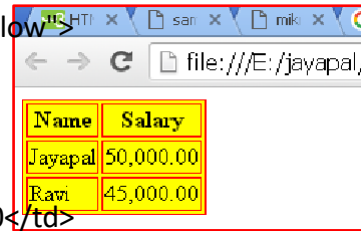
    <table border="1"bordercolor="red" bgcolor="yellow">

        <tr> <th>Name</th>

        <th>Salary</th> </tr>

        <tr>

            <td>Jayapal    </td> <td>50,000.00</td>
```



Name	Salary
Jayapal	50,000.00
Ravi	45,000.00

Images are very important to beautify as well as to depict many complex concepts in simple way on your web page.

Insert Image:

insert any image in the web page by using tag.

Attribute Values

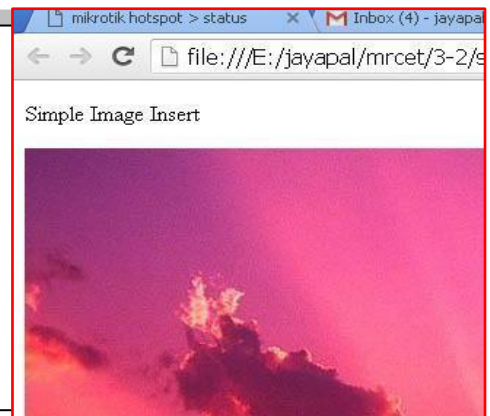
Value	Description
left	Align the image to the left
right	Align the image to the right

<html>

 <head>

 <title>Using Image in Webpage</title>

 </head>
```



## HTML FORMS:

HTML Forms are required to collect some data from the site visitor. For example, during user registration you would like to collect information such as name, email address, credit card, etc. A form will take input from the site visitor and then will post it to a back-end application such as CGI, ASP Script or PHP script etc. The back-end application will perform required processing on the passed data based on defined business logic inside the application. There are various form elements available like text fields, text area fields, drop-down menus, radio buttons, checkboxes, etc.

```
<form action="Script URL" method="GET|POST"> form elements like input, text area etc. </form>
```

## Form Attributes

Apart from common attributes, following is a list of the most frequently used form attributes:

Attribute	Description
action	Backend script ready to process your passed data.
method	Method to be used to upload data. The most frequently used are GET and POST methods.
target	Specify the target window or frame where the result of the script will be displayed. It takes values like _blank, _self, _parent etc.
enctype	You can use the enctype attribute to specify how the browser encodes the data before it sends it to the server. Possible values are:  application/x-www-form-urlencoded - This is the standard method most forms use in simple scenarios.  multipart/form-data - This is used when you want to upload binary data in the form of files like image, word file etc.

## HTML Form Controls :

There are different types of form controls that you can use to collect data using HTML

``

- Text Input Controls
- Checkboxes Controls
- Radio Box Controls
- Select Box Controls
- File Select boxes
- Hidden Controls
- Clickable Buttons
- Submit and Reset Button

## Text Input Controls:-

There are three types of text input used on forms:

- 1) **Single-line text input controls** - This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML `<input>` tag.

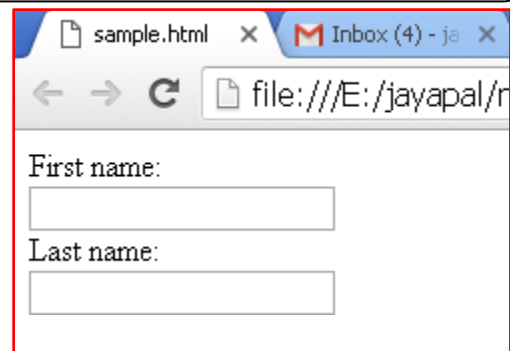
**`<input type="text">`** defines a one-line input field for **text input**:

Example:

```
<form>
 First name:

 <input type="text"
 name="firstname">
 Last name:

 <input type="text" name="lastname">
```



- 2) **Password input controls** - This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTML `<input>` tag.

Input Type Password



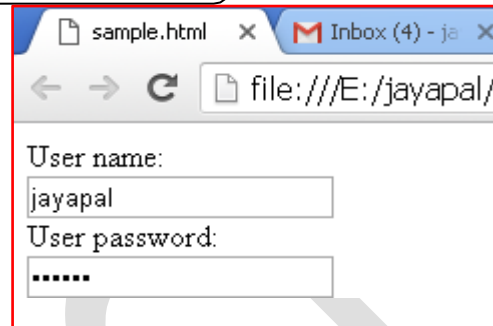
`<input type="password">` defines a **password field**:

`<form>`

User name:<br>

`<input type="text" name="username"><br>`

User password:<br>



**3) Multi-line text input controls** - This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML `<textarea>` tag.

```
<!DOCTYPE html>

<html>

 <head>

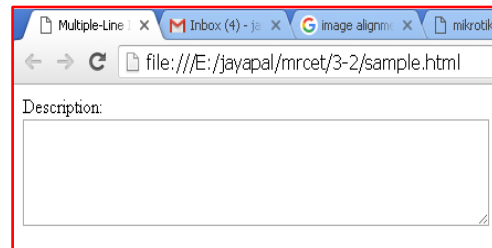
 <title>Multiple-Line Input Control</title>

 </head>

 <body>

 <form> Description:

```



### Checkboxes Controls:-

Checkboxes are used when more than one option is required to be selected. They are also created using HTML `<input>` tag but type attribute is set to checkbox.

Here is an example HTML code for a form with two checkboxes:

```
<!DOCTYPE html>

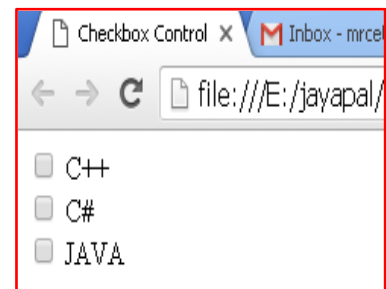
<html> <head> <title>Checkbox Control</title> </head>

<body>

 <form>

 <input type="checkbox" name="C++" value="on"> C++


```



### Radio Button Control:-

Radio buttons are used when out of many options, just one option is required to be selected. They are also created using HTML `<input>` tag but type attribute is set to radio.

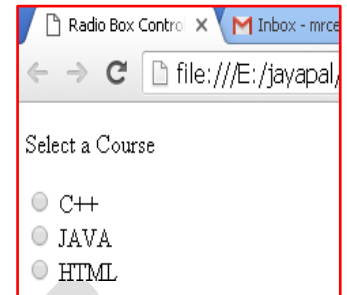
```
<!DOCTYPE html>
```

```
<html> <head> <title>Radio Box Control</title> </head>
```

```
 <body> <p>Select a Course</p>
```

```
 <form>
```

```
 <input type="radio" name="subject" value="C++"> C++
```



**Select Box Controls :-** A select box, also called drop down box which provides option to list down various options in the form of drop down list, from where a user can select one or more options.

```
<!DOCTYPE html>

<html>

<head>

<title>Select Box Control</title>

</head>

<body>

<form>

<select name="dropdown">

<option value="C++" selected>C++</option>

<option value="JAVA">JA VA</option>

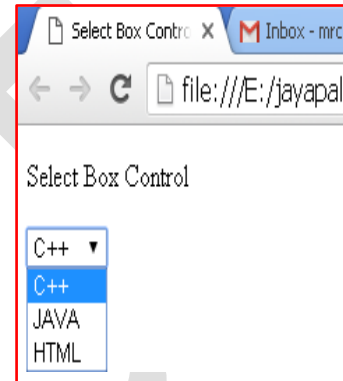
<option value="HTML">HTML</option>

</select>

</form>

</body>

</html>
```



**File Select boxes:-** If you want to allow a user to upload a file to your web site, you will need to use a file upload box, also known as a file select box. This is also created using the

`<input >` element but type attribute is set to **file**.

```
<!DOCTYPE html>

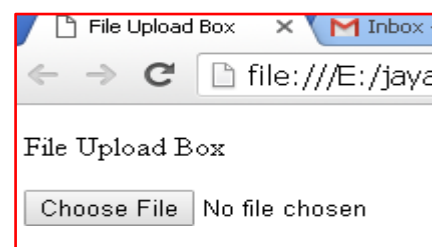
<html>

<head>

<title>File Upload Box</title>

</head>

<body>
```



**Hidden Controls:-** Hidden form controls are used to hide data inside the page which later on can be pushed to the server. This control hides inside the code and does not appear on the actual page. For example, following hidden form is being used to keep current page number. When a user will click next page then the value of hidden control will be sent to the web server and there it will decide which page will be displayed next based on the passed current page.

```
<html> <head> <title>File Upload Box</title> </head>

 <form>

 <p>This is page 10</p>

 <input type="hidden" name="pagename" value="10" />
```