**#What is internet**

The internet is a global network of inter connected. Computers that allow users to share and access information communicate, and interact with each other. It works by connecting millions of Private, Public, academic, business and government networks around the world. These networks use a common Set of Protocols (such as TCP/IP) to transmit data enabling. Services like websites meal, social media, and online gaming the internet has transformed how people work, learn, and socialize, making vast amounts of information and Services accessible from almost anywhere.

**#What is HTML**

HTML (Hypertext Markup Language) is the Standard language used to create and Structure content on the web. It is the foundation of Web Page, alle developers to define the structure and layout of a Webpage by Using a Serdes of elements. and tags. These tags are used of format feet, embed images create links, and organize content into headings, Paragraphs, lists, and other elements

**#What is home - Page**

A home page is the main or introductory. Page of a website it serves as the Starting point for visitors Providing an overview of what the website is about and of ten including links to other Important sections or content. The home Page is designed to guide users to the most relevant information helping them navigate the site easily

Key features

1) Navigation menu

2) Welcome Message or Introduction.

3) Featured Content

4) Search Bar

5) Contant Information

The home page is typically the first Page users see when they visit a website, making it crucial for cracking a good first impression

**#What is website**

A Website is a collection of related Web Pages that are typically identified by a Common domain name and published on at least one web server. Website core be accessed through the internet and are designed to Present information, Provide Services, or facilitate Commotions cation to users Websites serve various purposes and can be Personal, Commercial, educational governmental or nonportal.

**# Whal is www**

www stands for the World Wide Web, which is a system of interlinked hypertext documents and multimedia content accessed via the Internet browser (like Chrome, Firefox, or Safari), you are accessing the world wide web

**What is a Website Protocol?**

A **website protocol** is a set of rules that determines how data is transmitted over the internet. These protocols ensure that different devices and systems can communicate with each other in a standard way. The most common protocols used on websites are **HTTP** (Hypertext Transfer Protocol) and **HTTPS** (Hypertext Transfer Protocol Secure).

**Key Points about HTTP and HTTPS:**

1. **HTTP (Hypertext Transfer Protocol):**
   * It's the foundation of data communication on the World Wide Web.
   * When you visit a website starting with "http://", it uses HTTP.
   * HTTP is not secure, meaning the data transferred between your browser and the server is not encrypted.
2. **HTTPS (Hypertext Transfer Protocol Secure):**
   * It's the secure version of HTTP.
   * When you visit a website starting with "https://", it uses HTTPS.
   * HTTPS uses encryption to secure data transfer, making it safer from interception by third parties.

**What is SMTP?**

**SMTP** stands for **Simple Mail Transfer Protocol**, It is a protocol used for sending emails across the internet, at operates at the **Application layer** of the OSI (open systems interconnection) model.

### ****Gopher****

Gopher is a protocol designed for distributing, searching, and retrieving documents over the Internet. It was developed in 1991 at the University of Minnesota, at operates at the **Application layer** of the OSI (open systems interconnection) model.

**Telnet**

Telnet is a network protocol that allows a user to communicate with a remote device or server, at operates at the **Application layer** of the OSI (open systems interconnection) model.

**TFTP (Trivial File Transfer Protocol)** is a simple, lightweight file transfer protocol used primarily for transferring small files in network environments. Unlike FTP, TFTP operates without authentication and uses UDP (User Datagram Protocol) on port 69, making it faster but less secure. It's commonly used for tasks like booting devices over a network, transferring configuration files, or updating firmware in embedded systems. Due to its simplicity, TFTP lacks advanced features like encryption and error handling, which makes it unsuitable for transferring sensitive or large data over insecure networks.

**SNMP (Simple Network Management Protocol)**

SNMP is a protocol used for managing devices on a network, at operates at the **Application layer** of the OSI (open systems interconnection) model.

**What is an Email?**

**Email** (short for *electronic mail*) is a method of exchanging digital messages over the internet. It allows people to send and receive messages, documents, images, and other files quickly and easily.

**How Emails Work**

Emails are sent from one computer to another using email servers. Here’s a simplified process of how an email is sent and received:

1. **Composing the Email**: You write a message and click "Send." The email is sent to your email provider's outgoing mail server.
2. **Outgoing Mail Server (SMTP)**: The email goes through the Simple Mail Transfer Protocol (SMTP) server, which forwards it to the recipient’s incoming mail server.
3. **Incoming Mail Server (POP3/IMAP)**: The recipient’s incoming mail server receives the email and stores it until the recipient checks their email.
4. **Receiving the Email**: The recipient checks their email, and the email is downloaded from the server to their email client (like Gmail, Outlook, etc.).

**A modem** (short for **Modulator-Demodulator**) is a hardware device that enables a computer or another device to connect to the internet by converting digital data from a computer into analogy signals that can be transmitted over traditional phone lines or cable systems, and vice versa.

**Client-Server Computing** is a model where tasks and services are distributed between two types of entities: clients and servers.

* **Client**: The client is the device (computer, smartphone, etc.) that requests services or resources from the server. It sends a request over the network for tasks like retrieving data or performing computations.
* **Server**: The server is a more powerful system that stores data, manages resources, and processes client requests. It listens for client requests and provides the appropriate services, such as database access, file storage, or web services.

**Key Concepts:**

1. **Separation of Concerns**: The server handles resource management and data processing, while the client focuses on presenting data and interacting with the user.
2. **Networking**: Communication between client and server occurs over a network, typically using protocols like HTTP, FTP, or SMTP.
3. **Stateless vs. Stateful**: Some client-server interactions are stateless (e.g., HTTP requests, where each request is independent), while others maintain state (e.g., sessions in web applications).
4. **Scalability**: Servers can serve multiple clients simultaneously, making the system scalable. Server performance can be enhanced by adding more resources (horizontal scaling) or more powerful hardware (vertical scaling).
5. **Security**: Often, security protocols like SSL/TLS are used to ensure secure communication between client and server, especially in sensitive transactions.
6. **Examples**: Common examples include web browsers (client) accessing websites (server), email services, file-sharing systems, and cloud computing platforms.

The client-server model is fundamental in modern computing, allowing efficient resource sharing, centralized control, and easier maintenance.

**1. Client-Server Model**

The **client-server model** is a network architecture that separates computers (or processes) into two roles: clients and servers. This model is fundamental to how many networked applications, including the web, operate.

* **Client**: A client is a computer or software application that requests a service or resource from a server. Clients initiate communication and send requests to servers for data, files, or processing power.
* **Server**: A server is a computer or software application that provides services or resources to clients. Servers listen for requests from clients and respond by providing the requested data or performing the requested tasks.

**How It Works**:

* Clients and servers communicate over a network, typically using the Internet or a local area network (LAN).
* The client sends a request to the server.
* The server processes the request and sends back the appropriate response.

**Examples**:

* **Web Browsing**: A web browser (client) requests web pages from a web server.
* **Email**: An email client (like Outlook) requests to send or receive messages from an email server.

**2. Web Client**

A **web client** is any device or software that accesses a web server over a network to request web resources. The most common type of web client is a web browser, but other applications like mobile apps or command-line tools (e.g., curl or wget) can also act as web clients.

**Characteristics of a Web Client**:

* **Initiates Requests**: Sends HTTP/HTTPS requests to web servers to access resources like HTML pages, images, videos, or APIs.
* **Displays or Processes Data**: Receives the data sent by the server and displays it (as in a web browser) or processes it (as in a mobile app or API client).

**3. Web Server**

A **web server** is software and hardware that serves web pages and other web resources to clients over the Internet or an intranet.

**Key Functions**:

* **Handles Requests**: Listens for incoming HTTP/HTTPS requests from web clients.
* **Serves Content**: Retrieves requested content from the server’s file system or generates it dynamically (via server-side scripts) and sends it back to the client.
* **Manages Connections**: Controls how multiple clients connect to and interact with the server simultaneously.

**Examples of Web Servers**:

* **Apache HTTP Server**: A widely used open-source web server software.
* **Nginx**: Known for its high performance and ability to handle a large number of simultaneous connections.
* **Microsoft Internet Information Services (IIS)**: A web server created by Microsoft for use with the Windows NT family.

**4. Web Browser**

A **web browser** is a software application used to access and display web pages and other content on the web. It acts as a web client in the client-server model.

**Functions of a Web Browser**:

* **Requesting Web Pages**: Sends HTTP requests to web servers for web pages or resources.
* **Rendering Content**: Interprets HTML, CSS, JavaScript, and other web technologies to render the web page on the screen.
* **User Interface**: Provides an interface for navigating the web, managing bookmarks, and other functions like history and downloads.

**Examples of Web Browsers**:

* **Google Chrome**: A widely used web browser known for its speed and extensive extensions.
* **Mozilla Firefox**: An open-source browser known for its privacy features and flexibility.
* **Microsoft Edge**: The default web browser for Windows, built on the Chromium engine.
* **Internet Explorer** was a web browser developed by Microsoft, first released in 1995. It was one of most widely used browsers in the early 2000s but was eventually replaced by Microsoft Edge and discontinued in 2022.
* **Safari**: Apple's web browser for macOS and iOS devices.
* **Netscape Navigator** was one of the earliest and most popular web browsers during the early days of the Internet. Launched in 1994 by Netscape Communications Corporation, it played a key role in popularizing web browsing for the general public.

**Summary**

* The **client-server model** divides roles into clients (requesters) and servers (providers).
* A **web client** initiates requests to a web server, often through a web browser.
* A **web server** responds to client requests, serving web pages and resources.
* A **web browser** is a type of web client that displays web pages and interacts with web servers.

**The Birth of HTML**

It was the year **1990** when **Tim Berners-Lee**, a British computer scientist working at **CERN (European Organization for Nuclear Research)**, began developing a way to share and link documents across computers. He envisioned a system that could be easily accessed and understood by anyone, and this idea led to the creation of **HTML (HyperText Markup Language).**

Tim wanted a way for scientists to share their research papers and access documents no matter where they were in the world. To do this, he created the first web page and browser. The invention of HTML allowed people to create web pages with linked documents, which would eventually lead to the development of the **World Wide Web**.

**How HTML Works: Tags and Structure**

HTML is the language that tells a web browser how to display a web page. It works using **tags** that structure content. Tags are enclosed in angle brackets < > and usually come in pairs: an **opening tag** and a **closing tag**.

1. **Opening tag**: This tells the browser where the HTML element starts.

- Example: <p> (for a paragraph)

2. **Closing tag**: This tells the browser where the element ends, marked with a slash /.

- Example: </p> (closes the paragraph tag)

Between the opening and closing tags, you can put text, images, links, and more.

[**Example………**](../../../Documents/IIMT%20code/HTML)

**<html>:** This is the root element that contains all the HTML code.

**<title>:** specifies the webpage title displayed on the browser tab.

**<head>:** This section contains meta-information, like the title of the page.

**<body>:** This section contains the visible content of the page.

**<h1>:** This tag defines a heading.

**<p>:** This tag defines a paragraph.

Etc……………….

[**Example(4**](file:///C:\My%20Data\Documents\IIMT%20code\HTML)**)**

This HTML code is structured to create a webpage titled "Centralized Data" with content centered both horizontally and vertically using HTML. Let's break it down:

**Key Sections Explained:**

1. **<html> and <head>:**
   * The <html> tag wraps the entire document, indicating the start and end of an HTML page.
   * Inside the <head>, the <title>Centralized Data</title> sets the title that will appear on the browser tab (not on the page itself).
2. **<body>:**
   * The <body> contains the visible content of the webpage.
3. **Centering Content with a Table:**
   * **<table width="100%" height="100%">**: This creates a table that spans the entire width and height of the browser window. The table layout is used to center content both vertically and horizontally.
   * **<td align="center" valign="middle">**:
     + align="center" horizontally centers the content inside the <td> (table data cell).
     + valign="middle" vertically aligns the content in the middle of the cell, thus achieving full-page centering.
4. **Content Inside the Table Cell:**
   * **<h1>Centralized Data</h1>**: This is a heading inside the centered table cell. The heading level <h1> makes the text bold and large.
   * **<p>This content is centered using only HTML</p>**: A paragraph below the heading, which contains a short message explaining that centering was achieved using only HTML.

**Client-Side Scripting Languages**

* **Client-Side Scripting** runs in the user’s browser and enhances interactivity without needing server interaction.
* **VBScript**: A scripting language developed by Microsoft, mainly used for Internet Explorer. Similar to Visual Basic.
* **JavaScript**: A widely-used scripting language, compatible with multiple browsers, allowing dynamic and interactive content on web pages.

**ActiveX Control and Plug-ins**

* **ActiveX**: A Microsoft technology that enables interactive content such as videos, games, and forms on websites, mostly used in Internet Explorer.
* **Plug-ins**: External applications or components that extend the capabilities of web browsers (e.g., Adobe Flash, Java Applets).

**Web Server Architecture**

* The structure that defines how a web server handles client requests. It includes:
  + **Static content** (HTML files)
  + **Dynamic content** (generated by scripts and applications)
  + Load balancing, caching, and security layers.

**Image Maps**

* A feature in HTML that defines clickable areas on an image, allowing navigation to different web pages or content based on where the user clicks.

**CGI (Common Gateway Interface)**

* A standard protocol used to enable web servers to execute programs or scripts (written in Perl, Python, etc.) to generate dynamic content.

**API (Application Programming Interface)**

* Set of protocols and tools for building software and web applications. It enables different software components to communicate. For web applications, it allows integration of services like social media, payment gateways, etc.

Here are some popular APIs:

1. **Google Maps API** – Used for embedding maps, geolocation, and other location-based services.
2. **Twitter API** – Allows access to Twitter data for posting tweets, reading user data, and interacting with the platform.
3. **GitHub API** – Allows interaction with GitHub repositories, commits, and other user activity.

Top of Form

Bottom of Form

**Web Database Connectivity**

* Refers to the methods used for connecting a web application to a database to manage and retrieve data.
  + **DBC (Database Connectivity)**: General term for connecting databases with applications.
  + **ODBC (Open Database Connectivity)**: A standard API for accessing database management systems (DBMS), providing a way to write database-independent applications.