**INTERNET:**

* The Internet can be defined as a global network of interconnected computers and devices that use a standardized set of protocols to transmit and exchange data.
* It is a vast infrastructure that allows for communication, information sharing, and access to a wide range of resources and services.
* The Internet operates on the basis of the TCP/IP protocol suite, which enables the transfer of data packets across networks, facilitating the seamless connectivity of computers and devices worldwide.
* It has become an essential tool for communication, research, entertainment, commerce, and various other aspects of modern life.

**Components of the Internet:**

a) **Infrastructure**: The Internet relies on a complex infrastructure, including undersea cables, satellites, routers, and servers, which enable data transmission across the globe.

b) **Protocols**: Protocols like TCP/IP (Transmission Control Protocol/Internet Protocol), HTTP (Hypertext Transfer Protocol), and DNS (Domain Name System) facilitate communication and data transfer over the Internet.

c) **Devices**: Computers, smartphones, tablets, and various Internet of Things (IoT) devices form the interconnected network that enables access to the Internet.

**Functions and Features of the Internet:**

a) **Communication**: The Internet enables various forms of communication, such as email, instant messaging, voice and video calls, connecting people globally in real-time.

b) **Information access**: Websites, search engines, and online databases allow users to access vast amounts of information, facilitating research, learning, and knowledge sharing.

c) **Social networking**: Social media platforms and online communities provide avenues for connecting, sharing experiences, and fostering virtual communities.

d) **Entertainment**: Streaming services for music, movies, and TV shows, as well as online gaming platforms, offer a wide range of entertainment options for users.

e) **E-commerce**: The Internet has transformed the way we shop and conduct business, with online stores and electronic payment systems offering convenience and global reach.

**INTERNET VS WEB:**

1. *The Internet* is a global network of networks while *the Web*, also referred to formally as [World Wide Web (www)](https://www.geeksforgeeks.org/world-wide-web-www/) is a collection of information that is accessed via *the*[*Internet*](https://news.geeksforgeeks.org/knowledge/history-of-internet).
2. Another way to look at this difference is; *the Internet* is infrastructure while *the Web* is served on top of that infrastructure.
3. Alternatively, *the Internet* can be viewed as a big book store while *the Web* can be viewed as a collection of books on that store.
4. At a high level, we can even think of *the Internet* as hardware and *the Web* as software.
5. Web applications use [HTTP](https://www.geeksforgeeks.org/understanding-http-using-browsers/) protocol which is a layer over [TCP protocol](https://www.geeksforgeeks.org/what-is-transmission-control-protocol-tcp/). Whereas internet applications can use either TCP or [UDP protocol](https://www.geeksforgeeks.org/user-datagram-protocol-udp/).

**INTERNET TERMINOLOGIES**

1. **Client**: Any computer on the network that requests services from another Computer on the network.
2. **Server**: Any computer that receives requests from client computers, processes and sends the output.
3. **Web Page**: Any page that is hosted on the Internet.
4. **Web Development:** The process of creating, modifying web pages.
5. **Web server:** software that listens for web page requests. E.g: Apache, Microsoft
6. Internet Information Server (IIS) (part of Windows)

**HISTORY OF INTERNET:**

1. **ARPANET**: The Advanced Research Projects Agency Network (ARPANET) was created in **1969** by the U.S. Department of Defense. It was served as the foundation for the Internet. ARPANET connected computers at various research institutions and universities.
2. **TCP/IP Protocol**: In the 1970s, the Transmission Control Protocol/Internet Protocol (TCP/IP) was developed.
3. **Email and File Transfer:** In the early 1970s, Ray Tomlinson developed the first email program, enabling users to send messages between computers. File Transfer Protocol (FTP) was also developed during this time, allowing users to share files across networks.
4. **Domain Name System (DNS):** The DNS was introduced in 1983 to simplify the process of accessing websites. Instead of relying on numerical IP addresses, DNS provided a hierarchical naming system, making it easier for users to remember and access websites by their domain names.
5. **World Wide Web:** In 1989, Tim Berners-Lee, a British scientist, proposed the concept of the World Wide Web. He developed the necessary technologies, including HTML (Hypertext Markup Language), HTTP (Hypertext Transfer Protocol), and the first web browser, called WorldWideWeb.
6. **Netscape Navigator**: Developed by Netscape Communications Corporation, Netscape Navigator became one of the most widely used web browsers in the mid-1990s. It played a key role in popularizing the Internet and contributed to the rapid growth of the World Wide Web.
7. **Internet Explorer:** Released by Microsoft in 1995, Internet Explorer quickly gained popularity, becoming the dominant web browser in the late 1990s and early 2000s.
8. **Social Media and Web 2.0:** Around the mid-2000s, the Web transformed into a more interactive and participatory platform, often referred to as Web 2.0. Social media platforms like Facebook, Twitter, and YouTube gained prominence, allowing users to create, share, and interact with content.
9. **Mozilla Firefox**: Firefox, developed by the Mozilla Foundation, was first released in 2004 as an open-source web browser.
10. **Google Chrome:** Introduced by Google in 2008, Chrome rapidly became one of the most popular web browsers.

**INTRANET:**

Intranet is a private computer network that operates within an organization or a specific group, allowing its members to share information, resources, and collaborate.

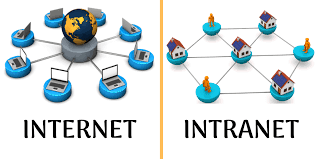
It functions similarly to the Internet but is restricted to a specific internal network, accessible only by authorized individuals within the organization.

**Advantages of Intranet:**

1. **Improved Communication:** Intranets provide a centralized platform for internal communication within an organization. It allows employees to share information, collaborate on projects, and stay updated on company news and announcements.
2. **Enhanced Collaboration:** Intranets facilitate collaboration among employees by providing tools for document sharing, version control, and team collaboration. This streamlines workflows, promotes teamwork, and improves productivity.
3. **Easy Access to Information**: Intranets serve as a repository of important information, policies, procedures, and resources. Employees can access this information quickly and easily, saving time and effort compared to traditional methods like searching through physical documents or contacting colleagues for information.
4. **Secure Environment:** Intranets are private networks, meaning that access is restricted to authorized individuals within the organization. This ensures that sensitive company information remains secure and protected from external threats.
5. **Cost Savings:** Intranets can help reduce costs associated with printing, distributing physical documents, and conducting in-person meetings. By providing an online platform for information sharing and communication, organizations can save on paper, printing, and travel expenses.

**Disadvantages of Intranet:**

1. **Implementation and Maintenance:** Setting up and maintaining an intranet can be a complex process that requires technical expertise. It may involve initial investment in hardware, software, and IT infrastructure. Ongoing maintenance and updates are also necessary to keep the intranet running smoothly.
2. **Training and User Adoption:** Introducing an intranet may require training employees on how to use the platform effectively. Some employees may find it challenging to adapt to the new system, leading to slower adoption and potential resistance to change.
3. **Accessibility Limitations:** Intranets typically require employees to have access to a computer or device connected to the internal network. This may limit accessibility for employees who work remotely or do not have regular access to the intranet.
4. **Content Management:** Managing and organizing content within an intranet can be a challenge. Without proper content management processes in place, finding relevant information can become difficult, leading to frustration and decreased efficiency.
5. **Potential Security Risks:** While intranets are generally secure, there is still a risk of internal security breaches. Organizations must implement robust security measures, such as user access controls, encryption, and regular security audits, to mitigate these risks.

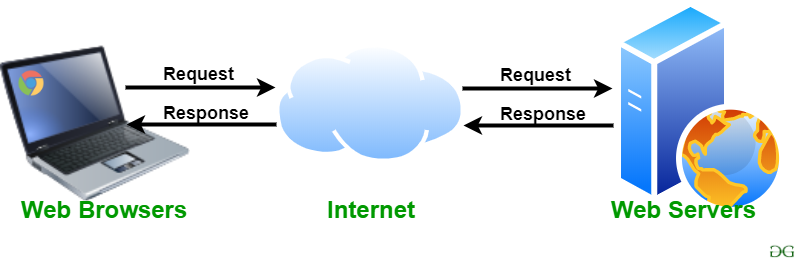


**WEB BROWSER:**

1. A web browser takes you anywhere on the internet.
2. It retrieves information from other parts of the web and displays it on your desktop or mobile device.
3. The information is transferred using the Hypertext Transfer Protocol, which defines how text, images and video are transmitted on the web.

**WEB SERVER:**

1. Web Server is a piece of software running on a computer whose primary job is to distribute web pages to users whenever they demand it and provides an area in which to store and organize the pages of the website.
2. The machine that executes the web server software can be a remote machine placed at the other side of your network or even on the other end of the globe, or it be your very own personal computer at home.
3. We also introduced the idea that the user’s browser was the client in this relationship.



**Difference between Web Browser and Web Server**

|  |  |  |
| --- | --- | --- |
| **Factor** | **Web Browser** | **Web Server** |
| Purpose | Web Browser is a software which is used to browse and display pages available over internet. | Web server is a software which provides these documents when requested by web browsers. |
| Function | A web browser sends request to server for web based documents and services. | Web server sees and approves those requests made by web browsers and sends the document in response |
| Process | Web browsers send HTTP Request and receive HTTP Response | Web servers receive HTTP Request and send HTTP Response. |
| Processing Model | Web browser has no processing model. | Web servers follow three major processing models: process based, thread based or hybrid. |
| Data Storage | Web browsers stores user data in cookies in local machine. | Web servers provide an area to store the website. |
| Installation | Web Browser is installed on user's machine. | Web servers can be installed anywhere but it need to be on a network or on local computer. |
| Acts as | Web browser acts as an interface between the web server and client. | Web servers act as the sender of web resources like web pages. |
| Responsibility | Web browser is responsible to request for a website or webpage located on the internet. | Web server is responsible for hosting websites, processing web requests, and sending the demanded document to the client. |
| Example | Examples of web browsers include Google Chrome, Internet explorer, Mozilla Firefox, etc. | Apache Server is an example of a web server. |

**WEB APPLICATION:**

1. 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.
2. Web services are Web apps by definition and many, although not all, websites contain Web apps.

**URLs (Uniform Resource Locators:**

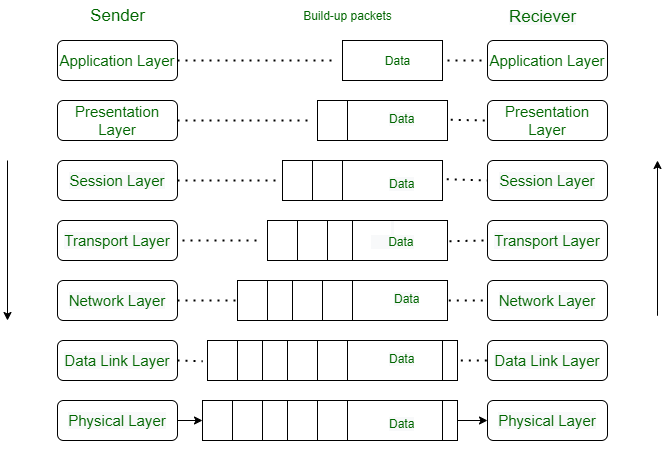
* URLs are addresses that identify specific resources on the internet, such as web pages, images, videos, or files.
* A URL typically consists of several components:
  + **Protocol**: It specifies the protocol to be used for communication, such as HTTP (Hypertext Transfer Protocol) or HTTPS (HTTP Secure).
  + **Domain Name**: It represents the unique address of a website, often preceded by "www" (e.g., [www.example.com](http://www.example.com/)).
  + **Path**: It indicates the specific location of a resource within the website's directory structure (e.g., /products/index.html).
  + **Query Parameters:** These optional parameters follow a question mark and provide additional information or instructions to the server (e.g., ?category=books&sort=price).
* Example: <https://www.example.com/products/index.html?category=books&sort=price>

**Navigation:**

* Navigation refers to the process of moving between web pages or resources within a website or across different websites.
* Web navigation can be accomplished through various actions performed within a web browser:
  + **Typing or Pasting URL:** Users can enter a URL directly into the address bar of a web browser to access a specific web page.
  + **Clicking Links**: Websites often contain hyperlinks that users can click on to navigate to different pages within the same website or external websites.
  + **Bookmarks/Favorites**: Users can save specific URLs as bookmarks or favorites within their web browser for quick access to frequently visited websites.
  + **Back and Forward Buttons**: Web browsers provide navigation buttons that allow users to move back and forth between previously visited pages.
  + **Search Engines**: Users can enter search queries in a search engine's search bar to find relevant websites or specific web pages.

**OSI MODEL:**

1. OSI stands for **Open Systems Interconnection**.
2. It has been developed by ISO – ‘**International Organization for Standardization**‘, in the year 1984.
3. It is a 7-layer architecture with each layer having specific functionality to perform.
4. All these 7 layers work collaboratively to transmit the data from one person to another across the globe.

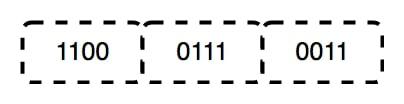


**Layers of OSI Model**

1. [Physical Layer](https://www.geeksforgeeks.org/physical-layer-in-osi-model/)
2. [Data Link Layer](https://www.geeksforgeeks.org/data-link-layer/)
3. [Network Layer](https://www.geeksforgeeks.org/network-layer-services-packetizing-routing-and-forwarding/)
4. [Transport Layer](https://www.geeksforgeeks.org/transport-layer-responsibilities/)
5. [Session Layer](https://www.geeksforgeeks.org/session-layer-in-osi-model/)
6. [Presentation Layer](https://www.geeksforgeeks.org/presentation-layer-in-osi-model/)
7. [Application Layer](https://www.geeksforgeeks.org/application-layer-in-osi-model/)

**Layer 1- Physical Layer**

1. The lowest layer of the OSI reference model is the physical layer.
2. It is responsible for the actual physical connection between the devices.
3. The physical layer contains information in the form of**bits.**
4. It is responsible for transmitting individual bits from one node to the next.
5. When receiving data, this layer will get the signal received and convert it into 0s and 1s and send them to the Data Link layer, which will put the frame back together.



**Devices used in Layer1: Hub, Repeater, Modem, and Cables are Physical Layer devices.**

**Layer 2- Data Link Layer (DLL)**

The data link layer is responsible for the node-to-node delivery of the message.

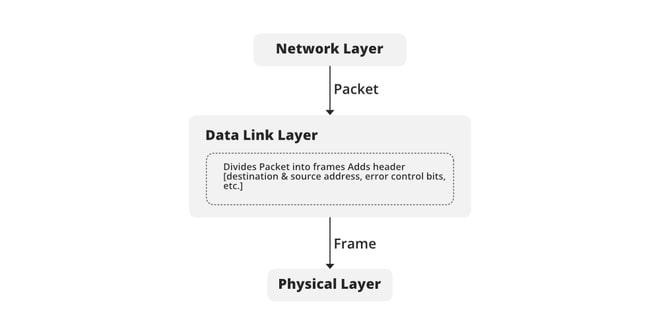
The main function of this layer is to make sure data transfer is error-free from one node to another, over the physical layer.

When a packet arrives in a network, it is the responsibility of the DLL to transmit it to the Host using its MAC address.

The packet received from the Network layer is further divided into frames.

DLL also encapsulates Sender and Receiver’s MAC address in the header.

The Receiver’s MAC address is obtained by placing an [ARP(Address Resolution Protocol)](https://www.geeksforgeeks.org/how-address-resolution-protocol-arp-works/)request onto the wire asking “Who has that IP address?” and the destination host will reply with its MAC address.



           Devices: **Switch** & **Bridge** are Data Link Layer devices.

**Layer 3- Network Layer**

The network layer works for the transmission of data from one host to the other located in different networks.

It also takes care of packet routing i.e. selection of the shortest path to transmit the packet, from the number of routes available.

The sender & receiver’s IP addresses are placed in the header by the network layer.

Network layer is implemented by networking devices such as **routers**.

**Layer 4- Transport Layer**

The data in the transport layer is referred to as *Segments*.

It is responsible for the End to End Delivery of the complete message.

The transport layer also provides the acknowledgment of the successful data transmission and re-transmits the data if an error is found.

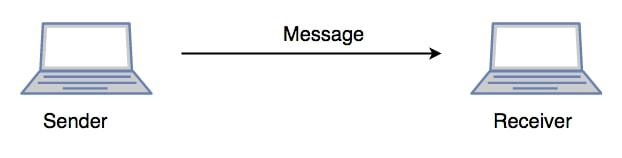
  The transport layer is called as **Heart of the OSI** model.

**Layer 5- Session Layer**

This layer is responsible for the establishment of connection, maintenance of sessions, and authentication, and also ensures security.

**Scenario**

Let us consider a scenario where a user wants to send a message through some Messenger application running in his browser. The “Messenger” here acts as the application layer which provides the user with an interface to create the data. This message or so-called Data is compressed, encrypted (if any secure data), and converted into bits (0’s and 1’s) so that it can be transmitted.



**Layer 6- Presentation Layer**

The presentation layer is also called the **Translation layer**.

The data from the application layer is extracted here and manipulated as per the required format to transmit over the network.

In this layer, the data is encrypted**.** Data encryption translates the data into another form or code. The encrypted data is known as the ciphertext and the decrypted data is known as plain text.

**Layer 7- Application Layer**

At the very top of the OSI Reference Model stack of layers, we find the Application layer which is implemented by the network applications.

These applications produce the data, which has to be transferred over the network.

This layer also serves as a window for the application services to access the network and for displaying the received information to the user.

Example: Application – Browsers, Skype Messenger, etc.

The application Layer is also called **Desktop** **Layer**.

**Example of OSI Model (Just For Understanding):**

Certainly! Let's consider a scenario of Alice and Bob, who want to communicate using their computers over a network. Each layer of the OSI model can be illustrated through their interaction:

1. Physical Layer:
   * Alice wants to send a message to Bob. She types the message on her computer and presses the "Send" button.
   * The message is converted into electrical signals by the network interface card (NIC) in Alice's computer.
2. Data Link Layer:
   * The Data Link layer adds a header to the message, including source and destination MAC addresses, to create a data frame.
   * The frame also contains error detection information, such as a checksum, to ensure data integrity.
   * The NIC in Alice's computer sends the data frame over the network cable to the switch.
3. Network Layer:
   * The Network layer adds an IP (Internet Protocol) header to the data frame, including the source and destination IP addresses.
   * The IP address uniquely identifies Alice's and Bob's computers on the network.
   * The data frame is then forwarded to the appropriate router based on the destination IP address.
4. Transport Layer:
   * The Transport layer segments the data frame into smaller packets and adds a transport layer header, such as a TCP (Transmission Control Protocol) header.
   * The TCP header includes source and destination port numbers, ensuring that the message reaches the correct application on Bob's computer.
5. Session Layer:
   * The Session layer establishes and manages a session between Alice and Bob's computers.
   * It coordinates the communication and ensures that both parties are ready to receive and send data.
6. Presentation Layer:
   * The Presentation layer is responsible for data formatting and encryption, ensuring that the message is in a suitable format for transmission.
   * It may compress the message and encrypt it to maintain confidentiality and security during transmission.
7. Application Layer:
   * The Application layer is where the actual application or program resides. It allows Alice to compose the message and Bob to receive and read it.
   * Examples of application layer protocols include email (SMTP), file transfer (FTP), and web browsing (HTTP).

Once the message reaches Bob's computer, the process reverses as the layers work together to interpret and deliver the message to Bob's application for him to read it.

This example/story covers all seven layers of the OSI model, demonstrating how each layer plays a distinct role in the communication process between Alice and Bob over the network.

# **TCP/IP model:**

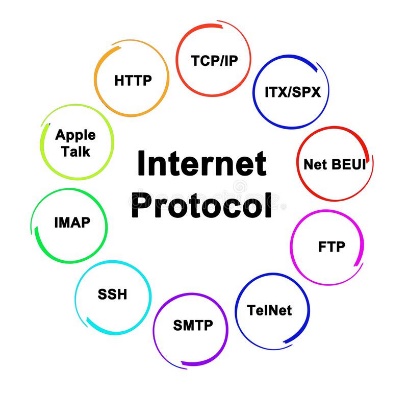
1. **TCP/IP** was designed and developed by the Department of Defense (DoD) in the 1960s and is based on standard protocols.
2. It stands for Transmission Control Protocol/Internet Protocol.
3. The [TCP/IP model](https://www.geeksforgeeks.org/tcp-ip-model/) is a concise version of the OSI model.
4. The TCP/IP model consists of five layers: the application layer, transport layer, network layer, data link layer and physical layer.
5. TCP/IP is a hierarchical protocol made up of interactive modules, and each of them provides specific functionality. Here, hierarchical means that each upper-layer protocol is supported by two or more lower-level protocols.

## Functions of TCP/IP layers:



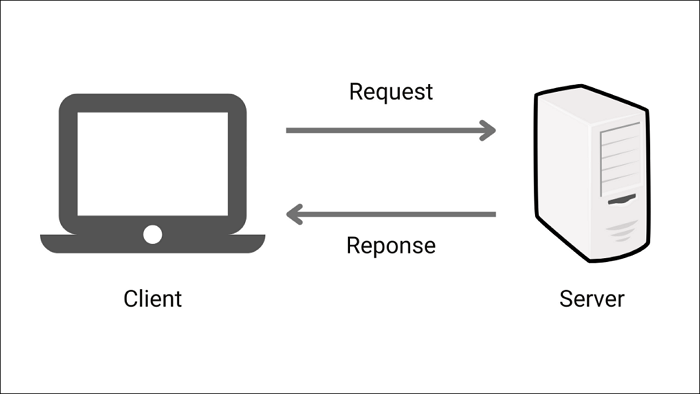
**Internet Protocol (IP):** The IP protocol provides the addressing scheme and routing mechanism that enables data to be sent and received across the Internet.

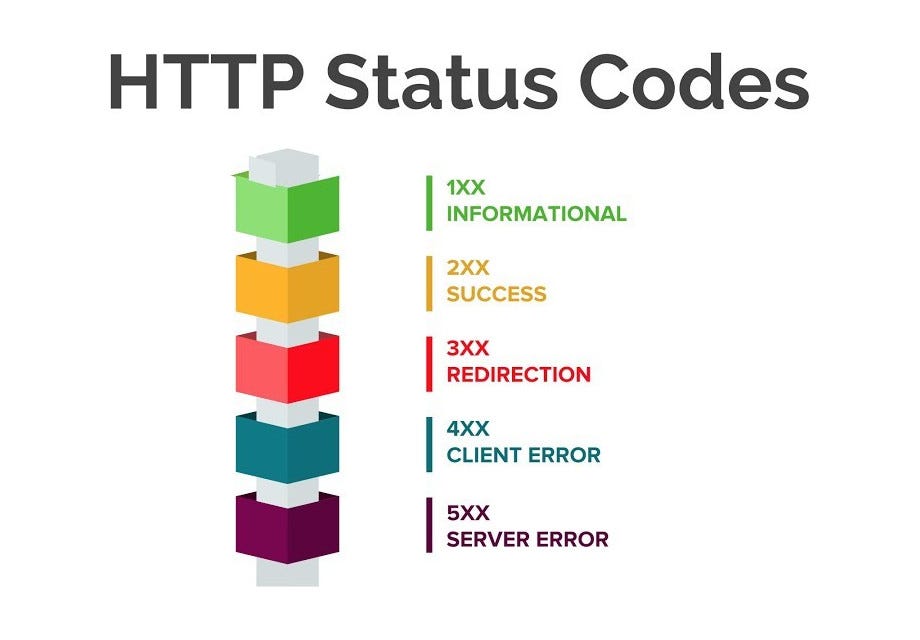
1. **Transmission Control Protocol (TCP):** TCP ensures reliable delivery of data by dividing it into packets, numbering them, and reassembling them at the receiving end.
2. **Hypertext Transfer Protocol (HTTP):** HTTP is the protocol used for communication between web browsers and web servers, enabling the retrieval and display of web pages.
3. **File Transfer Protocol (FTP):** FTP is used for transferring files between a client and a server over a network, allowing efficient file sharing and management.
4. **Simple Mail Transfer Protocol (SMTP):** SMTP is responsible for sending and delivering email messages between mail servers, ensuring proper routing and delivery.
5. **Domain Name System (DNS):** DNS translates domain names (e.g., example.com) into IP addresses, enabling users to access websites using human-readable names.
6. **Secure Shell (SSH):** SSH provides secure remote access to networked devices or servers, allowing encrypted and authenticated communication.
7. **Internet Message Access Protocol (IMAP):** IMAP enables email clients to access and manage email messages stored on a remote mail server, providing advanced email features.
8. **Post Office Protocol (POP):** POP retrieves email from a remote mail server and delivers it to a client device, allowing offline access to email messages.
9. **Simple Network Management Protocol (SNMP)**: SNMP facilitates network management by allowing devices to be monitored and managed centrally, collecting information and performing configuration tasks.



**CLIENT REQUEST AND SERVER RESPONSE**

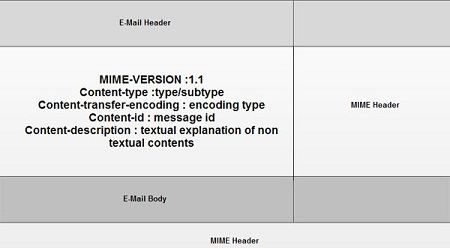
1. When a request is sent by the client, a response is generated by the server.
2. This response contains a status line that shows HTTP version, status code and a message.
3. This message describes the status code generated by the server.
4. 200 is the default status code that tells that the request is OK.





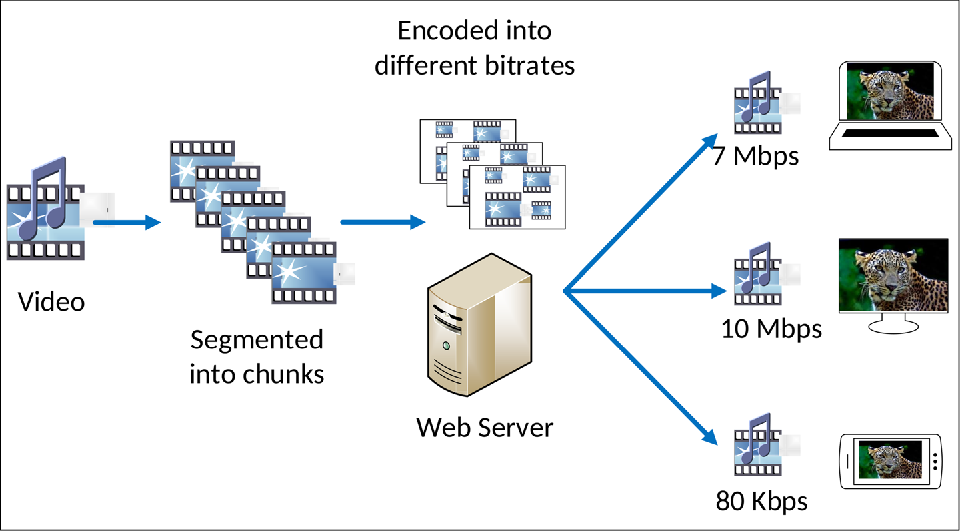
**MIME:**

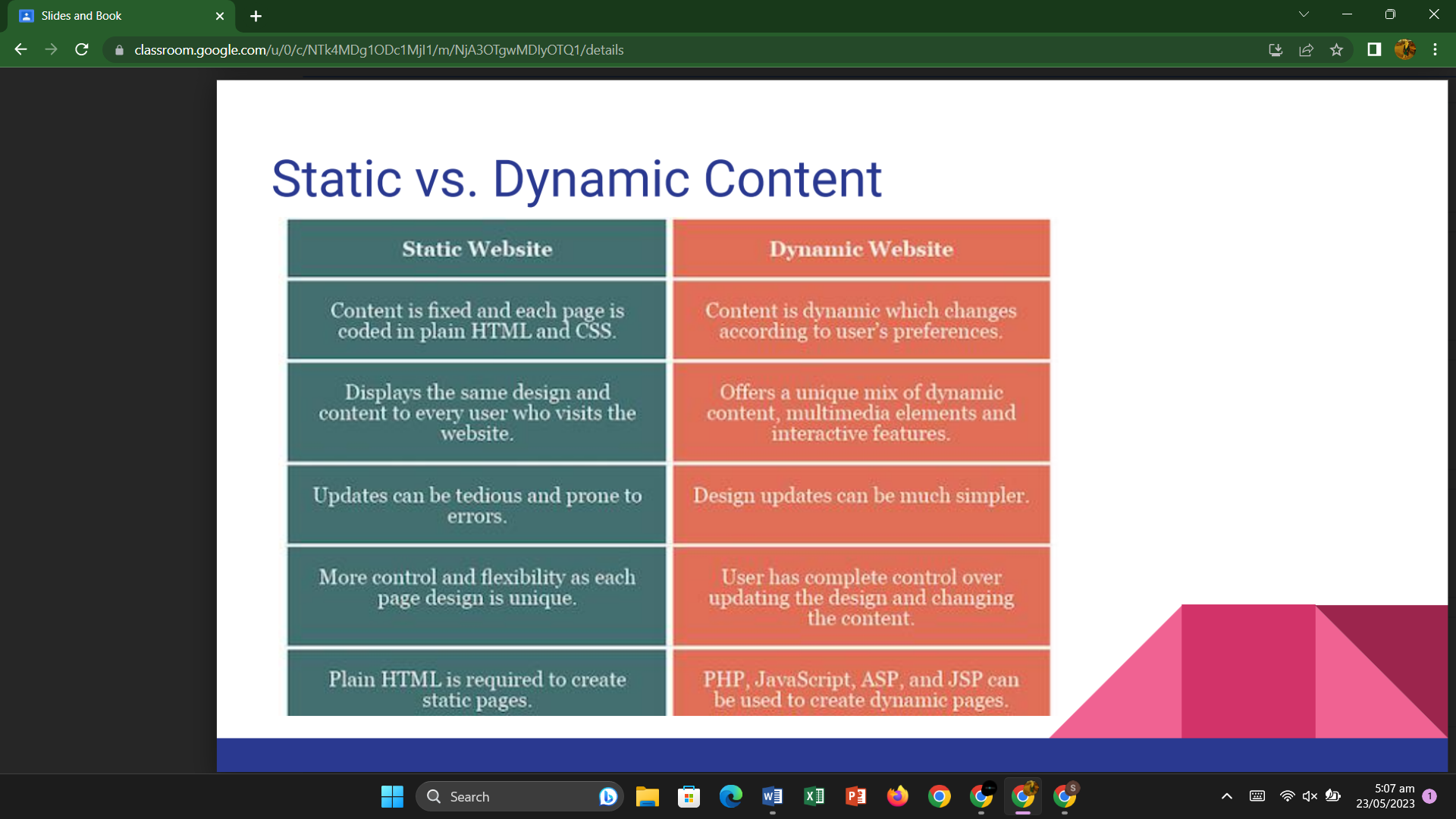
1. MIME stands for Multipurpose Internet Mail Extensions.
2. It is an Internet standard that extends the capabilities of email messages to support different types of files beyond plain text
3. . In simple terms, MIME allows you to attach and send various types of files, such as images, videos, documents, or audio, within an email.
4. Before MIME, email messages could only contain plain text. With MIME, you can now include attachments and send multimedia content through emails.



**DASH:**

1. Dynamic Adaptive Streaming over HTTP (DASH), also known as MPEG-DASH, is a technology that allows smooth and high-quality streaming of media content, like videos, over the internet.
2. It works by adapting the quality of the streaming video in real-time based on the viewer's internet connection and device capabilities.
3. Imagine you are watching a video on your computer or smartphone. With DASH, the video is divided into small chunks or segments.
4. Each segment is available in multiple versions or bitrates, with varying levels of quality.
5. The DASH player on your device continuously monitors your internet connection speed and adjusts the video quality accordingly.

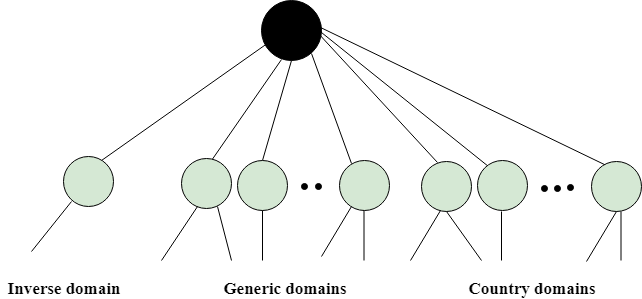




**DNS:**

* DNS stands for Domain Name System.
* DNS is a directory service that provides a mapping between the name of a host on the network and its numerical address.
* DNS is required for the functioning of the internet.
* Each node in a tree has a domain name, and a full domain name is a sequence of symbols specified by dots.
* DNS is a service that translates the domain name into IP addresses. This allows the users of networks to utilize user-friendly names when looking for other hosts instead of remembering the IP addresses.
* For example, suppose the FTP site at EduSoft had an IP address of 132.147.165.50, most people would reach this site by specifying ftp.EduSoft.com. Therefore, the domain name is more reliable than IP address.

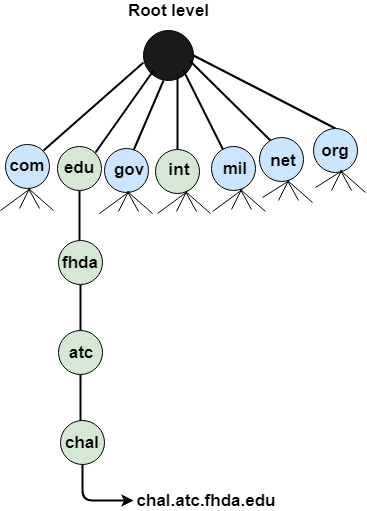
DNS is a TCP/IP protocol used on different platforms. The domain name space is divided into three different sections: generic domains, country domains, and inverse domain.



**Generic Domains**

* It defines the registered hosts according to their generic behavior.
* Each node in a tree defines the domain name, which is an index to the DNS database.
* It uses three-character labels, and these labels describe the organization type.

|  |  |
| --- | --- |
| **Label** | **Description** |
| com | Commercial Organizations |
| coop | Cooperative business Organizations |
| edu | Educational institutions |
| gov | Government institutions |
| info | Information service providers |
| net | Network Support centers |
| org | Nonprofit Organizations |



**Country Domain**

The format of country domain is same as a generic domain, but it uses two-character country abbreviations (e.g., us for the United States) in place of three character organizational abbreviations.

**Inverse Domain**

The inverse domain is used for mapping an address to a name. When the server has received a request from the client, and the server contains the files of only authorized clients. To determine whether the client is on the authorized list or not, it sends a query to the DNS server and ask for mapping an address to the name.

**Working of DNS**

* DNS is a client/server network communication protocol. DNS clients send requests to the. server while DNS servers send responses to the client.
* Client requests contain a name which is converted into an IP address known as a forward DNS lookups while requests containing an IP address which is converted into a name known as reverse DNS lookups.
* DNS implements a distributed database to store the name of all the hosts available on the internet.
* If a client like a web browser sends a request containing a hostname, then a piece of software such as **DNS resolver** sends a request to the DNS server to obtain the IP address of a hostname. If DNS server does not contain the IP address associated with a hostname, then it forwards the request to another DNS server. If IP address has arrived at the resolver, which in turn completes the request over the internet protocol.

## Web Architecture:

## • Web architecture refers to the overall structure of a website or web application, including the way it is designed, implemented, and deployed.

## • It involves the use of technologies and protocols such as HTML, CSS, JavaScript, and HTTP to build and deliver web pages and applications to users.

## • Web architecture also includes the design and layout of the website or web application, as well as the way it is organized and the relationships between different pages and components.

## • It also includes the way the website or web application is built and maintained, including the use of frameworks and libraries, and the deployment and hosting of the website or web application.

## Client-server architecture:

## is a network architecture in which each computer or process on the network is either a client or a server. Clients request for an application and the application server responds according to the request.

## 1-Tier architecture: 1-Tier Architecture is the simplest, single tier on single user, and is the equivalent of running an application on a personal computer. All the required component to run the application are located within it. User interface, business logic, and data storage are all located on the same machine.

## 1 Tier Architecture in DBMS - javatpoint

## 2-Tier architecture:

## A two-tier architecture is a software architecture in which a presentation layer or interface runs on a client, and a data layer or data structure gets stored on a server.

## Software Architecture and its types - 1-tier 2 tier 3 tier N-tier Architecture |

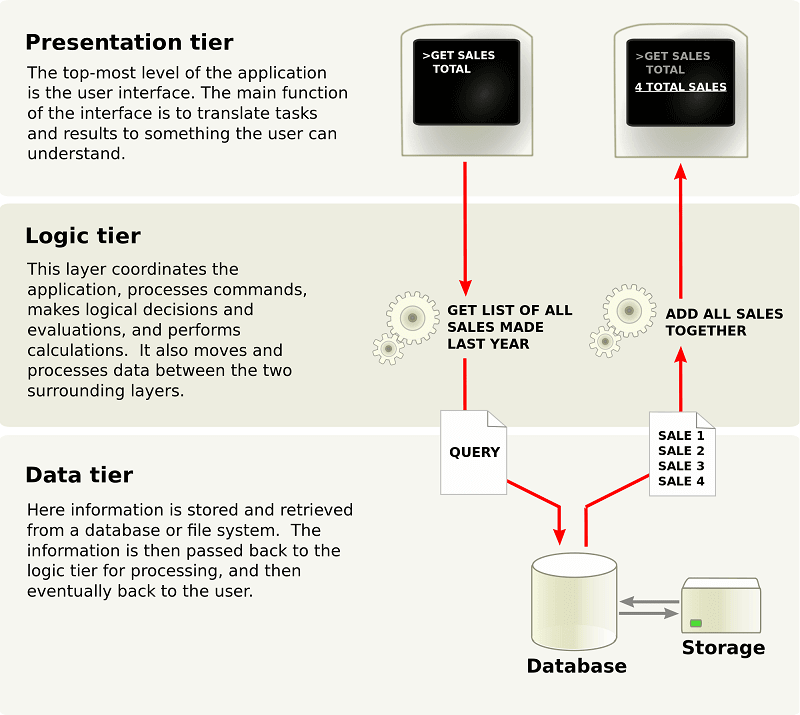
## Definition of N-Tier Architecture

1. N-tier architecture is also called multi-tier architecture because the software is engineered to have the processing, data management, and presentation functions physically and logically separated.
2. That means that these different functions are hosted on several machines or clusters, ensuring that services are provided without resources being shared and, as such, these services are delivered at top capacity.
3. The “N” in the name n-tier architecture refers to any number from 1.
4. Not only does your software gain from being able to get services at the best possible rate, but it’s also easier to manage.
5. This is because when you work on one section, the changes you make will not affect the other functions.
6. And if there is a problem, you can easily pinpoint where it originates.

### **A More In-Depth Look at N-Tier Architecture**

N-tier architecture would involve dividing an application into [three different tiers](https://msdn.microsoft.com/en-us/library/bb384398.aspx).  These would be the

1. logic tier,
2. the presentation tier, and
3. the data tier.



1. **Presentation Tier (Client Tier):**
   * This is the topmost layer that interacts directly with the end-user or client.
   * It handles the user interface, input validation, and presentation logic.
   * It can be a web browser, desktop application, or mobile app.
   * Web browser (Google Chrome, Mozilla Firefox)
   * Mobile app (iOS/Android)
2. **Application Tier (Logic Tier):**
   * The application tier contains the core business logic and processing rules.
   * It handles the request processing, business workflows, and data manipulation.
   * It can also implement security, caching, and session management.
   * Web server (Apache, Nginx)
   * Application server (Node.js, Java EE, ASP.NET)
   * Business logic components (Java classes, C# libraries)
   * Security modules (authentication, authorization)
3. **Data Tier (Persistence Tier):**
   * The data tier is responsible for data storage and retrieval.
   * It includes databases, file systems, and other data repositories.
   * It handles data access, data validation, and data integrity.
   * Relational Database Management System (RDBMS) such as MySQL, PostgreSQL, or Oracle
   * Object-Relational Mapping (ORM) framework (Hibernate, Entity Framework)
   * Data access components (Java Database Connectivity - JDBC, Object-Relational Mapping - ORM)

**Advantages of N-tier architecture:**

1. **Scalability**: N-tier architecture allows scaling individual tiers independently. For example, you can add more application servers to handle increased processing load without affecting the presentation or data tiers.
2. **Reusability**: With clear boundaries between layers, components and services can be reused across different applications or projects, leading to faster development and improved productivity.
3. **Flexibility**: N-tier architecture provides flexibility in technology selection for each tier. Different tiers can use different technologies or programming languages that best suit their specific requirements.
4. **Security**: The layered architecture helps enforce security measures at different tiers. For example, the application tier can handle authentication and authorization, while the data tier can control access to sensitive data.
5. **Fault Isolation:** Issues or failures in one tier generally do not impact the other tiers. This isolation helps in identifying and resolving problems more efficiently and ensures high availability of the overall system.

**Example:**

1. For instance, consider an e-commerce application based on this architecture.
2. The web browser or mobile app allows users to browse products, add items to a shopping cart, and place orders.
3. The web application framework handles these requests, validates user input, and communicates with the database to store and retrieve product information, user data, and order details.
4. The business logic components in the application tier manage the ordering process, handle payment integration, and enforce authentication and authorization rules.

**MIDDLEWARE:**

1. In the context of web development, middlewares are components or functions that sit between the web server and the application.
2. They intercept and process incoming requests and outgoing responses, adding extra functionalities or performing certain operations before passing them to the next step in the request/response cycle.
3. In easy words, middlewares act as a bridge between the web server and the application, allowing developers to modify or enhance the request/response flow.

**Example:**

Let's say you have a web application built with a framework like Express.js (a popular web framework for Node.js). In this application, you want to add a middleware that logs information about each incoming request to the console.

You would define a middleware function that takes three parameters: **request**, **response**, and **next**. The **next** parameter is a function that indicates when the middleware has finished its work and allows the request to move forward to the next middleware or the final request handler.

**Usage:**

1. Middlewares are commonly used for various purposes in web applications, such as authentication, authorization, request parsing, error handling, and more.
2. They provide a way to inject custom logic into the request/response cycle, allowing developers to extend and customize the behavior of their applications.