

Web Application Development and Software Frameworks (CSEN3071)

Course Educational Objectives:

- ✓ Design static web page using Markup languages.
- ✓ Design and implement web pages using style sheets.
- ✓ Implement with java script web applications with dynamic web pages.
- ✓ Understand working of Web servers
- ✓ Develop web applications using frameworks.

Course Outcomes:

After successful completion of the course the student will be able to:

1. Understand the fundamentals of web application development and frameworks.
2. Design interactive web pages with client and server side scripting.
3. Apply validations on user input using Java script.
4. Compare and analyze XML and JSON documents.
5. Create and deploy Web Applications over web server.

UNIT-I:

Introduction to Web Application Designing

Chapter-I

Introduction:

- ❖ Building a Web Application
- ❖ Components – Client Side and Server-side
- ❖ 2 tier
- ❖ n-tier architectures
- ❖ Networks
- ❖ Protocols
- ❖ MVC Pattern

Reference:

- **Programming the World Wide Web**, 7th Edition, Robert W Sebesta, Pearson, 2013.
- Dietel and Nieto, **Internet and World Wide Web - How to program**, PHI/Pearson Education, 2006.

A Brief Introduction to the Internet

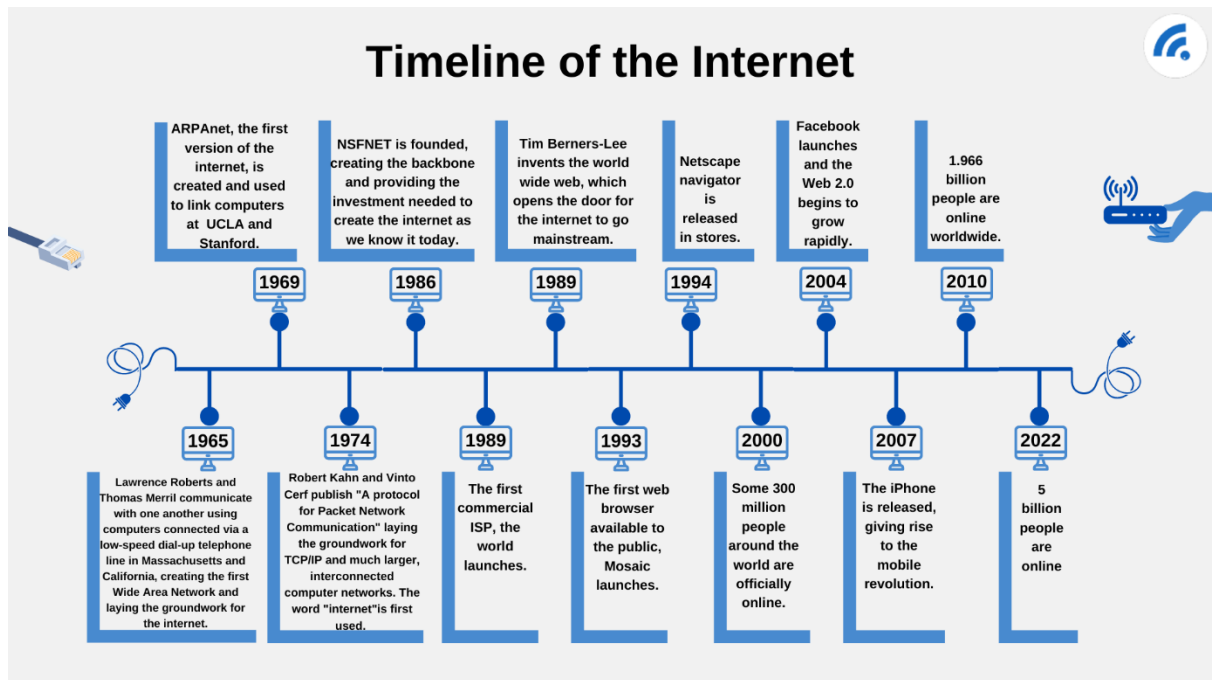
- ✓ In the 1960s, the U.S. Department of Defense (DoD) became interested in developing a new large-scale computer network.

Purposes:

- Communications
- Program sharing and
- Remote Computer Access for researchers working on defense-related contracts.

Fundamental Requirement:

- ✓ Network be sufficiently robust so that even if some network nodes were lost to sabotage, war, or some more benign cause, the network would continue to function.
- ✓ The DoD's Advanced Research Projects Agency (**ARPA**) funded the construction of the first such network, which **connected** about a **dozen** ARPA funded research laboratories and universities.
- ✓ The **first node** of this network was established at **UCLA** in **1969**.
- ✓ Because it was **funded** by **ARPA**, the network was named **ARPAnet**.
- ✓ Primary early **use** -> ARPAnet -> simple **text-based** communications through electronic mail.
- ✓ ARPAnet was **available** only to **laboratories** and **universities** but great majority of **educational institutions** were not connected.
- ✓ Other **networks** were developed during the late **1970s** and early **1980s**, with **BITNET** and **CSNET** among them.
- ✓ **Because It's Time Network** (BITNET), began at the City University of New York. It was built initially to provide **electronic mail** and **file transfers**.
- ✓ CSNET (**Computer Science Network**) connected to different universities and initial **purpose** was to provide **electronic mail**. For a variety of reasons, **neither** BITNET **nor** CSNET became a widely used national network.
- ✓ A new national network, **NSFnet**, was created in **1986** -> **sponsored** by the National Science Foundation (**NSF**).
- ✓ NSFnet **initially** connected the NSF-funded **supercomputer centers** that were at **five universities**. Soon **after** -> available to other academic institutions and research laboratories.
- ✓ By **1990**, NSFnet had replaced **ARPAnet** for most nonmilitary uses.
- ✓ A wide **variety** of organizations had established nodes on the **new** network— by **1992**.
- ✓ NSFnet connected more than **1 million** computers around the world.
- ✓ In **1995**, a small part of NSFnet returned to being a **research network**.
- ✓ The rest became known as the **Internet**, although this term was used much earlier for both ARPAnet and NSFnet.



❖ What Is the Internet?

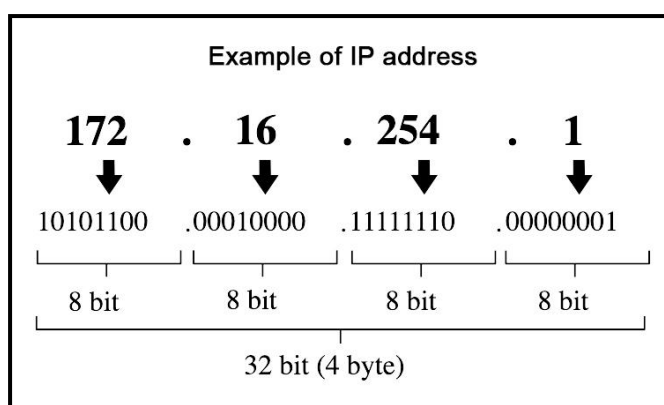
- ✓ The Internet is a **huge collection of computers connected in a communications network**. These computers are of every **imaginable size, configuration, and manufacturer**.
- ✓ some of the devices connected to the Internet—such as **plotters and printers**—are not computers at all.
- ✓ The **innovation** that allows all of these diverse devices to communicate with each other is a **single, low-level protocol** named Transmission Control Protocol/Internet Protocol (**TCP/IP**).
- ✓ Internet is actually a **network of networks**, rather than a network of computers.

Example. Local network -> local network is physically connected to the Internet.

❖ Internet Protocol Addresses

- ✓ For **people**, Internet nodes are identified by **names**; for **computers**, they are identified by numeric **addresses**.

Example: Variable name in a program versus address in memory

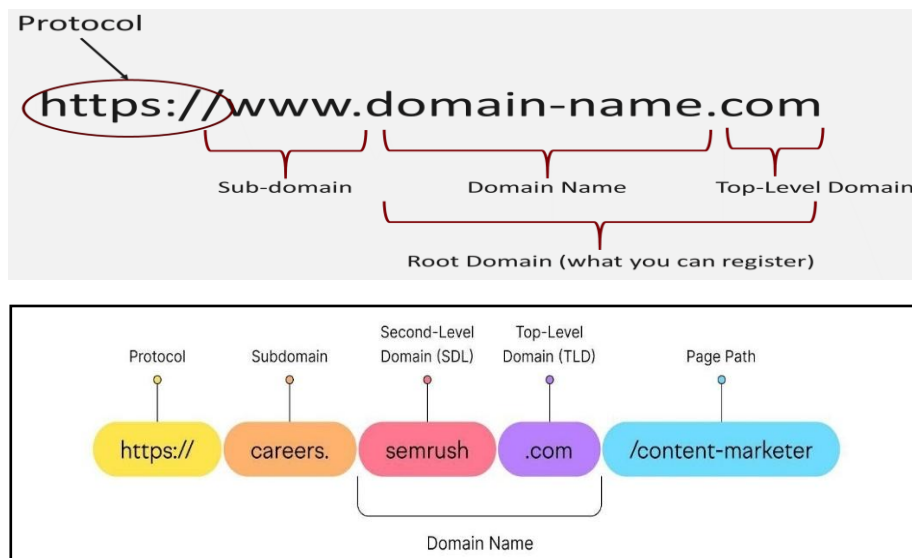


For example:

- A **small** organization may be assigned 256 IP addresses, such as **191.57.126.0** to **191.57.126.255**.
- **Very large** organizations, such as the Department of Defense, may be assigned **16 million** IP addresses, which include IP addresses with one particular first **8-bit** number, such as **12.0.0.0** to **12.255.255.255**.

In late **1998**, a new IP standard, **IPv6**, was approved, although it still is not widely used. The most significant change was to expand the address size from **32 bits** to **128 bits**.

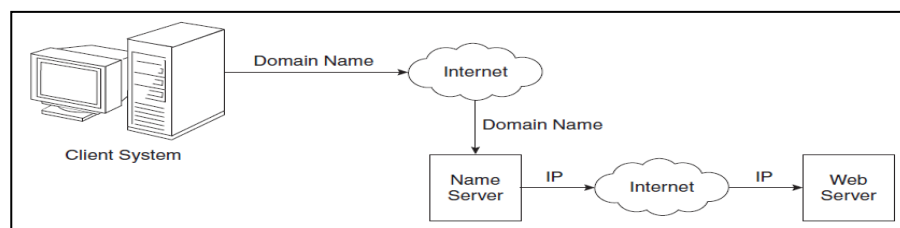
❖ Domain Names



Consider this sample address:

`movies.marxbros.comedy.com`

- Here, **movies** is the **hostname** and **marxbros** is **movies's local domain**, which is a part of **comedy's domain**, which is a part of the **com domain**.
- The hostname and all of the domain names are together called a **fully qualified domain name**.
- The **fully qualified domain name** of the destination for a message, which is what is given by a **browser user**, must be **converted** to an **IP address** before the message can be transmitted over the Internet to the destination.
- These conversions are done by software systems called **name servers**, which implement the Domain Name System (**DNS**).



❖ The World Wide Web

- ❖ In **1989**, a small group of people led by **Tim Berners-Lee** at **CERN** (European Organization for Nuclear Research) proposed a new protocol for the Internet - **World Wide Web**:

- To allow scientists around the world to use the Internet to **exchange documents** describing their work.
- To **search** for and **retrieve** documents from databases on any number of different document-serving computers connected to the Internet.

- ❖ By late **1990**, the basic ideas for the new system had been fully developed and implemented on a **NeXT** computer at CERN.

- ❖ In **1991**, the system was **ported** to other computer platforms and **released** to the rest of the world.

- **Web or Internet?**

- ❖ **“The Internet”**: It is collection of information stored on the networked computers over the world.

- ❖ **“Web”**: It is collection of documents stored that are interconnected by hyper-links. These documents accessed by web browsers and provided by web servers.

❖ What is a web page?

- Web pages are text files containing HTML
- HTML – Hyper Text Markup Language
- A notation for describing
- document structure (semantic markup)
- formatting (presentation markup)
- Looks (looked?) like:
- A Microsoft Word document
- The markup tags provide information about the page content structure

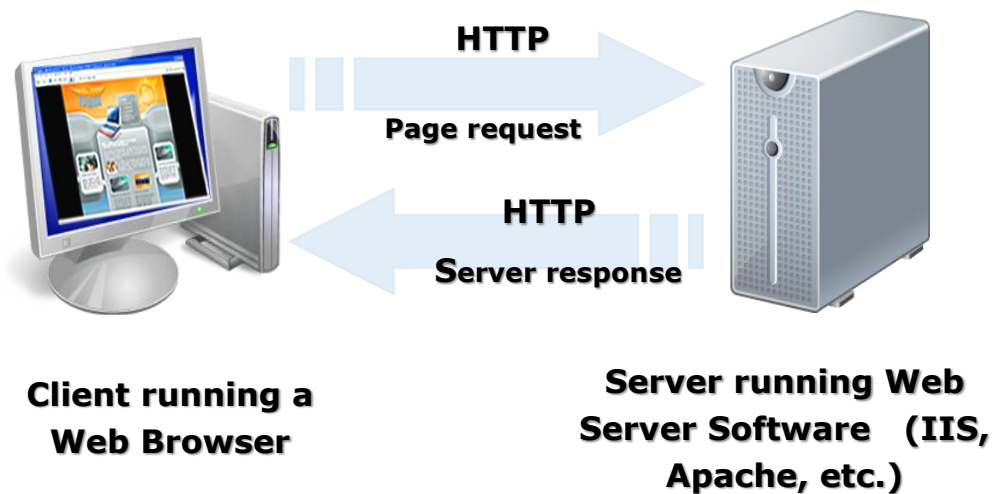
❖ other terms

- HTML – **HyperText Markup Language** – The Language of Web Pages on the World Wide Web.
- HTML is a text formatting language.
- **URL** – Uniform Resource Locator.
- **Browser** – A software program which is used to show web pages.
- Hypertext Markup Language
- **Hypertext**:
- Allows for non-linear linking to other documents

- **Markup** Language:
- Content to be displayed is “marked up” or tagged to tell the browser how to display it.
- In 1980’s, **HyperCard** information management system from Apple.

❖ How the Web Works?

- WWW use classical client / server architecture
- HTTP is text-based request-response protocol



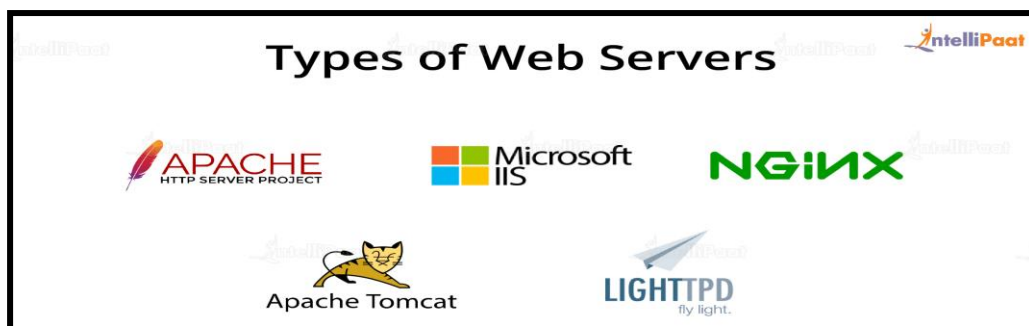
❖ Web browsers:

- They are called browsers because they **allow** the **user** to **browse** the **resources** available on servers.
- The first browsers were **text based**—they were not capable of displaying **graphic** information, nor did they have a **graphical user interface**.
- In early **1993**, things changed with the release of **Mosaic**, the **first browser** with a **graphical** user interface. Mosaic was developed at the National Center for Supercomputer Applications (**NCSA**) at the University of Illinois.
- Although the Web supports a variety of **protocols**, the most common one is the Hypertext Transfer Protocol (**HTTP**).
- HTTP provides a standard form of **communication** between **browsers** and **Web servers**.



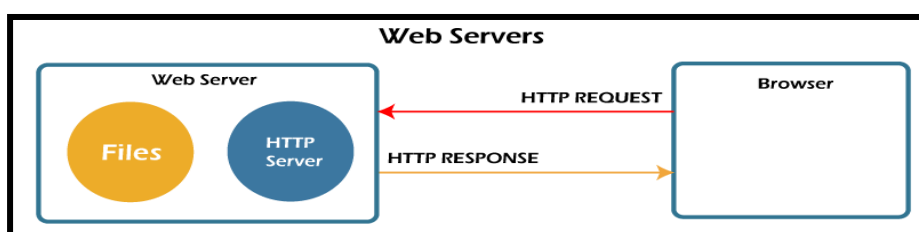
❖ Web Servers

- ✓ Web servers are programs that **provide documents** to requesting **browsers**.
- ✓ Servers are **slave** programs: They **act** only when **requests** are **made** to them by browsers running on other computers on the Internet.



❖ Web Servers

- ✓ Web browsers initiate network communications with servers by sending them **URLs**.
- ✓ A **URL** can specify one of **two** different things:
 - The **address** of a data **file** stored on the server that is to be sent to the client.
 - A **program** stored on the server that the client wants **executed** and the output of the program **returned** to the client.
- ✓ All the communications between a Web client and a Web server use the standard Web protocol, Hypertext Transfer Protocol (HTTP),
- ✓ When a Web server **begins** execution, it **informs** the **operating system** under which it is running that it is now ready to **accept** incoming network connections through a specific **port** on the machine.



❖ General Server Characteristics

- ✓ The **file structure** of a Web server has **two** separate **directories**.

1. The root of one of these is called the **document root**.

- The file hierarchy that **grows** from the document root stores the **Web documents** to which the **server** has **direct access** and normally serves to clients.

2. The root of the other directory is called the **server root**.

- This directory, along with its **descendant directories**, stores the server and its support software.
- The **files stored** directly in the document root are those available to clients through **top-level URLs**.
- **Example:** Suppose further that the document root is named topdocs and is stored in the /admin/web directory, making its address /admin/web/topdocs.
- A request for a file from a client with the URL <http://www.tunias.com/petunias.html> will cause the server to search for the file with the file path **/admin/web/topdocs/petunias.html**.
- Similarly, URL <http://www.tunias.com/bulbs/tulips.html> will cause the server to search for the file with the address **/admin/web/topdocs/bulbs/tulips.html**.
- Many servers allow part of the **servable document** collection to be stored **outside** the **directory** at the document root. The secondary areas from which documents can be served are called **virtual document trees**.
- Many servers can support **more than one site** on a computer, potentially reducing the cost of each site and making their maintenance more convenient. Such secondary hosts are called **virtual hosts**.
- Some servers can **serve** documents that are in the document root of **other machines** on the Web; in this case, they are called **proxy servers**.

Examples:

1. **Apache,**
2. **IIS**

- ✓ The **Microsoft IIS** server is supplied as part of Windows—and because it is a reasonably good server—most **Windows based** Web servers use IIS.
- ✓ Apache and IIS provide **similar varieties** of services.
- ✓ From the point of view of the **site manager**: **difference** between Apache and IIS is that:
- ✓ **Apache** is controlled by a **configuration file** that is edited by the manager to change Apache's behavior.

- ✓ With **IIS**, server behavior is modified by changes made through a window-based management **program**, named the IIS **snap-in**, which controls both IIS and ftp. This program allows the site manager to set parameters for the server.

❖ Uniform Resource Locators

- ✓ Uniform (or universal) resource locators (**URLs**) are used to **identify** documents (resources) on the Internet.
- ✓ There are many different **kinds of resources**, identified by different **forms** of URLs.
- ✓ **URL Formats**

- All URLs have the same general format:

scheme:object-address

- The scheme is often a communications protocol. Common schemes include:

http, ftp, gopher, telnet, file, mailto, and news.

- Different schemes use object addresses that have different forms.
- In the case of HTTP, the form of the object address of a URL is as follows:

//fully-qualified-domain-name/path-to-document

Another scheme of interest to us is file.

file://path-to-document

○

✓ **URL Paths**

- The **path** to the document for the **HTTP protocol** is similar to a path to a **file** or **directory** in the file system of an operating system and is given by a sequence of directory names and a file name, all **separated** by whatever **separator character** the operating system uses.
- For UNIX servers: **forward** slashes
- For Windows servers: **backward** slashes
- **Example:** <http://www.gumboco.com/files/f99/storefront.html>
- A path that includes all directories along the way is called a **complete path**.

✓ **Multipurpose Internet Mail Extensions**

- **Without** knowing the **form** of the document received from a Web server, the browser would **not** be **able** to **render** it.
- The **forms** of these documents are specified with Multipurpose Internet Mail Extensions (**MIME**).
- **Type Specifications**

- MIME was developed to **specify the format** of different kinds of documents to be sent via Internet mail.
- These documents could contain various kinds of text, video data, or sound data.
- ✓ **MIME specifications have the following form:**
 - **type/subtype**
 - The most common MIME **types** are text, image, and video.
 - The most common **text subtypes**: plain and html
 - **image** subtypes: gif and jpeg
 - **video** subtypes: mpeg and quicktime.
- ✓ **The Hypertext Transfer Protocol:**
 - **All** Web communications transactions use the same protocol: the Hypertext Transfer Protocol (**HTTP**).
 - The **current version** of HTTP is 1.1 (Versions: **1.1/2/3**), formally defined as **RFC 2616**, which was approved in June 1999.
 - RFC 2616 is **available** at the Web site for the World Wide Web Consortium (**W3C**), <http://www.w3.org>.
 - HTTP consists of **two phases**: the **request** and the **response**.
 - Each HTTP communication (request or response) between a browser and a Web server consists of **two parts**:
 - 1. a **header** and
 - 2. a **body**.
 - The **header** contains information about the communication.
 - The **body** contains the data of the communication if there is any.
 - The **general form** of an **HTTP request** is as follows:
 - 1. 1. HTTP method Domain part of the URL HTTP version
 - 2. 2. Header fields
 - 3. 3. Blank line
 - 4. 4. Message body
 - The following is an **example** of the first line of an HTTP request:
 - GET /storefront.html HTTP/1.1

Table-1: HTTP request methods

Method	Description
GET	Returns the contents of the specified document
HEAD	Returns the header information for the specified document
POST	Executes the specified document, using the enclosed data
PUT	Replaces the specified document with the enclosed data
DELETE	Deletes the specified document

There are **four categories** of header fields:

1. **General**: For general information, such as the date
 2. **Request**: Included in request headers
 3. **Response**: For response headers
 4. **Entity**: Used in both request and response headers
- One common request field is the **Accept field**, which specifies a preference of the browser for the MIME type of the requested document.
 - For **example**; we might have any of the following:
 - Accept: text/plain
 - Accept: text/html
 - Accept: image/gif

A wildcard character, the asterisk (*): Accept: text/*

○ **The general form of an HTTP response is as follows:**

1. Status line
 2. Response header fields
 3. Blank line
 4. Response body
- The status line includes the HTTP version used, a three-digit status code for the response, and a short **textual explanation** of the status code.

For example, most responses begin with the following:

HTTP/1.1 **200** OK

Table-2: First digits of HTTP status codes

First Digit	Category
1	Informational
2	Success
3	Redirection
4	Client error
5	Server error

❖ Security

Consider a **simple case** of transmitting a credit card number to a company from which a purchase is being made. The security issues for this transaction are as follows:

1. **Privacy**—it must not be possible for the credit card number to be stolen on its way to the company's server.
2. **Integrity**—it must not be possible for the credit card number to be modified on its way to the company's server.
3. **Authentication**—it must be possible for both the purchaser and the seller to be certain of each other's identity.
4. **Nonrepudiation**—it must be possible to prove legally that the message was actually sent and received.

Note:

- The basic tool to support privacy and integrity is encryption.

❖ Some common terms to remember

- ✓ . **Distributed Network:** It is a **network architecture**, where the components located at network computers coordinate and communicate their actions only by **passing messages**.
 - It is a collection of **multiple systems** situated at different nodes but **appears** to the user as a **single system**.
 - It provides a **single data communication network** which can be managed separately by different networks.
 - An **example** of Distributed Network— where different **clients** are connected within **LAN** architecture on one side and on the **other side** they are connected to high-speed **switches** along with a rack of servers containing service nodes.
- ✓ **Client-Server Architecture:** It is an **architecture model** where the client (one program) requests a service from a server (another program) **i.e.** It is a request-response service provided over the internet or through an intranet.
 - In this model, **Client** will serve as one set of program/code which executes a set of actions over the network.
 - While **Server**, on the other hand, is a set of another program, which sends the result sets to the client system as requested.

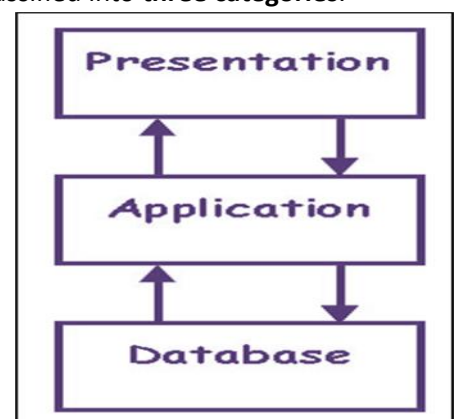
- In this, **client computer** provides an **interface** to an end user to request a service or a resource from a server and on the other hand **server** then **processes** the request and **displays** the result to the **end user**.
- ✓ An **example of Client-Server Model**– an **ATM** machine.
 - A **bank** is the **server** for processing the application within the large customer databases and **ATM** machine is the **client** having a user **interface** with some simple application processing.
 - **Platform:** In computer science or software industry, a platform is a system on which **applications program can run**.
 - It consists of a **combination** of hardware and software that have a built-in instruction for a processors/microprocessors to perform specific operations.
 - An **example** of Platform – A personal machine loaded with **Windows 2000 or Mac OS X** as examples of 2 different platforms.
 - **Database:** It is a collection of information in an **organized way** so that it can be easily accessed, managed and updated.
- ✓ **Examples of Database** – **MySQL**, **SQL Server**, and **Oracle**
- ❖ **N-Tier architecture:**
- ✓ An **N-Tier Application** program is one that is distributed among three or more separate computers in a distributed network.

The most common form of n-tier is the 3-tier Application, and it is classified into **three categories**.

1. **User interface** programming in the user's computer
2. **Business logic** in a more centralized computer, and
3. Required **data** in a computer that manages a database.

Ex. popular sites who have applied this architecture are

- MakeMyTrip.com
- Sales Force enterprise application
- Indian Railways – IRCTC, Amazon.com, etc



There are different types of N-Tier Architectures, like

3-tier Architecture, 2-Tier Architecture and 1- Tier Architecture.

3-Tier Architecture

1. Presentation layer
2. Business Logic layer
3. Database layer

1. User Interface Layer or Presentation Layer

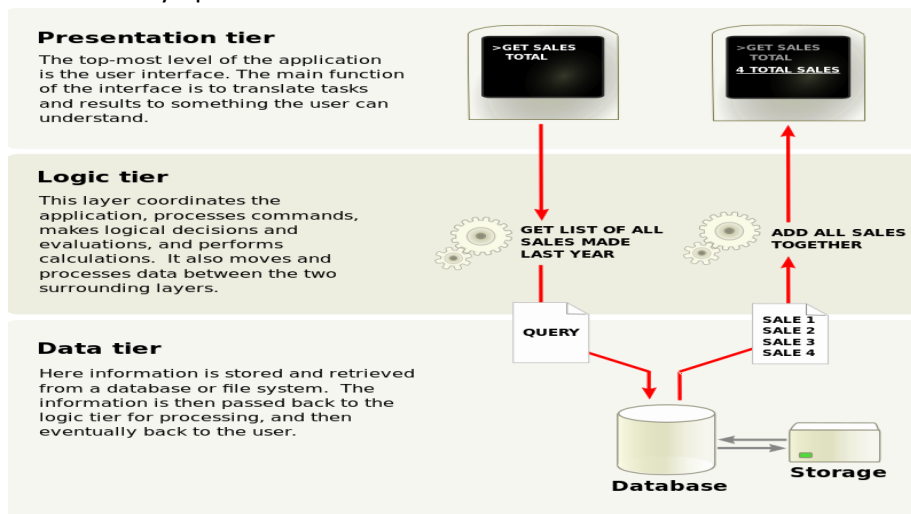
Students Information					
	ID	Name	Address	Email	Picture
➤➤					

2. Business layer

- Business logic acts as an interface between **Client layer** and **Data Access Layer**
- All business logic – like **validation** of data, **calculations**, data **insertion/modification** are written under business logic layer.

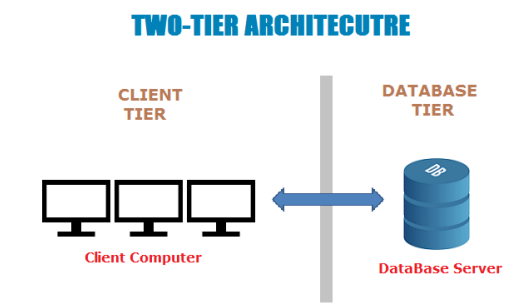
3. Data Access layer

- This is the data layer function, which receives the data from the business layer and performs the necessary operation into the database.



❖ 2-Tier Architecture:

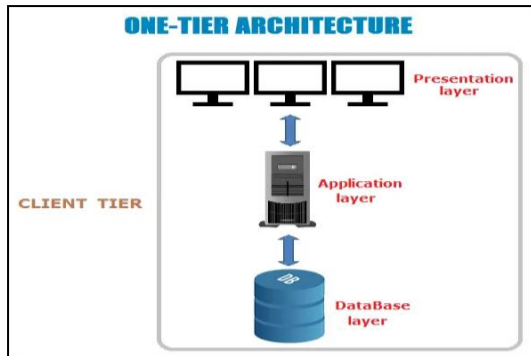
- ✓ It is like Client-Server architecture, where communication takes place between client and server.
- ✓ In this type of software architecture, the presentation layer or user interface layer runs on the client side while dataset layer gets executed and stored on server side.
- ✓ There is no Business logic layer or immediate layer in between client and server.



Single Tier or 1-Tier Architecture

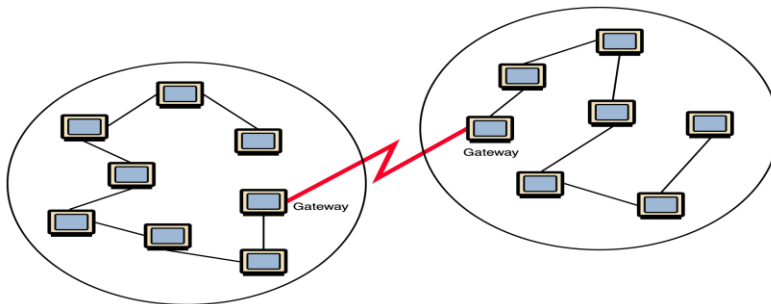
- It is the simplest one as it is equivalent to running the application on the **personal computer**.

- All of the required components for an application to run are on a single application or server.
- **Presentation** layer, **Business** logic layer, and **data** layer are all located on a **single machine**.



❖ Networks

- ✓ **A collection of computing devices connected in order to communicate and share resources.**
- ✓ Connections between computing devices can be physical using wires or cables or wireless using radio waves or infrared signals.



✓

✓ Networking key terms

- **Node (host)**
 - Any device on a network.
 - **Data transfer rate (bandwidth)**
 - The speed with which data is moved from one place to another on a network.
 - **Protocol**
 - A set of rules that defines how data is formatted and processed on a network; i.e., rules that allow client/server interaction.
- ✓ **Issue: Topology**
- If every pair of computers on a network had a dedicated communication link, passing messages would be direct and straightforward. (e.g., the “hotline” between the White House and Kremlin)
 - Unfortunately, with hundreds, thousands or millions of computers on a network, they cannot all have direct links to each other.

✓ **Network topology**

- **Ring topology**

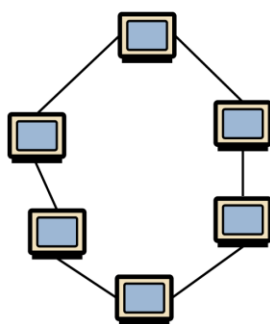
connects all nodes in a closed loop on which messages travel in one direction.

- **Star topology**

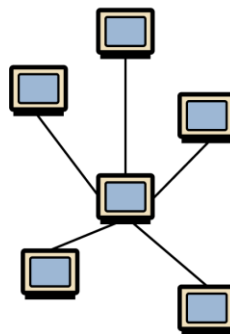
centers around one node to which all others are connected and through which all messages are sent.

- **Bus topology**

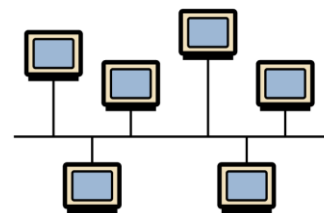
nodes are connected to a single communication line that carries messages in both directions.



Ring topology



Star topology



Bus topology

Types of Networks

✓ **Local-area network (LAN)**

- A network that connects a relatively small number of machines in a relatively close geographical area.

✓ **Wide-area network (WAN)**

- A network that connects local-area networks over a potentially large geographic distance

✓ **Metropolitan-area network (MAN)**

- The communication infrastructures that have been developed in and around large cities

✓ **Gateway**

- One particular set up to handle all communication going between that LAN and other networks

❖ Network Protocols

- ✓ In networking, a **protocol** is a standardized set of rules for formatting and processing data
- ✓ Network protocols are **layered** such that each one relies on the protocols that underlie it
- ✓ Sometimes referred to as a **protocol stack**

SMTP	FTP	Telnet
Transmission Control Protocol (TCP)		User Datagram Protocol (UDP)
Internet Protocol (IP)		

✓ Transmission Control Protocol (TCP)

- Software that **breaks messages** into packets, hands them off to the IP software for delivery, and then orders and **reassembles** the packets at their destination.

✓ Internet Protocol (IP)

- Software that deals with the **routing of packets** through the maze of interconnected networks to their final destination

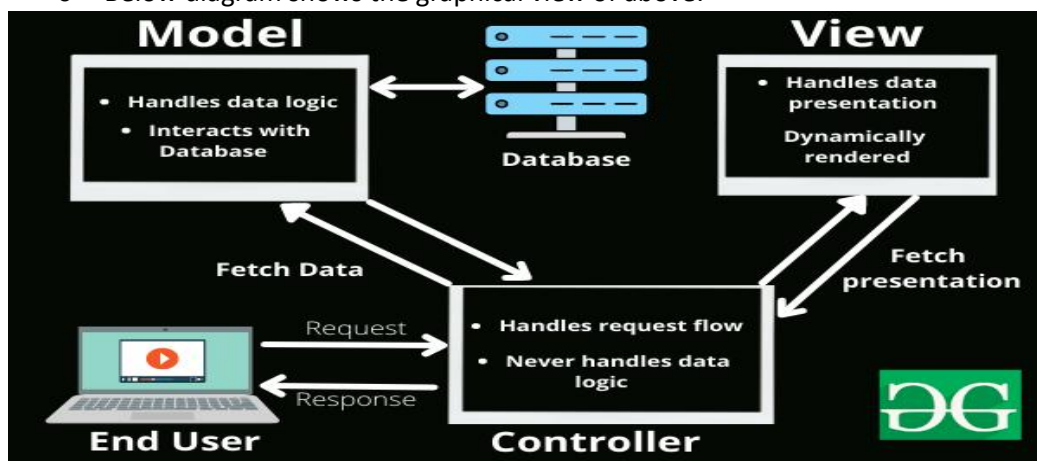
❖ High-Level Protocols:

- ✓ **Simple Mail Transfer Protocol (SMTP)** used to specify transfer of electronic mail
- ✓ **File Transfer Protocol (FTP)** allows a user to transfer files to and from another computer
- ✓ **Telnet** used to log onto one computer from another
- ✓ **Hyper Text Transfer Protocol (http)** allows exchange of Web documents

❖ MVC Pattern:

- ✓ The **intent** of MVC is to **reduce** the **coupling** among the three parts of an application, making the application easier to develop and maintain.
- ✓ In an MVC Web application, a **browser** submits requests to the **controller**, which consults the model (which in turn consults its database).
- ✓ Next, the **model** reports results to the controller and, indirectly, to the view.
- ✓ The controller then instructs the **view** to produce a result document, which is transmitted to the client for display.

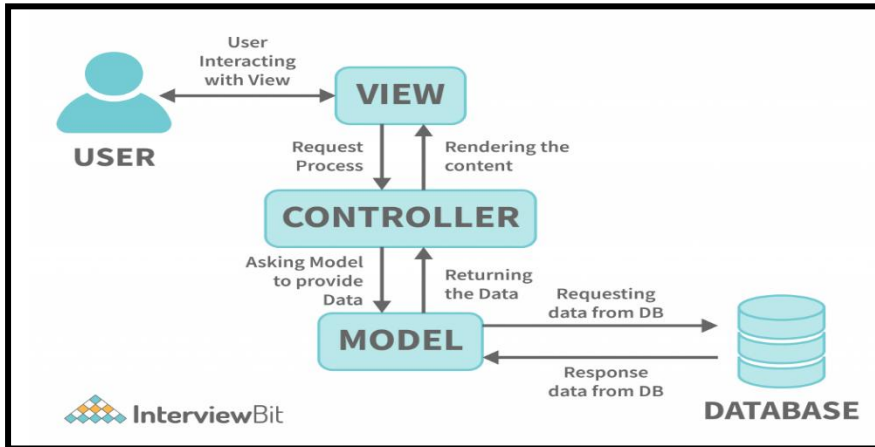
- Below diagram shows the graphical view of above:



- Web applications using Java server software can be designed and implemented with the MVC architecture.
- There are **three** general **approaches** to designing Web applications with the MVC architecture and Java server software.

- The **first** of these is the pure **JSP** approach.
- The **second** approach to MVC with Java server software is to use a combination of **servlets, JSP, and beans**.

The **third** approach to MVC with Java server software is to use **servlets** to implement the controller, **JSP** to implement the view, and Enterprise JavaBeans (EJBs) for the model.



- **The Model**

- The model is the central component of the pattern. It expresses the application's behaviour in terms of the problem domain, independent of the user interface. It directly manages the data, logic and rules of the application.
- The model is responsible for managing the data of the application. It responds to the request from the view and it also responds to instructions from the controller to update itself.

- **The View**

- A view can be any output representation of information, such as a chart or a diagram. Multiple views of the same information are possible, such as a bar chart for management and a tabular view for accountants.
- It means presentation of data in a particular format, triggered by a controller's decision to present the data. They are script-based templating systems like JSP, ASP, PHP.

- **The Controller**

- The third part or section, the controller, accepts input and converts it to commands for the model or view.
- The controller is responsible for responding to the user input and perform interactions on the data model objects. The controller receives the input, it validates the input and then performs the business operation that modifies the state of the data model.