

LECTURE NOTES ON

WEB TECHNOLOGIES

PREPARED BY:

SHIBANI MADHUSMITA SAHOO

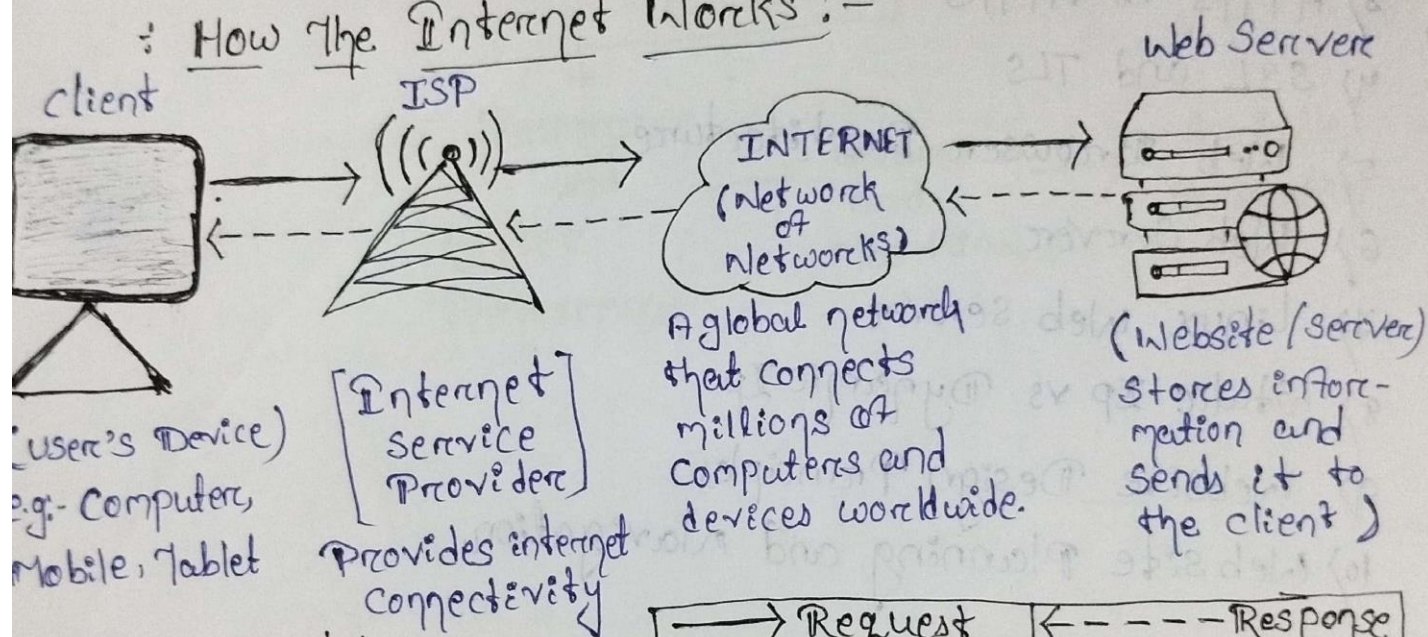
Introduction to Internet :-

* What is Internet :-

↳ Definition:- The Internet is a "Global System of interconnected computer networks" that connects millions of computers and electronic devices around the world.

→ It enables users to communicate, access information, share resources, and use various online service through standard communication protocols such as TCP/IP (Transmission Control Protocol / Internet Protocol).

How The Internet Works :-



* It allows used to :-

The internet is used for communication through emails, chats, and videocalls.

It is used for Education by attending online classes and accessing study materials.

It helps people search for information and conduct research on any topic.

The Internet is used for online shopping, banking, and digital payments.

It provides entertainment through videos, music, games and social media.

* Features of the Internet:-

- 1) Global Connectivity:- Connects computers and devices all over the world.
- 2) Fast Communication:- Instant communication through email, chat, and video calls.
- 3) Information sharing:- Allows users to access and share information easily.
- 4) 24x7 Availability:- Internet services are available anytime and anywhere.
- 5) Resource Sharing:- Allows sharing of files, software, printers, and other resources.
- 6) Multimedia Support:- Supports text, images, audio, video and animations.
- 7) Easy Access :- Information can be found quickly using browsers and search engines.
- 8) Cost-Effective:- Reduces the cost of communication and online services.
- 9) Scalability:- New users and devices can be added easily.
- 10) Reliability:- Provides dependable communication with multiple data paths.

* Advantages:-

- Instant communication.
- Easy access to information
- Online education and research.
- E-Commerce and online banking
- Entertainment and social networking

* Disadvantages:-

- Cybercrime and hacking
- Privacy & security issues
- Spread of false news.
- Internet addiction.
- Malware & viruses.

* Conclusion:-

- The Internet has become an essential part of modern life that connects people worldwide.
- It provides quick access to information and supports communication, education, business, and entertainment.

* What is WWW:-

→ WWW (World Wide Web)

→ The World Wide Web (WWW) is a collection of interconnected web pages and websites that can be accessed through the Internet using a web browser.

→ It allows users to view text, images, videos, audio, and other multimedia content using HTTP/HTTPS Protocols.

Important Point:-

* Internet:- network

* WWW:- Information available on that network.

* Father of WWW AND History of WWW:-

→ The Father of WWW is "Tim Berners-Lee".

↳ History of WWW:-

• The World Wide Web was invented to make sharing information over the Internet easier.

• Before WWW, people mostly used Command-line system. There were no modern websites.

TIME LINE:-

Year	Event
→ Before 1989	• Internet existed without websites.
→ 1989	• Tim Berners-Lee Proposed the WWW.
→ 1990	• Created HTML, HTTP, URL, First browser and First Server.
→ 1991	• WWW became Publicly available
→ 1993	• Mosaic browser increased web popularity.
→ 1994 - Present	• Billions of websites are available available worldwide.

* MOST Important Features of WWW:-

- 1) Hyperdocument:-
 - Web Pages are Connected through Hyperlinks.
- 2) Hypermedia:-
 - Supports text, images, audio, video, and animations.
- 3) User-Friendly:-
 - Easy to use through web browsers.
- 4) Global Access:-
 - Websites can be accessed from anywhere in the world.
- 5) Information sharing:-
 - Allows users to Publish and access information easily.
- 6) Interactivity:-
 - Users can interact with websites by filling forms, Searching, and shopping online.
- 7) Cross-Platform:-
 - Works on Computers, laptops, tablets, and smartphones.
- 8) Multimedia Support:-
 - Displays text, graphics, sound, videos, and animations.
- 9) Client-Server Architecture:-
 - Works on the request-response model between client and server.
- 10) Dynamic Content:-
 - Websites can display updated and real-time information.

* Components of WWW:-

- 1) Web Browser:-
 - A software application used to access and view web pages.
 - Example:- Google Chrome, Mozilla Firefox, Microsoft Edge.
- 2) Web Server:-
 - A computer that stores websites and sends web pages to users when requested. Example:- Google Web Server.

3) Web Page:-

→ A single document on the World Wide Web containing text, images, videos, and links.

Example:- Home Page, Contact Page.

4) Website:-

→ A collection of related web pages under one domain name.
Ex:- www.google.com, www.wikipedia.org.

5) URL (Uniform Resource Locator):-

→ The unique address of a web page used to locate it on the Internet.
Ex:- <https://www.google.com>.

6) HTTP / HTTPS:-

→ A communication protocol used to transfer web pages between the browser and the web server.

* HTTP = Hypertext Transfer Protocol.

* HTTPS = Secure version of HTTP.

7) HTML:- (Hypertext Markup Language)

→ The standard language used to create & structure web pages.

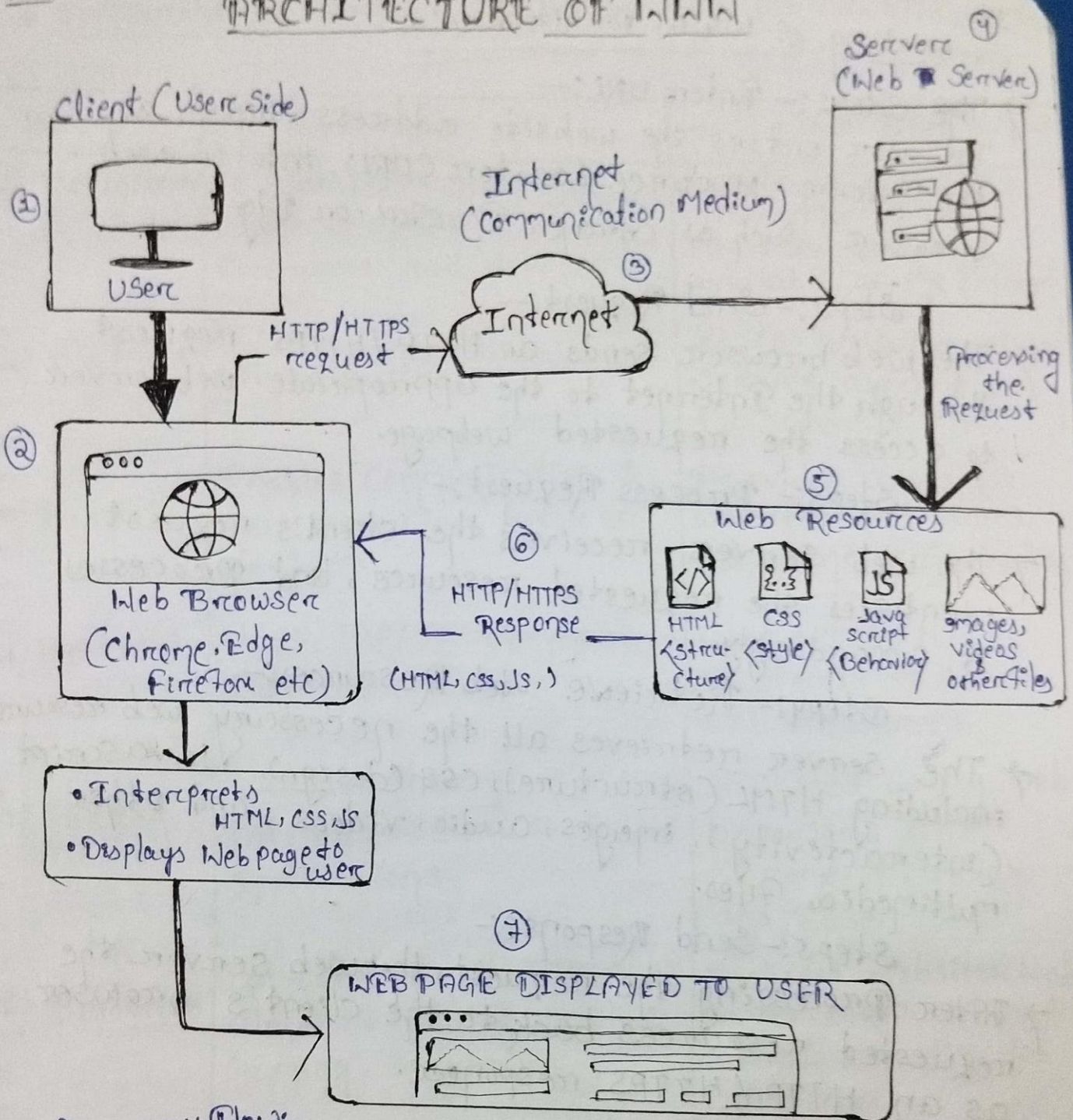
* Architecture of WWW:- 5 marks Important

Definition:-

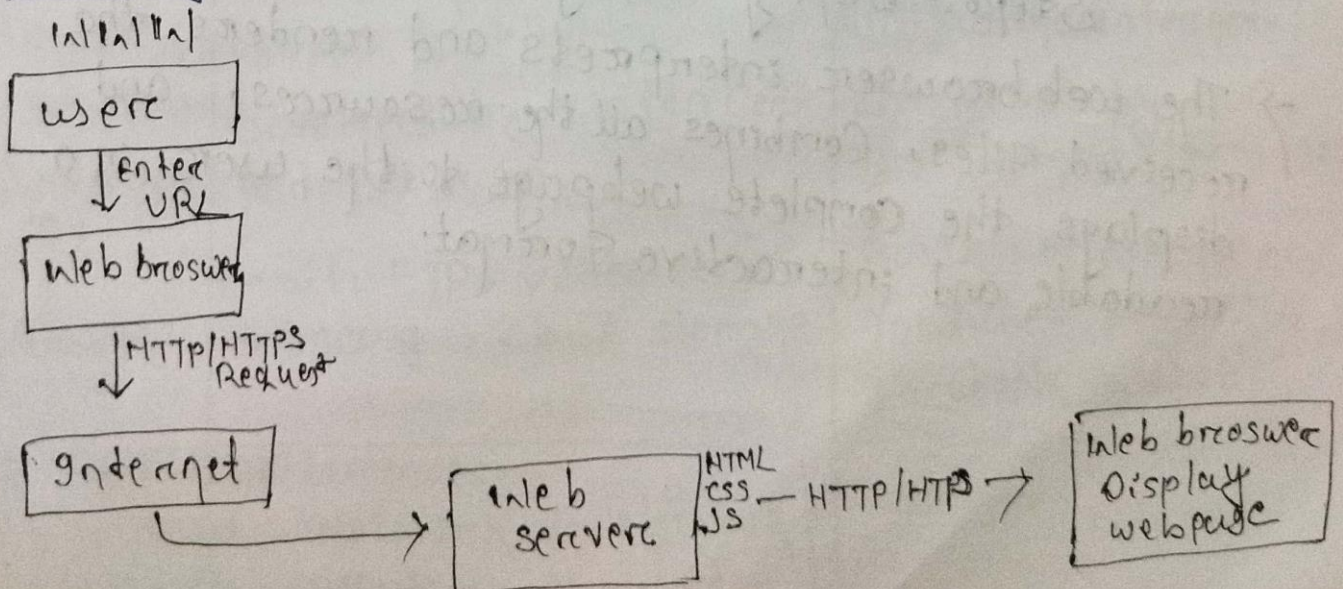
→ The Architecture of the World Wide Web (WWW) is the structural design and communication framework that enables users to access web pages over the Internet.

→ It follows a client-server architecture, where the client (web browser) sends a request to the web server, and the server processes the request and returns the requested web page using HTTP / HTTPS protocols.

ARCHITECTURE OF WWW



* Summary Flow:-



Working of WWW Architecture:-

Step 1:- Enter URL:-

→ The user enters the website address, known as the Uniform Resource Locator (URL), into a web browser such as Chrome, Firefox, or Edge.

Step 2:- Send Request:-

→ The web browser sends an HTTP/HTTPS request through the Internet to the appropriate web server to access the requested webpage.

Step 3:- Process Request:-

→ The web server receives the client's request, identifies the requested resource, and processes it accordingly.

Step 4:- Retrieve Web Resources:-

→ The server retrieves all the necessary web resources including HTML (structure), CSS (design), JavaScript (interactivity), images, audio, videos, and other multimedia files.

Step 5:- Send Response:-

→ After processing the request, the web server sends the requested resources back to the client's browser as an HTTP/HTTPS response.

Step 6:- Display Webpage:-

→ The web browser interprets and renders the received files, combines all the resources, and displays the complete webpage to the user in a readable and interactive format.

* Differentiate between Internet and World-Wide-Web.

Basis	Internet	World Wide Web
1) Definition	A global network that connects millions of computers and devices.	A collection of interconnected web-pages and websites available on the Internet.
2) Nature	It is a network infrastructure.	It is an information service running on the Internet.
3) Purpose	Enables communication & data transfer between devices.	Provides access to websites, web pages, and online information.
4) Protocol	Uses TCP/IP Communication Protocols.	Uses HTTP/HTTPS Protocols to access web resources.
5) Access Method	Accessed through various Internet services & applications.	Accessed using web browsers like Chrome, Firefox, and Edge.
6) Services	Includes Email, FTP, VoIP, cloud computing, online gaming, and WWW.	Includes websites, web pages, blogs, web applications, & multimedia content.
7) Dependency	The Internet can function without the WWW.	The WWW cannot exist without the Internet.
8) Invented By	Developed by Vinton Cerf and Bob Kahn	Invented by Tim-Berners-Lee in 1989
9) Example	Email, FTP, video conferencing, cloud storage	Google, Wikipedia, YouTube, Amazon, Facebook websites ... etc.

* Key - Difference:-

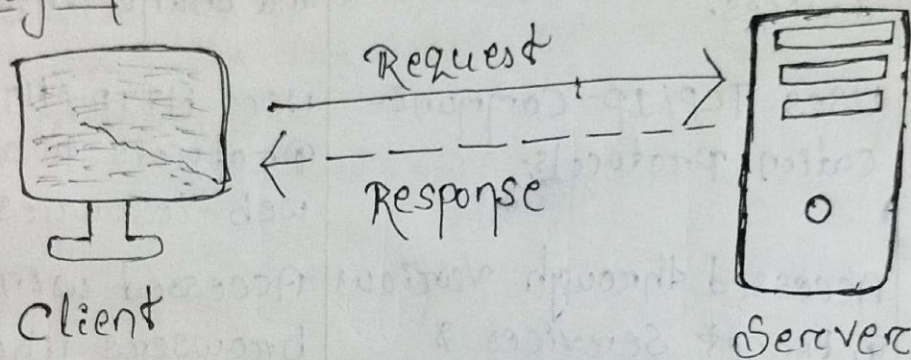
- ↳ Internet:- Network of Network
- ↳ WWW:- Collection of Websites

* Client - Server Model:-

↳ Definition:-

- The client-server Model is a network architecture in which a client requests services or resources, and a server processes the request and provides the required response.
- It enables efficient communication and centralized resources sharing in a network.
- This model forms the backbone of modern web applications, cloud computing and Internet services.

↳ Diagram:-



↳ Components:-

- 1) Client:- A client is a computer, smartphone, or software application that sends requests to the server to access information or services.

Example:- Google Chrome, Mozilla, Firefox, mobile apps.

- 2) Server:- A server is a powerful computer or software that receives requests from clients, processes them, retrieves data if necessary, and sends responses.

Example:- Web server, Database server, mail server, file server.

- 3) Database:- The database stores application data such as user information, products, transactions, and records. The server communicates with the

database whenever data needs to be retrieved or updated.

4) Network:- The network provides the communication path between the client and the server.

Examples:- Internet, LAN, Wi-Fi

5) Communication Protocols:- Protocols define the rules for communication betⁿ the client and server.

Examples:- HTTP/HTTPS, FTP, TCP/IP

* Working of Client - Server Model:-

- 1) The user enters a website address (URL) in the browser.
- 2) The browser (client) sends an HTTP/HTTPS request to the web server.
- 3) The server receives and analyze the request.
- 4) If required, the server accesses the database.
- 5) The database returns the requested information to the server.
- 6) The server prepares the response.
- 7) The browser receives the response and displays the webpage to the user.

DISADVANTAGES

* ADVANTAGES

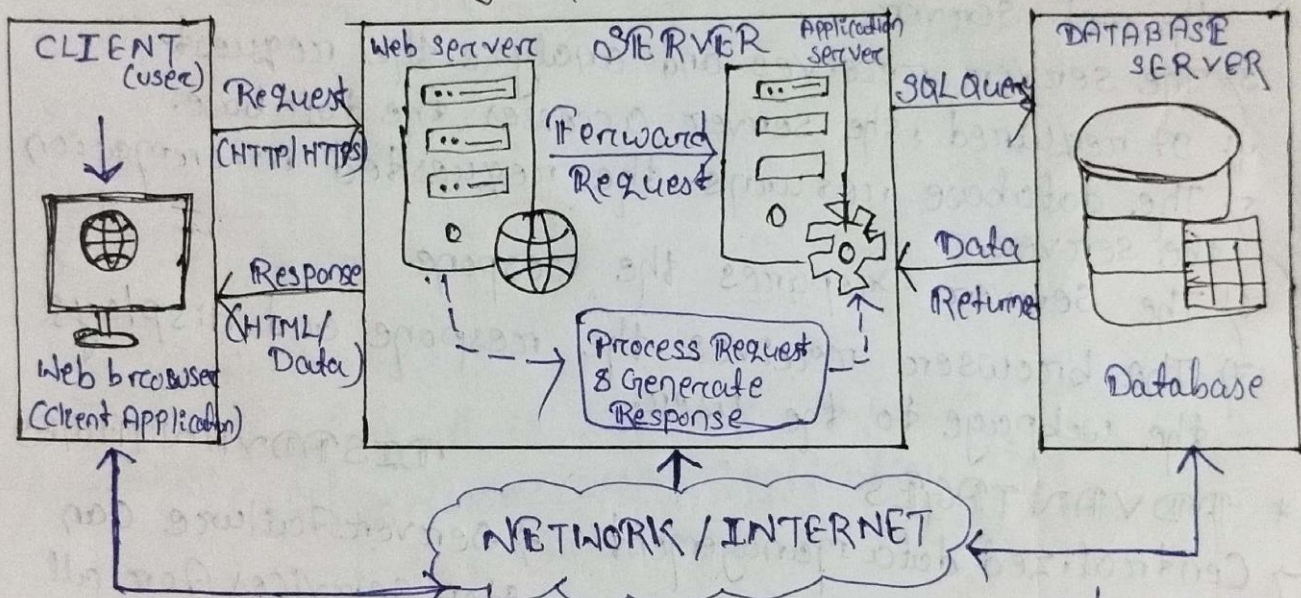
- Centralized data management.
- Better security.
- Efficient resource sharing.
- Easy maintenance and updates.
- Supports multiple users simultaneously.
- Reliable backup and recovery
- High scalability and performance.

- Server failure can stop services for all clients.
- High installation and maintenance cost.
- Performance may decrease during heavy traffic.
- Requires a stable network connection.
- Needs skilled administrators for management.

* Applications:-

- Web Browsing (Google, Chrome)
- E-Commerce (Amazon, Flipkart)
- Online Banking
- Email Services
- Social Media (Facebook, Instagram)
- Cloud Computing
- Online Education
- Video Streaming (YouTube, Netflix)

* Architecture Diagram of Client-Server Model:-



- 1) Client (User):- Sends a request to the web server using a browser.
- 2) Web server:- receives the request and forwards it to the Application server.
- 3) Application server:- Processes the request, interacts with the Database server if required.
- 4) Database server:- Sends the required data back to the Application server.
- 5) Application server generates the response and sends it back to the web server.
- 6) Web server sends the final response back to the client.

* Conclusion:-

- The Client-Server Model is a fundamental architecture in computer networking & web technologies. It enables secure, reliable, and efficient communication betⁿ clients and servers.

Web Systems Architecture:

* What is Architecture?

→ Architecture is the art and science of planning, designing, and constructing buildings or physical space. It is the perfect blend of art (how a space look and feels) and science (how it is safely and practically built)

↳ The three main goals of architecture are:-

- **Function:-** A building must work well for the people using it.
- **Durability:-** It must be structurally sound and strong enough to withstand the elements.
- **Beauty:-** It should be visually pleasing and emotionally uplifting.

* Web Systems Architecture:

↳ Architecture of web based system:-

↳ Definition:-

→ Web Systems Architecture is the structure and design of a web-based application that shows how different components such as the client (browser), web server, application server, database server, cache, proxy, load balancer, and queue communicate with each other to process user requests and provide web pages or services.

* In simple words, it is the framework that defines how a website or web application works internally.

* Need for Web Systems Architecture:—

- Web architecture is required because modern web applications may have thousands or millions of users at the same time.
- A proper architecture ensures that the application remains fast, secure, reliable, and scalable.

* Objectives:—

- To Process user requests efficiently.
- Improve the website Performance.
- Provide security for users and data.
- To reduce server overload.
- Support the easy maintenance and updates.
- To provide high availability of web services.

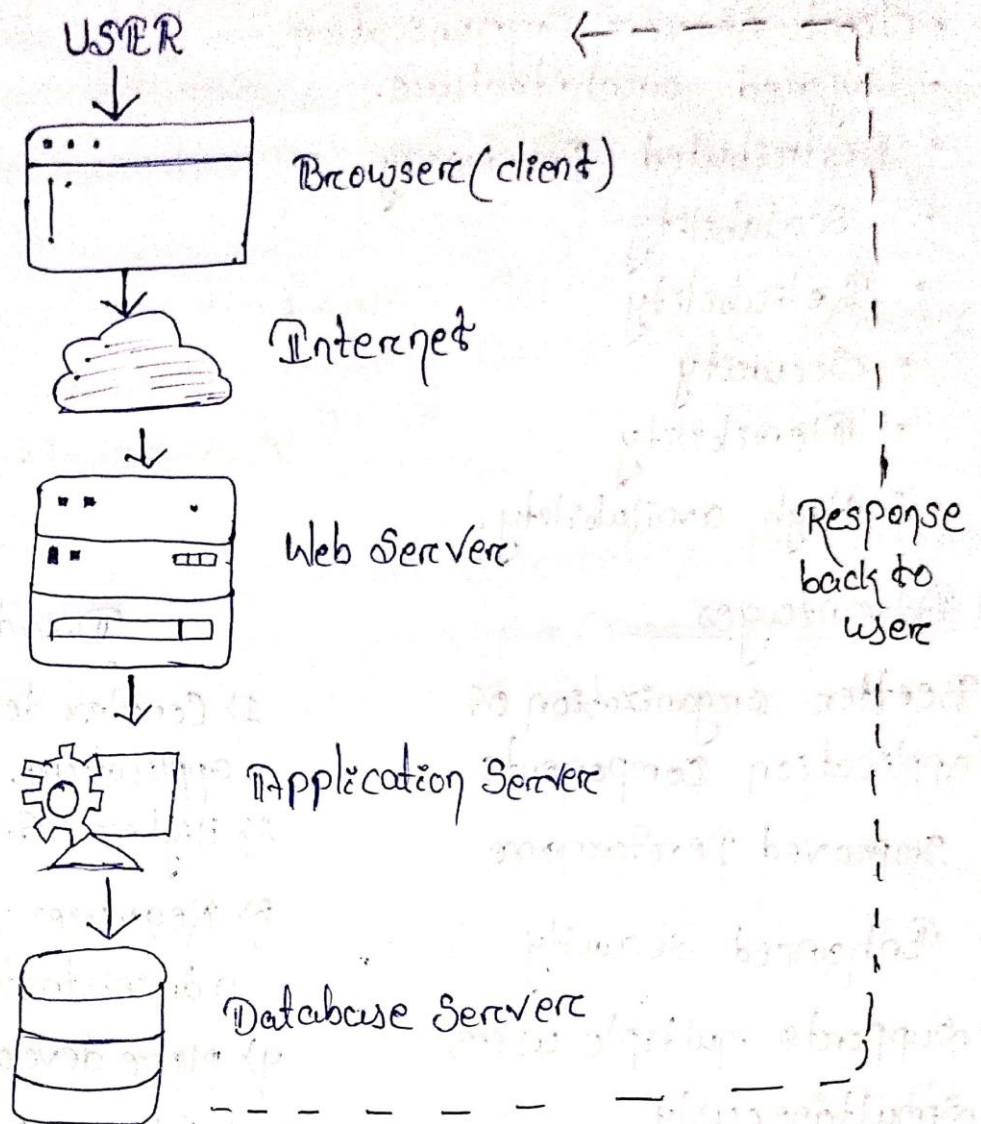
* Main Components of web Systems Architecture:—

Components

Function

- 1) Client (Browser) ---> Displays webpages and sends requests.
- 2) Web server -----> Receives HTTP requests.
- 3) Application server -----> Processes business logic.
- 4) Database server -----> Stores and retrieves data.
- 5) Cache - - - - -> Stores frequently accessed data.
- 6) Proxy - - - - -> Acts as an intermediary betⁿ client and server.
- 7) Load Balancer - - - -> Distributes traffic among multiple servers.
- 8) Queue - - - - -> Processes tasks asynchronously.

* Basic Architecture Diagram:-



- 1) User enters a URL in the web browser.
- 2) Browser sends an HTTP/HTTPS request through the Internet.
- 3) Web Server receives the request.
- 4) Web server forwards the request to the application server.
- 5) Application server processes the request and accesses the database if required.
- 6) Database returns the required data, and the application server prepares the response.
- 7) Web server sends the response to the browser and the browser displays the web.

* Characteristics:-

- Client-Server Communication.
- Layered architecture.
- Distributed Processing.
- Scalability
- Reliability
- Security
- Flexibility
- High availability.



* Advantages

- 1) Better organization of application components
- 2) Improved Performance
- 3) Enhanced Security
- 4) Supports multiple users simultaneously.
- 5) Easy scalability

* Disadvantages

- 1) Complex design for large applications.
- 2) Higher infrastructure cost
- 3) Requires server administration.
- 4) More development effort
- 5) Network dependency
- 6) Maintenance of multiple components.

* Applications:-

- E-Commerce website
- Online banking systems
- Social media platforms
- Hospital management systems
- Government portals
- Online examination systems

* ~~Client~~

* Tier-Based (Layered Architectures):—

• 1-Tier Architecture:—

→ The User interface, business logic, and database all sit on a single device.

• 2-Tier Architecture:— (client - Server)

→ The client (Frontend) directly communicates with the Server / database (Backend).

• 3-Tier Architecture:—

→ The standard enterprise model separating the Presentation Layer (UI), Business Logic Layer (Processing), and Data Layer (database).

• N-Tier Architecture:—

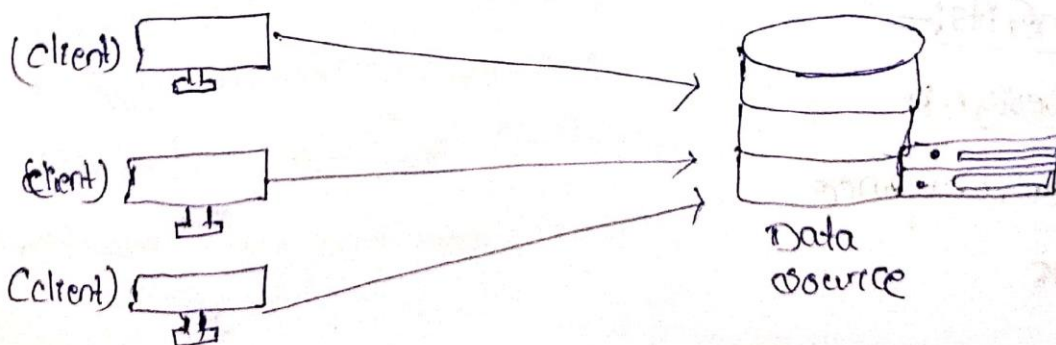
→ A multi-layered structure that breaks the application down into more than three specialized layers for Security and Scale.

* What is 2-Tier (client - Server) Architecture:—

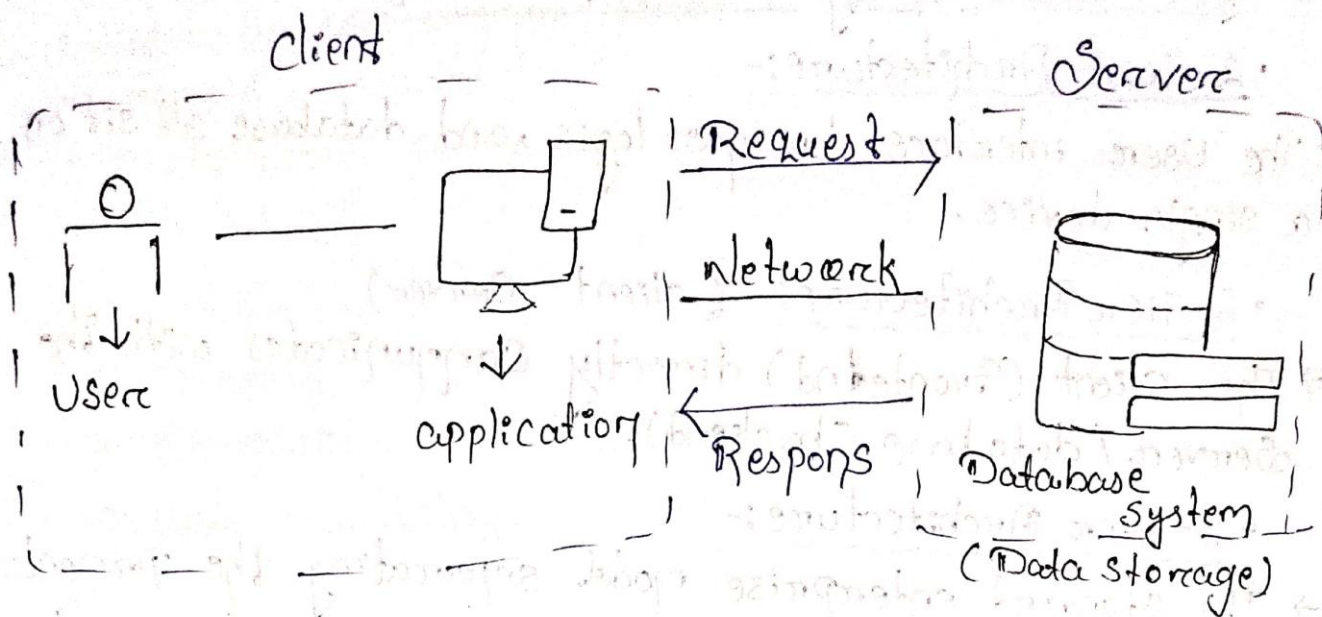
↳ Definition:—

→ In 2-tier architecture, the client communicates directly with the database server. The client contains the user interface and application logic, while the server stores data.

* Basic Diagram:—



Working Diagram of 2-Tier (Client-Server) Architecture



* Components:-

- Client:- User interface and application logic.
- Database Server:- Stores and retrieves data.

* Core Layers:-

* Client Tier (Tier 1):-

- The front-end user interface running on a user's device (like a desktop application or forms).
- It handles user interaction and sometimes basic business rules.

* Database Tier (Tier 2):-

- The back-end database server that stores data, runs queries, and manages transactions.

* Key Benefits:-

- Simple Design
- Fast Performance
- Low Cost

* Working:-

- 1) The user enters data through the client application.
- 2) Client sends a request directly to the database server.
- 3) The database server processes the request.
- 4) Required data is retrieved or updated.
- 5) The result is returned directly to the client.
- 6) The client displays the ~~result~~ result.

* Features:-

- Simple architecture
- Direct communication between client and database.
- Limited number of layers.
- Suitable for small applications.
- Fast communication in local networks.

* Advantages

- Easy to design
- Easy to implement
- Low development cost
- Fewer components
- Faster for small applications.
- Low hardware requirements

Disadvantages

- Poor scalability
- Weak security
- Heavy load on the database server.
- Difficult maintenance for large application
- Tight coupling betⁿ client & database
- Difficult software updates.

* Applications:-

- Library management system
- Small office applications
- Local inventory systems
- Desktop database software
- Student record applications

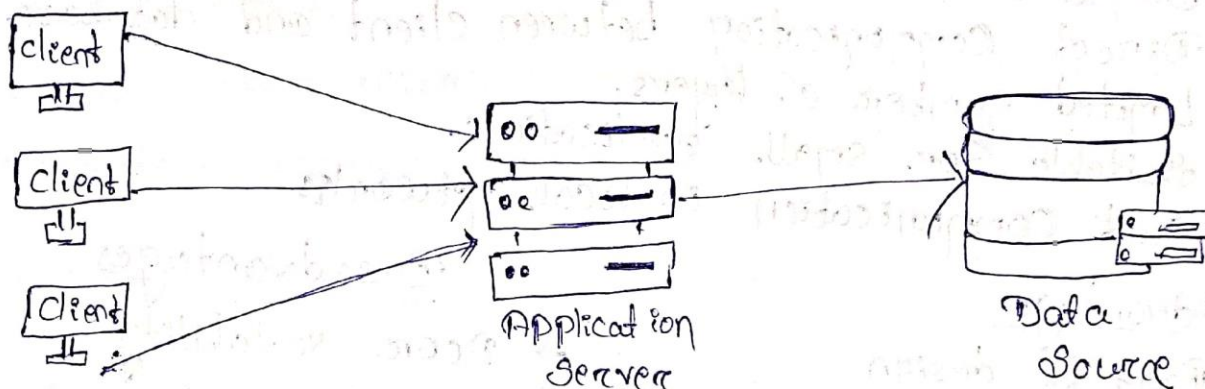
* 3-Tier Architecture:-

* Definition:-

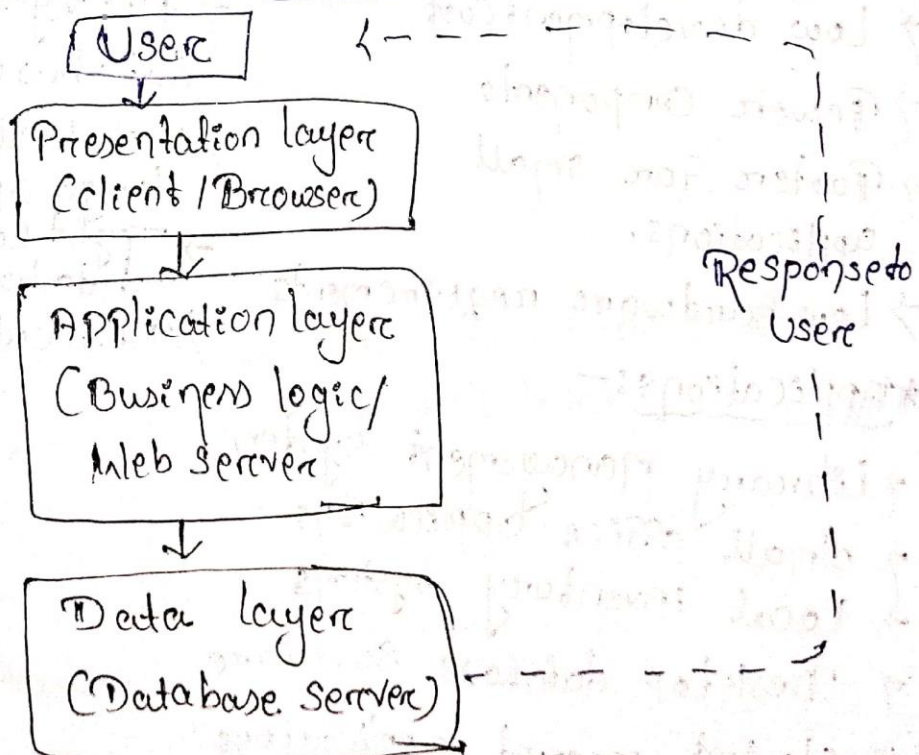
→ 3-Tier Architecture is a web application architecture that divides an application into three separate layers:- Presentation layer, Application layer, and Database-layer.

→ Each layer performs a specific function and communicates with the other layers.

* Basic Diagram:-



* Working Diagram of 3-Tier Architecture:-



* Main Component:-

- 1) Presentation Layer (Client Layer)
- 2) Application Layer (Business Logic Layer)
- 3) Data Layer (Database Layer).

* Layers in Detail:-

1) Presentation Layer:- (Client Layer)

- This is the front-end layer.
- It provides the user interface (UI).
- It receives input from the user and displays the output.
- Communicate with the application server.

Examples:- Web browser, HTML, CSS, JavaScript.

2) Application Layer (Business Logic):-

- This is the middle layer.
- It processes the business logic of the application.
- It validates user input.
- It communicates with the database layer.

Examples:- Java, PHP, Python, Node.js, .NET.

3) Data Layer (Database Layer):-

- This is the back-end layer.
- It stores and manages all application data.
- It performs CRUD operations [Create, Read, Update, Delete]
- It returns data to the application layer.

Example:- MySQL, PostgreSQL, Oracle, SQL Server.

* Working:-

- 1) User sends a request through the browser.
- 2) The request reaches the application server.
- 3) The application server processes business rules.
- 4) It communicates with the database server.
- 5) The database returns data.
- 6) The application server prepares the response.
- 7) The browser displays the ~~result~~ result.

* Example:-

Online Banking System:-

* Browser:- Customer interface

* Application:- account verification and transaction
server processing.

* Database:- Customer accounts and transaction records.

* Features:-

- Layer separation
- Better security
- High scalability
- Reusability of components.
- Independent deployment of layers

* Applications:-

- Amazon
- Flipkart
- Gmail
- Facebook
- Online banking
- Railway reservation systems.

* Advantages

- High Security
- Better Scalability
- Easy maintenance
- Reusable Components
- Easy deployment
- Supports large web application

Disadvantages

- More complex architecture.
- Higher Cost
- Requires additional Servers
- More development effort.

* Different between 2-Tier & 3-Tier Architecture:—

Basic of Comparison	2-Tier Architecture	3-Tier Architecture
1) Definition	Client directly communicates with the database server.	Client communicates with the database through an application server.
2) Number of layers	2-layers	3-layers
3) Business logic	Stored in the client applications.	Stored in the application server.
4) Communication	Direct Communication bet ⁿ client and database	Indirect Communication through the application server.
5) Security	less secure because the database is directly exposed.	More secure because the database is protected by the application layer.
6) Scalability	Limited Scalability	Highly Scalable.
7) Maintenance	Difficult to maintain and update.	Easy to maintain because layers are independent.
8) Cost	Lower implementation cost.	Higher implementation cost.
9) Complexity	Simple architecture.	More complex architecture.
10) Examples	• Library Management System. • Small office Application	• Online Banking • Amazon • Flipkart • College ERP

Q1 Building blocks of Fast and Scalable data access

* Building blocks of Fast and Scalable data access Concepts:-

- Modern web application such as Amazon, Flipkart, Gmail, YouTube, and online Banking are accessed by thousands or even millions of users simultaneously.
- If every user request goes directly to the database or server, the system becomes slow and overloaded.
- Therefore, web applications use several building blocks to improve speed performance, reliability, and scalability.

OR

* Definition:-

- Building blocks of Fast and Scalable Data Access are the components used in web systems to reduce server load, improve response time, increase availability, and efficiently manage large numbers of user requests.

* Main Building Blocks:-

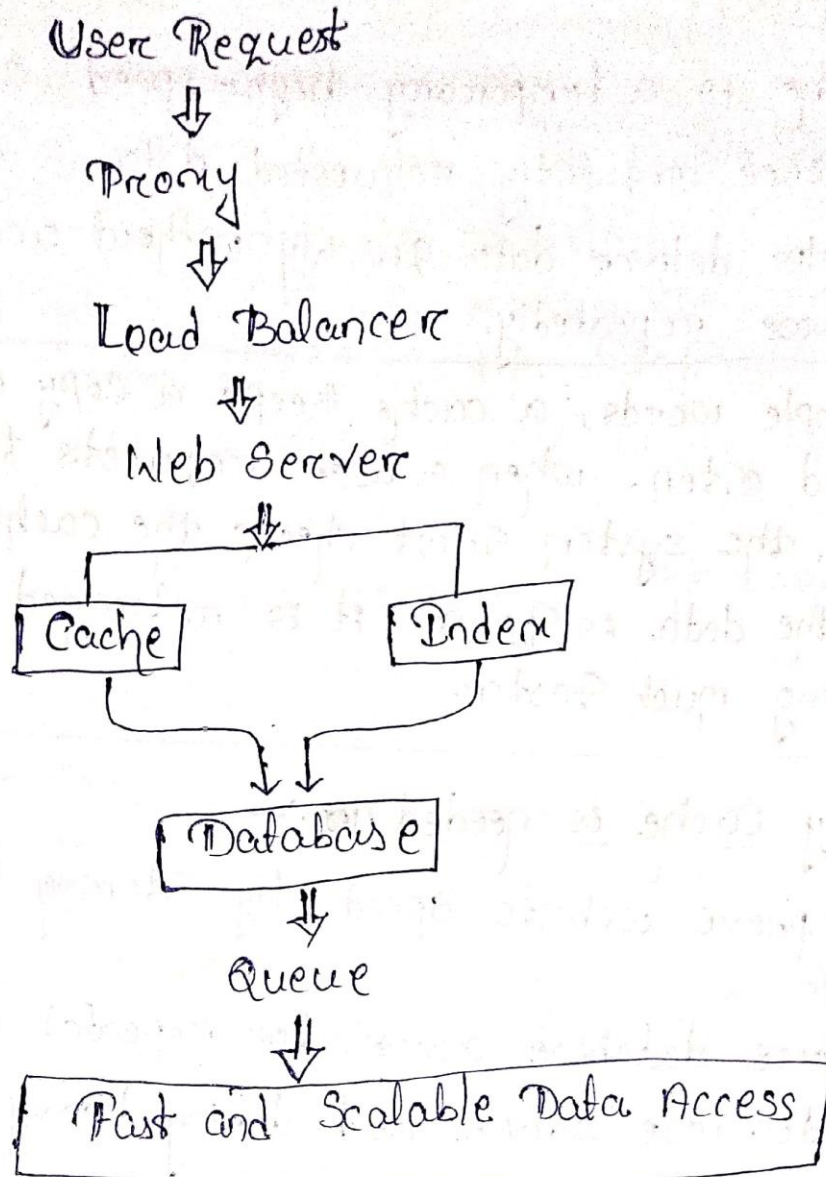
- 1) Cache
- 2) Proxy
- 3) Index
- 4) Load Balancer
- 5) Queue.

* Simple Definition:-

- * Building Blocks:- Basic components used to build a system.
- * Fast:- Able to process requests quickly.
- * Scalable:- Able to handle many users or requests efficiently.
- * Data Access:- The process of reading or writing data from a database.

* Building Blocks of Fast and Scalable Data Access:

Diagram:-



These Components work together to ensure that web applications remain fast, secure, and scalable even during heavy traffic.

* Used:-

- 1) Improve website speed by delivering data quickly.
- 2) Handle a large number of users at the same time.
- 3) Reduce server load
- 4) Reduce database load using Cache & indexing.
- 5) Support background processing through queues.
- 6) Provide a better user experience with faster response time.

* Cache:-

↳ Definition:-

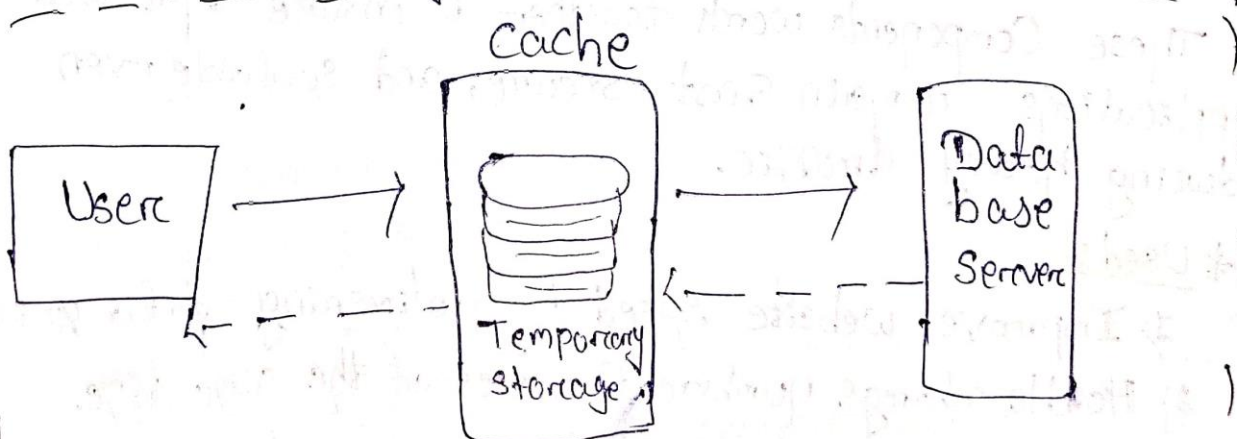
- A cache is a temporary high-speed storage area.
- It stores frequently requested data.
- It helps deliver data quickly without accessing the database repeatedly.

* In simple words, a cache keeps a copy of data that is used often. When a user requests the same data again, the system first checks the cache.

→ If the data is found, it is returned immediately, making much faster.

↳ Why Cache is needed :-

- * To improve website speed by storing frequently used data.
- * Reduce database access for repeated requests.
- * To decrease server load during heavy traffic.
- * Support a large number of users simultaneously.



Basic Working of cache

* Working of Cache / Step-by-step Process:-

- 1) The user requests a webpage.
- 2) Then server checks the cache.
- 3) If data is found (Cache Hit):- the cache returns the data immediately.
- 4) If data is not found (Cache Miss):- the server fetches data from the database.
- 5) The fetched data is stored in the cache for future requests.
- 6) The server sends the response to the user.

* Types of Cache:-

- 1) Web ^{Browser} Cache:-
 - Stores web pages, images, CSS, and JavaScript files in the user's browser.
- 2) Web server Cache:-
 - Stores frequently requested webpages on the webserver.
 - Reduces web server processing time.
 - Improves page loading speed.
- 3) Database Cache:-
 - Stores database query results.
 - Reduces repeated database access.
 - Speeds up data retrieval.
- 4) Memory ^{Cache} Cache:-
 - Stores data directly in RAM (Random Access Memory).
 - Provides extremely fast access.
 - Commonly used for sessions and application data.

* Examples:- of cache technologies:-

- 1) Redis → Fast memory Cache
- 2) Memcached → Memory-based Cache
- 3) Browser Cache → Browser stores website files.
- 4) CDN Cache → Nearby server stores website Content.

* Cache Hit and Cache Miss:-

<u>Term</u>	<u>Meaning</u>
Cache Hit	Requested data is found in the cache.
Cache Miss	Requested data is not found in the cache.
Cache Refresh	Old cache is replaced with updated data.
Cache Expiration	Cache data is removed after a timed time.

* Advantages

- Faster response time
- Reduced database load
- Improved website performance
- Better user experience
- Reduced network traffic

Disadvantages

- Extra memory is required
- Cache expiration management is necessary.
- Possibility of outdated or stale data.
- Cache synchronization issues.
- Additional Configuration Complexity.

* Applications:-

- E-Commerce websites
- Social media platforms
- News websites
- Online banking

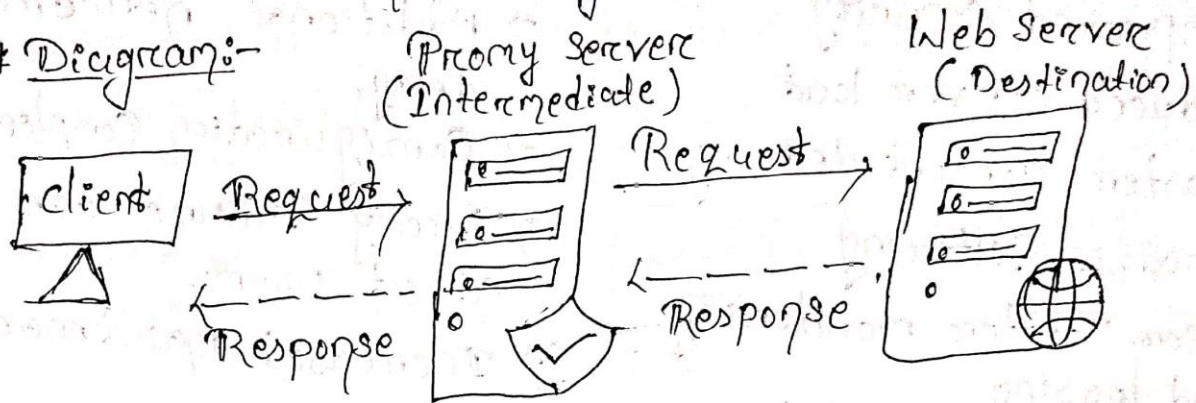
* Proxy:-

* Definition:-

→ A Proxy Server is an intermediary server that receives requests from clients and forwards them to another server.

→ It acts as a middle layer betⁿ the client and the

* Diagram:-



* Working Process:-

- 1) Client sends a request to the Proxy Server.
- 2) Proxy Server receives and checks the request.
- 3) If data is available in the proxy cache, it sends the data directly to the client.
- 4) If data is not available, the proxy forwards the request to the web server.
- 5) Web server processes the request and sends the response to the Proxy Server.
- 6) Proxy server may store the response in its cache for future requests.
- 7) Proxy server returns the response to the client.

* Types of Proxy:-

1) Forward Proxy:-

- Works on behalf of the client.
- Hides the client's identity.
- Used for internet access control.

Reverse Proxy:-

- Works on behalf of the server.
- Hides the server's identity.
- Used for security, caching, and load balancing.

* Advantages

- Improved Security
- Reduced server load
- Faster Content delivery
- Traffic Filtering
- ~~User~~ Better Monitoring and logging

Disadvantages

- Additional network delay.
- Configuration Complexity
- Proxy failure can affect users.
- Increased maintenance

* Functions:-

- Request Forwarding
- Content caching
- Security Filtering
- Access Control
- Hiding client identity
- Hiding server identity

* Applications:-

- Corporate networks
- Content Delivery Networks (CDN)
- Reverse Proxy web servers.
- Internet access control systems.
- Educational institutions

* Examples:-

- Nginx
- squid Proxy
- Apache HTTP server
- CloudFlare Reverse Proxy.

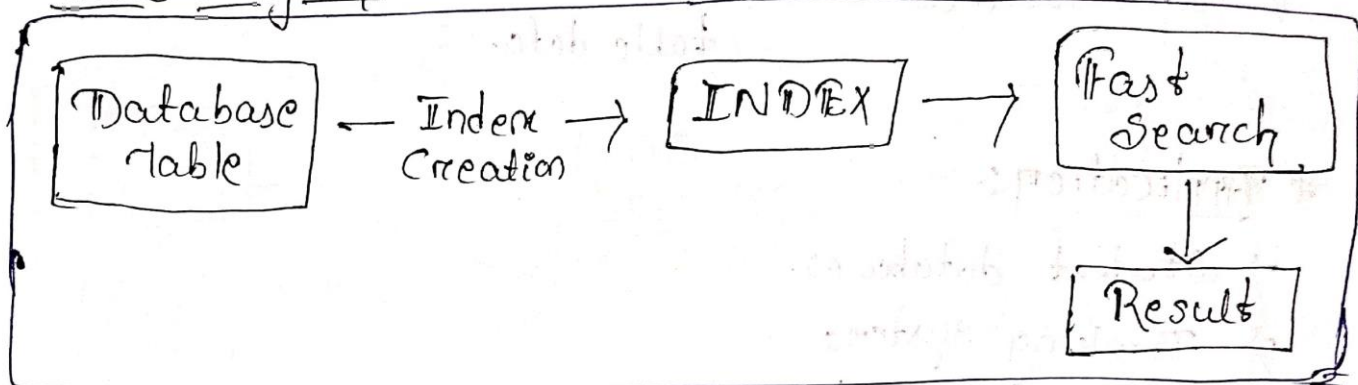
* Index :-

* Definition :-

→ An index is a database structure that improves the speed of searching, sorting, and retrieving records from a database table.

→ It works similarly to the index of a book, where required information can be found quickly.

* Basic Diagram :-



* Working :-

1) Without Index :-

→ The database scans every row in the table.

→ Search becomes slow for large tables.

2) With Index :-

→ The database uses the index to locate the required record directly.

→ Search becomes much faster.

* Examples :-

- B-tree Index
- Hash Index
- MySQL Index
- PostgreSQL Index

* Types of Index:-

- 1) Primary Index:- Created on the primary key.
- 2) Secondary Index:- Created on non-Primary key columns.
- 3) Clustered Index:- Determines the physical order of records.
- 4) Non-clustered Index:- stores index separately from table data.

* Application:-

- Student databases.
- Banking systems
- Hospital records
- E-Commerce Product Search
- Library management systems

* Advantages

- Faster searching
- Faster sorting
- Faster data retrieval
- Reduced search time
- Improved database

Disadvantages

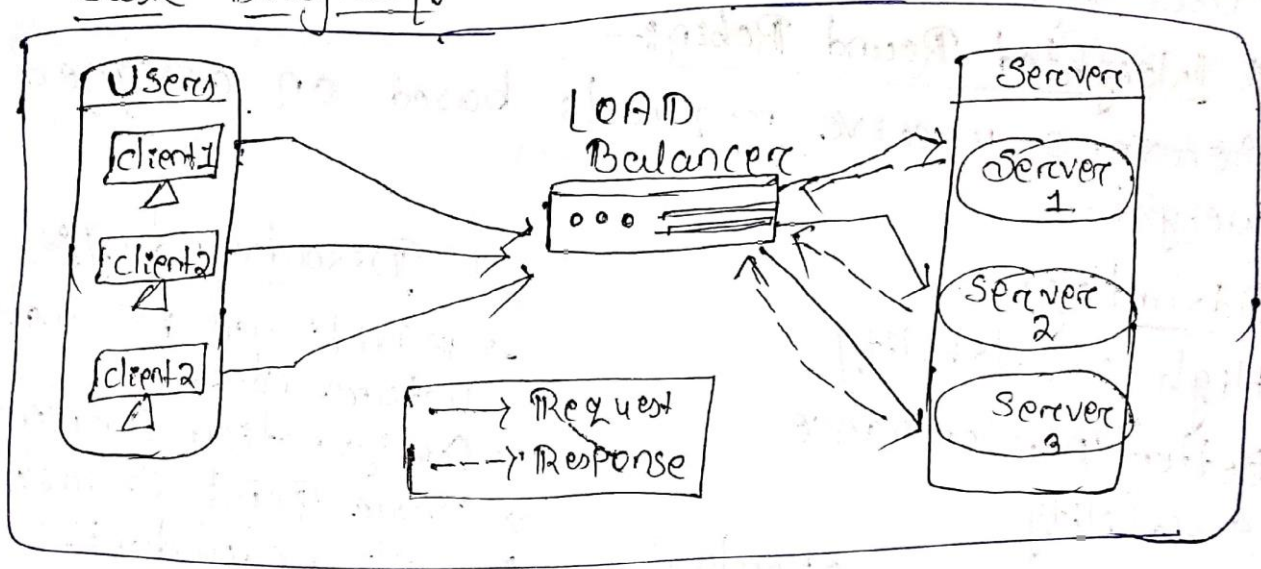
- Additional storage space
- slower insert operations
- slower update operations
- slower delete operations
- Index maintenance overhead.

* Load Balancer:-

* Definition:-

- A load balancer is a network component that distributes incoming requests among multiple servers so that no single server becomes overloaded.
- It ensures that all servers share the workload equally.

* Basic Diagram:-



* Interworking:-

- 1) Multiple users send requests.
- 2) The load balancer receives all requests.
- 3) It checks server availability and load.
- 4) It selects the most suitable server.
- 5) The request is forwarded to that server.
- 6) The server processes the request.
- 7) The response is returned to the user.

* Load Balancing Methods:-

- 1) Round Robin:-
 - Requests are distributed sequentially.
- 2) Least Connections:-
 - Selects the server with the fewest active connections.
- 3) IP Hash:-
 - Uses the client IP address for server selection.
- 4) Weighted Round Robin:-
 - Servers receive requests based on assigned weights.

* Advantages

- High availability
- Better performance
- Scalability
- Reduced server overload

Disadvantages

- Additional infrastructure cost.
- Configuration complexity
- Single point of failure if not redundant.
- Requires monitoring

* Examples:-

- Nginx
- HAProxy
- AWS Elastic Load Balancer
- F5 Load Balancer

* Applications:-

- E-Commerce websites
- Banking systems
- Cloud platforms
- Streaming services
- Government portals

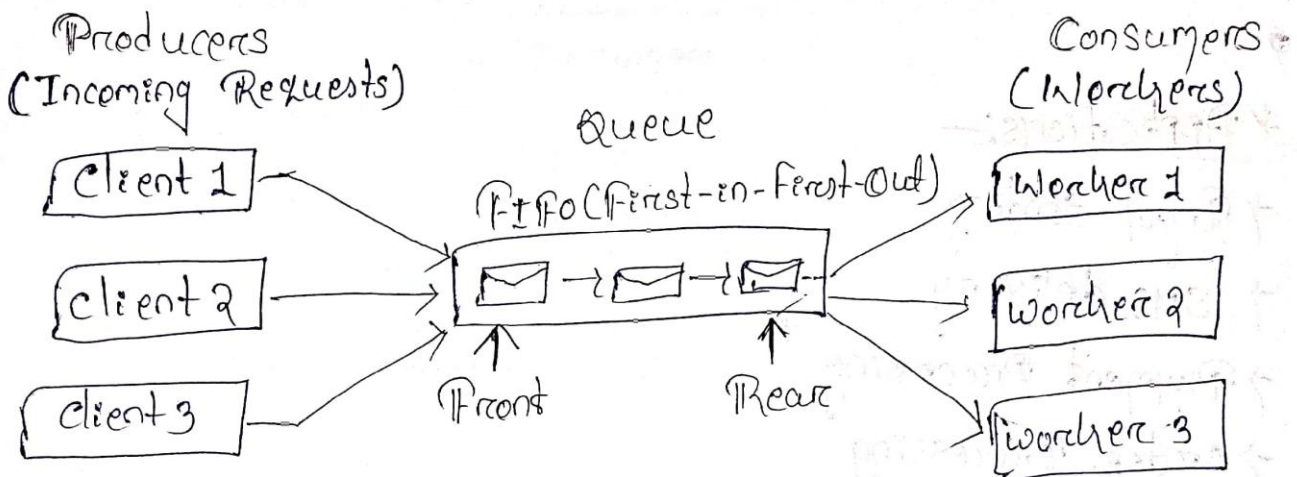
* Queue:-

* Definition:-

→ A queue is a data structure that stores tasks or requests temporarily and processes them in order, usually following FIFO (First-In-First-Out).

→ Queues are used for background or asynchronous processing.

* Basic Diagram:-



* Working:-

- 1) Requests arrive from users.
- 2) They are placed into the queue.
- 3) Worker servers process requests one-by-one.
- 4) The processed result is returned.
- 5) Long-running tasks are handled in the background.

* Examples:-

- Rabbit MQ
- Apache Kafka
- Amazon SQS
- Active MQ

MA

* Advantages

- Prevents Server Overload
- Handles high traffic efficiently.
- Improves reliability
- Better resource utilization
- Faster user response

Disadvantages

- Increased system complexity
- Delay in background task completion
- Queue management overhead
- Requires monitoring

* Applications:-

- Email Sending
- SMS delivery
- Payment Processing
- Order Processing
- Video processing
- Notification system

* Features of Queue:-

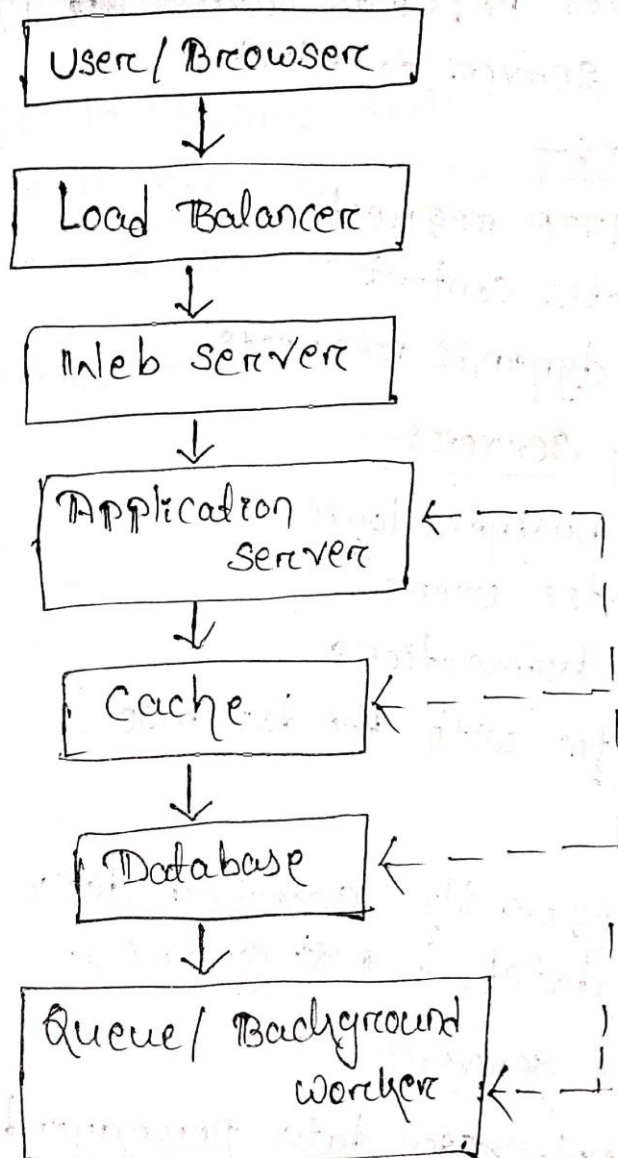
- FIFO (First-In-First-Out)
- Temporary storage
- Orderly Processing
- Reduces Server load
- Handles High Traffic

* Web Application Architecture:-

* Definition:-

- Web Application Architecture (WAA) is the complete structure of a web application that defines how the client, webserver, application server, database, cache, load balancer, and queues interact to provide fast, secure, reliable, and scalable web services.
- It is the blueprint of a web application.

* Diagram:-



* Components:-

1) Client (Browser):-

- Displays the user interface
- Sends HTTP requests
- Receives Server responses

2) Load Balancer:-

- Distributes requests across ~~not~~ multiple servers
- Prevents server overload.

3) Web Server:-

- Receives HTTP requests
- Serves static content
- Forwards dynamic requests

4) Application Server:-

- Executes business logic
- Authenticates users
- Processes transactions
- Communicates with the database

5) Cache:-

- Stores frequently accessed data
- Reduces database ~~and~~ queries.

6) Database Server:-

- Stores application data permanently.
- Performs CRUD operations (Create, Read, update delete).

7) Queue:-

- Processes background tasks asynchronously.
- Handles emails, notifications, invoices, & reports.

* Working:-

- 1) The user sends a request through the browser.
- 2) The load balancer selects an available web server.
- 3) Web server receives the request.
- 4) Request is forwarded to the application server.
- 5) Application server checks the cache.
- 6) If data is available in the cache, it is returned immediately.
- 7) Otherwise, the application server accesses the database.
- 8) Background tasks are sent to the queue.
- 9) The application server prepares the response.
- 10) The web server returns the response to the browser.
- 11) The browser displays the result.

* Advantages

- High Performance
- Better Scalability
- Improved Security
- High availability
- Easy maintenance
- Fault tolerance
- Efficient resource utilization
- Supports millions of users.

Disadvantages

- Complex architecture
- Higher infrastructure cost.
- Requires skilled administration.
- Requires continuous monitoring.
- More deployment and configuration efforts.

* Features:-

- Layered Architecture
- Client - Server Communication
- Scalability
- High Performance
- Security

* Uses:-

- Used to design and develop web-based applications.
- Online transactions and Payment systems.
- Used to manage large volumes of user requests.
- Data storage and retrieval through databases.
- Used for user authentication and authorization.
- Used to provide fast and secure access to web services.

* Applications:-

- E-Commerce websites (Amazon, Flipkart)
- Online banking systems.
- Hospital management systems.
- Library management systems.
- Online learning platforms.

* Examples:-

- | | | |
|------------|------------|---------|
| → Amazon | → Flipkart | → Gmail |
| → Facebook | → YouTube | → Paytm |

* Conclusion:-

→ Web Systems Architecture deals with the entire web system and infrastructure, whereas Web Application Architecture deals with the internal design and working of a specific web application.

* Difference Between Web Systems Architecture and Web Application Architecture.

Web Systems Architecture (WSA)

Web Application Architecture (WAA)

1) It is the overall structure of a web-based system.

2) It includes network, servers, databases, and communication components.

3) It focuses on the entire web infrastructure.

4) Manages communication between different system components.

5) Includes components such as cache, proxy, load balancer and queues.

6) It is used for designing large web systems.

7) It provides system-level architecture.

8) Example:- Overall architecture of an e-commerce platform.

1) It is the structure of a web application.

2) It includes client, web-server, application server, and database.

3) It focuses on how a web application works.

4) Manages the flow of user requests and responses.

5) It mainly includes presentation layer, business logic layer, and data layer.

6) It is used for developing and deploying web applications.

7) It provides application-level architecture.

8) Example:- Architecture of the Amazon shopping application.