



STUDY MATERIAL
On
WEB TECHNOLOGIES
(For 5th Semester CSE)
SESSION(2026-27)

Prepared by :

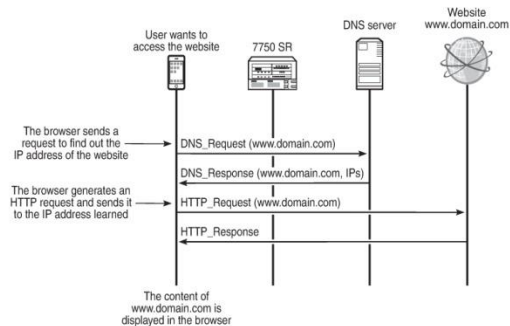
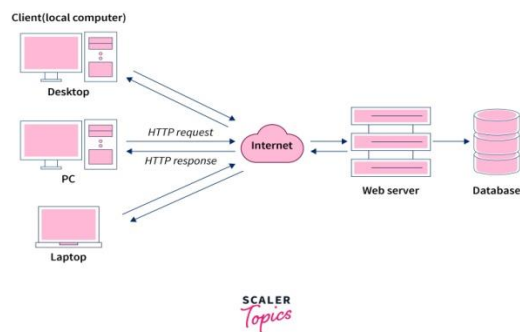
SURAJ KUMAR BHUJABAL (CSE DEPT),
GIET (POLY),JAGATPUR,CUTTACK

Introduction to World Wide Web (WWW):

The **World Wide Web (WWW)** is a global system of **interconnected web pages and resources** that are accessed through the **Internet** using web browsers. It was developed by **Tim Berners-Lee in 1989**.

- **Internet** is the physical network (cables, routers, computers)
- **WWW** is a service that runs on the Internet and provides information in the form of web pages

The WWW works on a **client-server model**, where the browser (client) requests information and the server provides it.



27196

Protocols Used in WWW:

Protocols are rules that control communication between client and server.

1. HTTP (HyperText Transfer Protocol)

- Main protocol of WWW
- Used to transfer web pages
- Stateless (each request is independent)

2. HTTPS (HyperText Transfer Protocol Secure)

- Secure version of HTTP
- Uses encryption (SSL/TLS)
- Protects sensitive data like passwords and banking info

3. DNS (Domain Name System)

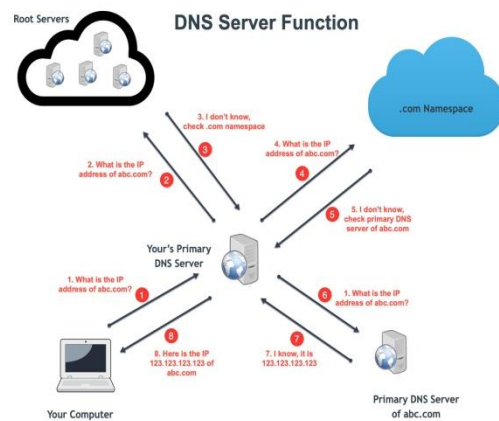
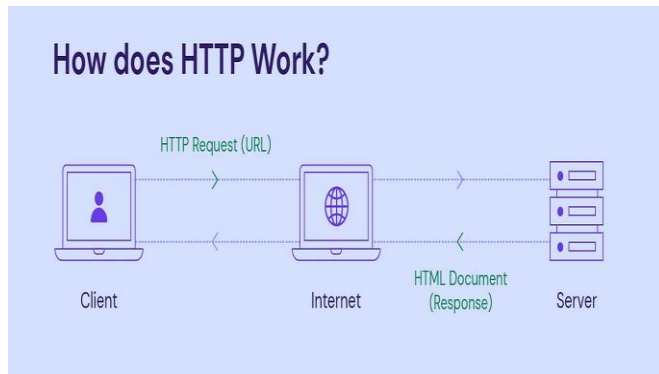
- Converts domain names into IP addresses
- Makes it easier to access websites

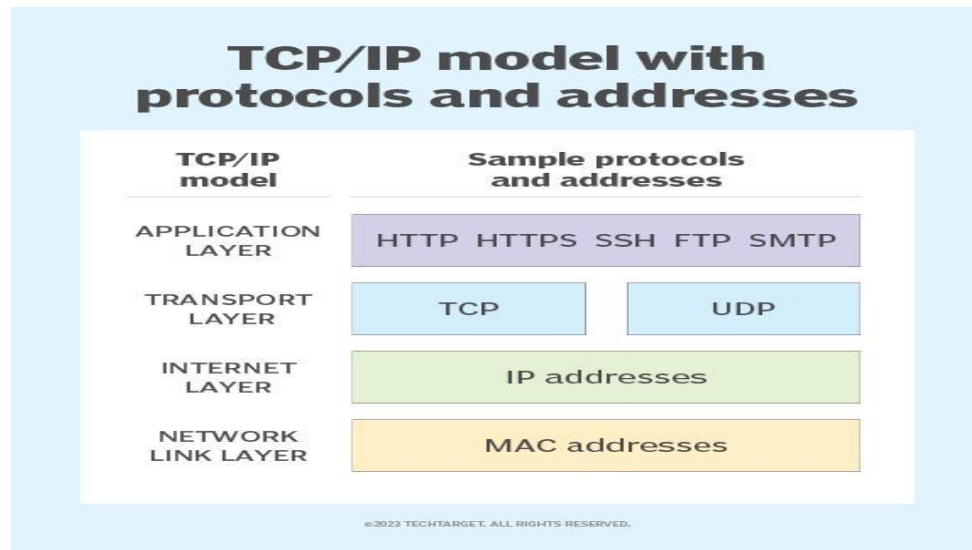
4. TCP/IP (Transmission Control Protocol / Internet Protocol)

- Basic communication protocol of the Internet
- Ensures reliable and accurate data transfer

5. FTP (File Transfer Protocol)

- Used for uploading and downloading files between computers





Programs Used in WWW:

1. Web Browsers

- Used by users to access websites
- Interprets HTML, CSS, and JavaScript

2. Web Servers

- Software that stores and delivers web pages
- Examples: Apache, Nginx

3. HTML (HyperText Markup Language)

- Provides structure of web pages
- Defines text, images, links

4. CSS (Cascading Style Sheets)

- Used for styling web pages
- Controls colors, layout, fonts

5. JavaScript

- Adds interactivity and dynamic behavior
- Used for animations, forms, validations

Secure Connections:

Browsers must ensure that data travelling across the internet cannot be intercepted, read, or modified by unauthorized parties. They achieve this through specific layers of encryption and verification:

HTTPS (Hyper Text Transfer Protocol Secure): It encrypts all data sent between a browser and a web server, shielding sensitive credentials and financial data.

TLS (Transport Layer Security): The modern protocol that succeeds SSL to manage the cryptographic handshake and data encryption keys.

Public Key Cryptography: Browsers use asymmetric encryption to safely exchange unique session keys with a server upon initialization.

Strict Transport Security (HSTS): A web header ensuring the browser *only* connects to a specific site via an encrypted HTTPS connection.

Certificate Transparency: An open logging framework that helps the browser verify that a website's security certificate was legitimately issued

Application & Development Tools:

Application and development tools in the context of the World Wide Web refer to the various software technologies and platforms that are used to design, create, test, and maintain websites and web applications. These tools play a very important role in transforming ideas into functional and interactive web pages that users can access through browsers.

At the basic level, web development is divided into two main parts: **front-end (client-side)** and **back-end (server-side)** development. Front-end tools are responsible for everything that users see and interact with on a website. Technologies like HTML, CSS, and JavaScript are used to build the structure, design, and behavior of web pages. HTML provides the basic framework of a webpage, CSS enhances its appearance by adding colors, layouts, and styles, while JavaScript makes the webpage interactive by enabling features such as forms, animations, and dynamic content.

On the other hand, back-end development tools handle the internal working of a website. These tools manage data processing, server communication, and database operations. Programming languages such as Python, PHP, and Java are commonly used to develop server-side logic. Databases like MySQL or MongoDB are used to store and retrieve data efficiently. In addition, frameworks such as Django, React, and Angular simplify the development process by providing ready-made

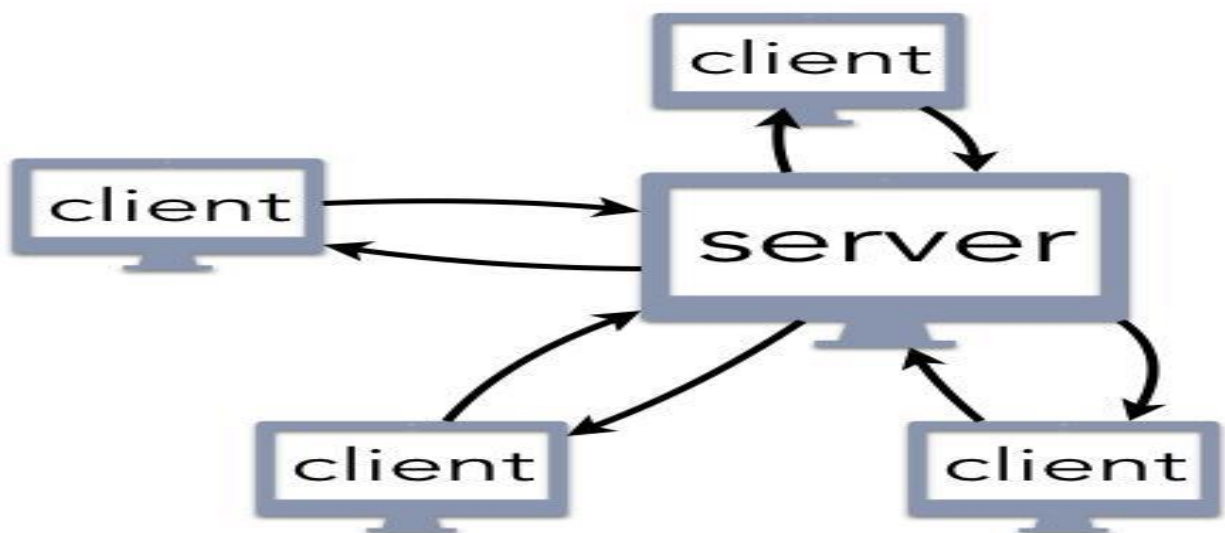
components and structured coding methods, which help developers build applications faster and more efficiently.

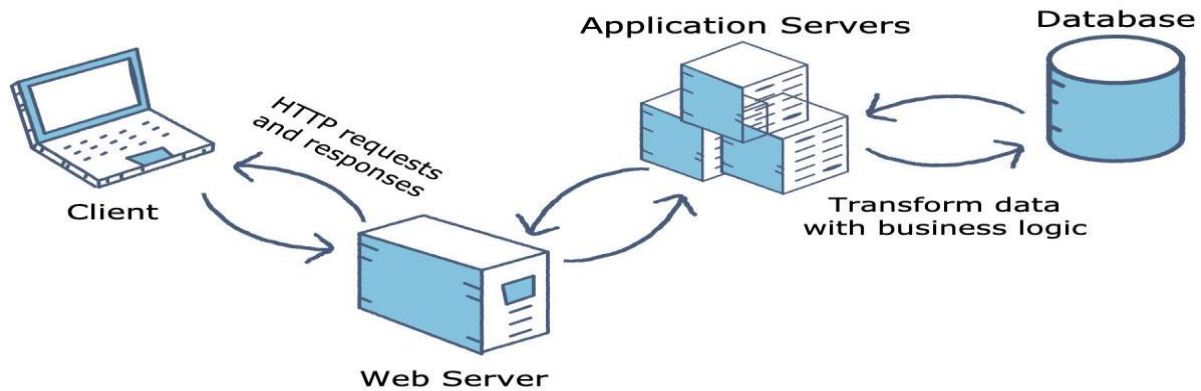
Apart from these, there are several **development tools** that assist programmers during the development process. Code editors and Integrated Development Environments (IDEs), such as Visual Studio Code, provide a platform for writing and editing code. Browser developer tools help in testing, debugging, and analyzing web pages directly within the browser. Version control systems like Git are used to track changes in code and support collaboration among multiple developers working on the same project.

In conclusion, application and development tools are essential for building modern websites and web applications. They not only make the development process easier and faster but also improve the quality, performance, and reliability of web systems. Together, these tools form the backbone of web development and play a crucial role in the functioning of the World Wide Web.

Server:

A **server** is a computer or software system that provides services, data, or resources to other computers known as clients over a network. In the context of the World Wide Web, a web server stores websites and delivers web pages to users when requested through a browser. When a user enters a URL, the browser sends a request to the server, and the server processes this request and sends back the required web page. Servers are designed to handle multiple requests at the same time and ensure smooth communication between users and web resources.





Setting up UNIX and LINUX Web Servers:

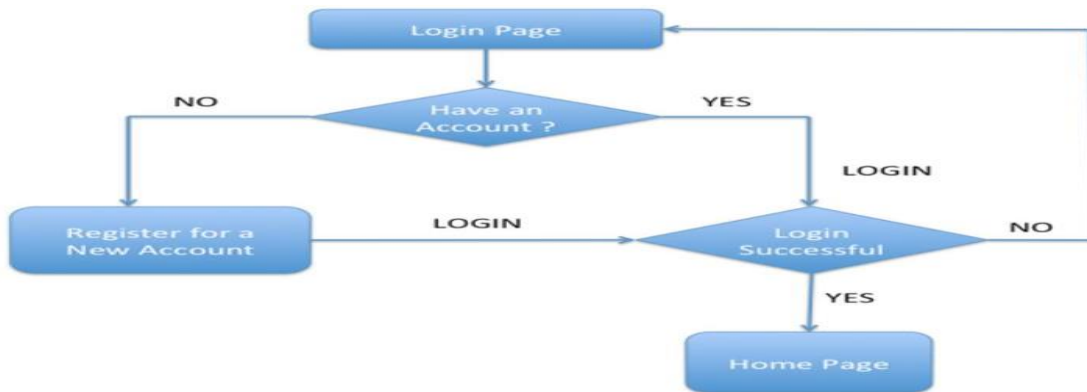
UNIX and LINUX are popular operating systems widely used for hosting web servers due to their stability, security, and performance. Setting up a web server in these systems involves installing server software such as Apache or Nginx, configuring system settings, and managing files and permissions.

The process generally begins with installing the operating system, followed by installing web server software using package managers. After installation, configuration files are edited to define how the server should respond to requests. Administrators also set permissions, manage directories, and ensure proper security settings such as firewalls. Once configured, the server can host websites and respond to client requests efficiently. These systems are preferred in real-world applications because they are open-source, reliable, and capable of handling large-scale web services.

Logging Users:

Logging users refers to the process of **authenticating and recording user activity** on a website or server. When a user enters a username and password, the system verifies the credentials using stored data in a database. If the information is correct, access is granted, and a session is created for that user.

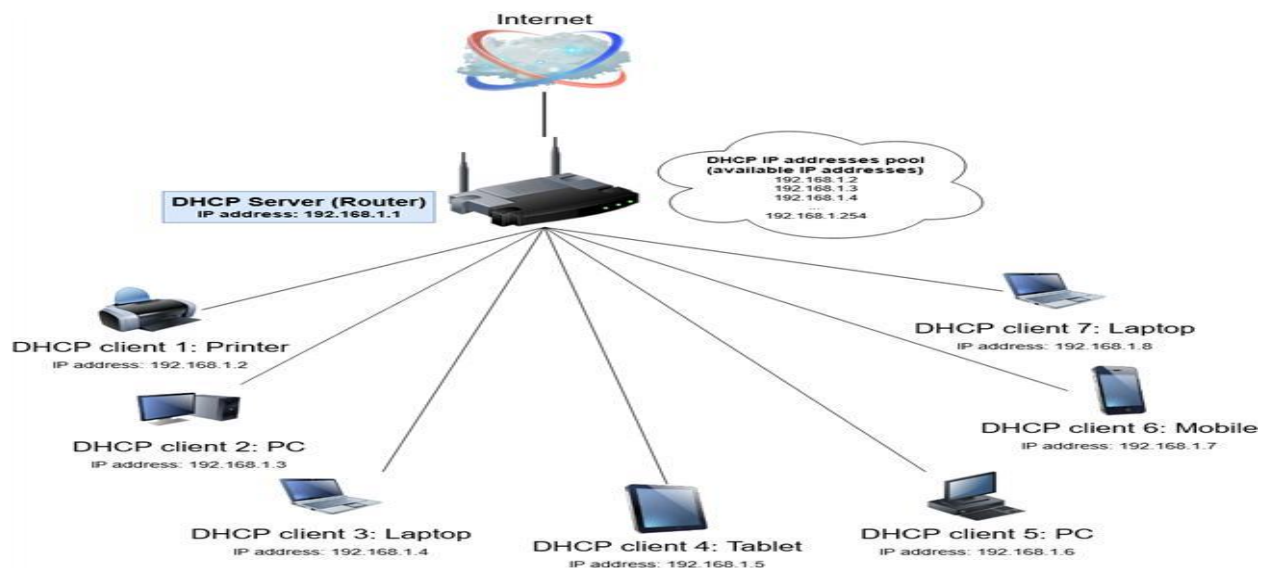
In addition to authentication, systems often maintain logs that record user actions such as login time, IP address, and activities performed. This helps in monitoring usage, improving security, and detecting unauthorized access. Logging is essential for maintaining accountability and ensuring safe access to web resources.



Dynamic IP Address:

A **dynamic IP address** is an IP address that is automatically assigned to a device by a network, usually through a **DHCP (Dynamic Host Configuration Protocol)** server. Unlike a static IP address, which remains fixed, a dynamic IP can change over time whenever the device reconnects to the network.

Dynamic IP addresses are commonly used in home and office networks because they are easy to manage and reduce administrative work. They allow efficient use of available IP addresses and provide a level of security since the address changes periodically. However, they may not be suitable for servers that require a permanent address.



Web Design: Website Design Principles & Navigation:

Web design refers to the process of planning and creating the visual layout and structure of a website. It focuses on both appearance and usability to provide a good user experience.

Website Design Principles

Good website design follows important principles such as simplicity, consistency, balance, and clarity. A well-designed website should be easy to understand, visually appealing, and responsive across different devices. Proper use of colors, fonts, images, and spacing improves readability and user engagement. Designers must also ensure fast loading speed and accessibility for all users.

Planning the Site

Planning is a crucial step before designing a website. It involves identifying the purpose of the website, target audience, and type of content to be included. A clear structure is created using a sitemap, which shows how different pages are connected. Proper planning ensures that the website meets user needs and functions effectively.

Navigation

Navigation refers to how users move from one page to another within a website. It includes menus, links, and buttons that guide users. Good navigation should be simple, consistent, and easy to use so that users can quickly find the information they need. Poor navigation can confuse users and reduce the effectiveness of the website.

Web Systems Architecture:

Architecture of Web-Based Systems:

Web systems architecture refers to the overall structure and organization of components that make a web application work. It defines how clients, servers, databases, and other elements interact to deliver web services efficiently. The main goal of architecture is to ensure **performance, scalability, and reliability**.

Client/Server (2-Tier) Architecture

In a **2-tier architecture**, the system is divided into two parts: the **client** and the **server**. The client is usually the web browser that sends requests, while the server processes these requests and sends back responses.

In this model, the client directly communicates with the server, which may also handle database operations. It is simple to design and easy to implement, making it suitable for small applications. However, as the number of users increases, this architecture may face performance issues because the server has to handle both application logic and data processing.

3-Tier Architecture

The **3-tier architecture** divides the system into three layers:

1. **Presentation Layer** (Client) – User interface (browser)

2. **Application Layer** (Server) – Business logic processing
3. **Data Layer** (Database) – Data storage and management

This separation improves scalability, security, and maintainability. Each layer can be developed and managed independently, allowing better performance for large-scale applications. It is widely used in modern web applications.

Building Blocks of Fast & Scalable Data Access:

To make web systems fast and scalable, several important components are used:

Caches

Caches store frequently accessed data temporarily so that it can be retrieved quickly without repeatedly accessing the database. This reduces response time and improves performance.

Proxies

A proxy server acts as an intermediary between the client and server. It can filter requests, improve security, and cache content to speed up access.

Indexes

Indexes are used in databases to quickly locate data without scanning the entire database. They improve query performance significantly.

Load Balancers

Load balancers distribute incoming network traffic across multiple servers. This prevents any single server from being overloaded and ensures high availability.

Queues

Queues store tasks or requests temporarily and process them in order. They help manage heavy workloads and ensure smooth processing of background tasks.

Web Application Architecture (WAA):

Web Application Architecture (WAA) is a framework that defines how different components of a web application interact with each other. It includes the **frontend (user interface)**, **backend (server logic)**, and **database (data storage)**.

When a user interacts with a web application, the request is sent from the browser to the server. The server processes the request, interacts with the database if needed, and sends the response

back to the client. Modern WAA may also include APIs, microservices, and cloud services to enhance flexibility and scalability.

A well-designed web application architecture ensures that the system is **efficient, secure, scalable, and easy to maintain**, which is essential for handling real-world applications with large numbers of users.

JavaScript:

Client-Side Scripting:

Client-side scripting refers to programs that run on the **user's browser (client side)** instead of the server. JavaScript is the most commonly used client-side scripting language. It helps make web pages **interactive, dynamic, and responsive** without sending requests to the server every time. For example, form validation, animations, and instant updates are done using client-side scripting.

What is JavaScript?

JavaScript is a **high-level, interpreted programming language** used to create interactive web pages. It works together with HTML and CSS to enhance user experience. While HTML provides structure and CSS provides design, JavaScript adds **behavior and functionality** such as responding to user actions, updating content, and controlling multimedia.

```
<!DOCTYPE html>

<html>

<body>

<h2>Hello Page</h2>

<button onclick="showMessage()">Click Me</button>

<script>

function showMessage() {

    alert("Hello, Welcome to JavaScript!");} </script></body></html>
```

Variables in JavaScript

A **variable** is used to store data values.

```
let name = "Rahul";  
let age = 20;  
const pi = 3.14;
```

- `let` → value can change
 - `const` → value cannot change
 - Used to store numbers, text, etc.
-

Functions in JavaScript

A **function** is a block of code used to perform a task.

```
function add(a, b) {  
    return a + b;  
}
```

Functions help in reusing code and organizing programs.

Conditions in JavaScript

Conditions are used to make decisions.

```
let marks = 60;  
  
if (marks >= 50) {  
    console.log("Pass");  
} else {  
    console.log("Fail");  
}
```

`if-else` checks conditions and executes code accordingly.

Loops and Repetition

Loops are used to repeat a block of code multiple times.

Example: for loop

```
for (let i = 1; i <= 5; i++) {  
    console.log(i);  
}
```

Example: while loop

```
let i = 1;
while (i <= 5) {
  console.log(i);
  i++;
}
```

Loops reduce repetition and save time.

Advanced Scripting & Related Technologies:

JavaScript and Objects:

JavaScript is an **object-based scripting language**, which means it uses objects to store and manage data. An object is a collection of properties (data) and methods (functions). For example, a student object can have properties like name and age, and methods like display details. JavaScript also provides built-in objects such as Array, Date, and Math, which help developers perform common tasks easily. Objects make programming more organized and reusable.

DOM and Web Browser Environment:

The **Document Object Model (DOM)** represents an HTML document as a tree structure where each element is an object. JavaScript can access and modify these elements dynamically, allowing changes in content, style, and structure without reloading the page. The web browser environment provides objects like `window`, `document`, and `navigator` that allow JavaScript to interact with the browser and user.

Forms and Validation:

Forms are used to collect user input on websites. JavaScript is used to **validate form data** before sending it to the server. Validation ensures that the user enters correct and complete information, such as checking if an email is valid or if a password meets requirements. This improves user experience and reduces unnecessary server load.

DHTML (Dynamic HTML):

Dynamic HTML (DHTML) is not a separate language but a combination of **HTML, CSS, and JavaScript** used to create dynamic and interactive web pages. It allows content to change without reloading the page, such as animations, dropdown menus, and dynamic content updates.

Events and Buttons:

Events are actions that occur in the browser, such as clicking a button, moving the mouse, or pressing a key. JavaScript can respond to these events using event handlers like `onclick`, `onmouseover`, and `onkeydown`. Buttons are commonly used elements that trigger these events to perform actions.

Controlling Your Browser:

JavaScript can control browser behavior using the **window object**. It can open new windows, display alerts, navigate between pages, and manage browser history. This allows developers to create interactive and user-friendly applications.

AJAX (Asynchronous JavaScript and XML):

AJAX is a technique that allows web pages to **send and receive data from the server without reloading the entire page**. It improves speed and user experience by updating only parts of the webpage.

Advantages:

- Faster and more responsive
- Reduces server load
- Improves user experience

Disadvantages:

- More complex to implement
- Security concerns
- Depends on JavaScript support

AJAX-based Applications:

Examples include Google Maps, Gmail, and live search features.

Alternatives:

- Fetch API
- WebSockets
- Server-Sent Events (SSE)

XML, XSL, and XSLT:

Introduction to XML

XML (eXtensible Markup Language) is used to **store and transport data** in a structured format.

Uses of XML

It is used in data sharing, web services, configuration files, and document storage.

Simple XML Example

```
<student>
  <name>Rahul</name>
  <age>20</age>
</student>
```

Key Components

- Elements
- Attributes
- Tags

DTD and Schemas

DTD and XML Schema define the structure and rules of XML documents to ensure correctness.

XML with Applications

XML is widely used in web applications for data exchange between systems.

XSL and XSLT

- **XSL:** Used for styling XML documents
- **XSLT:** Transforms XML into other formats like HTML

Introduction to Web Services:

Web services are software systems that allow different applications to **communicate over the Internet**. They use standard protocols like HTTP and

data formats like XML or JSON. Examples include REST APIs and SOAP services. Web services enable data sharing and integration between different systems and platforms.

PHP (Server-Side Scripting):

Server-Side Scripting with PHP:

PHP (Hypertext Preprocessor) is a **server-side scripting language** used to develop dynamic web pages. Unlike JavaScript, which runs in the browser, PHP runs on the **web server**. When a user requests a webpage, the server processes the PHP code and sends only the output (HTML) to the browser. This allows secure data processing, database interaction, and dynamic content generation.

Arrays, Functions, and Forms in PHP

In PHP, an **array** is used to store multiple values in a single variable. Arrays can be indexed (numeric keys) or associative (named keys). For example, an array can store a list of student names or marks. Arrays make data handling easier and more organized.

A **function** in PHP is a block of reusable code designed to perform a specific task. Functions help reduce repetition and improve code readability. For example, a function can be written to calculate the sum of two numbers or to process user input.

Forms are an essential part of web development. PHP is commonly used to **handle form data** sent through HTML forms. When a user submits a form, PHP collects the data using methods like `GET` or `POST`, processes it, and returns a response. This is widely used in login systems, registration forms, and feedback pages.

Advanced PHP with Databases:

PHP is strongly integrated with databases like MySQL, allowing developers to store, retrieve, and manipulate data efficiently.

Connection to Server

To work with a database, PHP first connects to the MySQL server using functions like `mysqli_connect()`.

```
<?php

$conn = mysqli_connect("localhost", "root", "", "test");

if (!$conn) {

    echo "Connection failed";

}

?>
```

Creating and Selecting a Database

A database can be created and selected using SQL commands in PHP:

```
mysqli_query($conn, "CREATE DATABASE mydb");

mysqli_select_db($conn, "mydb");
```

Listing Databases and Tables

PHP can display available databases and tables using queries like:

```
SHOW DATABASES;
```

```
SHOW TABLES;
```

Creating a Table

Tables are created to store structured data:

```
CREATE TABLE students (

    id INT,

    name VARCHAR(50),

    age INT
```

);

Inserting Data

Data can be inserted into tables:

```
INSERT INTO students (id, name, age)
```

```
VALUES (1, 'Rahul', 20);
```

Altering Tables

Tables can be modified using:

```
ALTER TABLE students ADD email VARCHAR(50);
```

Queries (Fetching Data)

To retrieve data:

```
SELECT * FROM students;
```

Deleting Database, Tables, and Data

- Delete Database:

```
DROP DATABASE mydb;
```

Delete Table:

```
DROP TABLE students;
```

Delete Data:

```
DELETE FROM students WHERE id=1;
```

phpMyAdmin and Database Bugs:

phpMyAdmin is a web-based tool used to manage MySQL databases easily. It provides a graphical interface to create databases, tables, run queries, and manage data without writing SQL commands manually.

Database bugs are common issues that occur during development, such as connection errors, incorrect queries, or data mismatch. These can be solved by checking error messages, validating queries, and ensuring proper database configuration.

Introduction to Modern Web Frameworks:

Modern web development uses powerful frameworks and libraries to make websites **faster, responsive, and easier to build**. These frameworks reduce coding effort and help developers create professional applications efficiently.

React:

React is a popular **JavaScript library** used for building user interfaces, especially single-page applications. It is based on the concept of **components**, where the UI is divided into small reusable parts. React uses a **Virtual DOM**, which improves performance by updating only the changed parts of a webpage instead of reloading the entire page. It is widely used for fast and interactive web applications.

Angular:

Angular is a **full-featured front-end framework** developed by Google. It follows a structured approach using components, modules, and services. Angular supports **two-way data binding**, meaning changes in the UI reflect in the data and vice versa. It is used to build large-scale and complex web applications with strong architecture.

Node.js:

Node.js is a **server-side runtime environment** that allows JavaScript to run on the server. It is built on a non-blocking, event-driven architecture, making it highly efficient and scalable. Node.js is commonly used to build backend services, APIs, and real-time applications like chat systems.

Bootstrap:

Bootstrap is a popular **CSS framework** used for designing responsive and mobile-friendly websites. It provides pre-designed components like buttons, forms, navigation bars, and grids. Developers can quickly create attractive layouts without writing much CSS code.

Tailwind CSS:

Tailwind CSS is a **utility-first CSS framework** that allows developers to design directly in HTML using small utility classes. Instead of writing custom CSS, developers apply ready-made classes for styling. It offers flexibility and helps in building modern and customized designs quickly.