

**SUDDHANANDA SCHOOL MANAGEMENT & COMPUTER  
SCIENCE**

**LECTURE NOTES ON  
INTERNET AND WEB TECHNOLOGY  
(MCPE2004)**

**UNIT – I**

**INTRODUCTION TO INTERNET AND WEB TECHNOLOGY:**

**INTRODUCTION TO INTERNET**

The Internet is a worldwide network of interconnected computers and devices that communicate with each other using standard communication protocols.

It allows users to share information, communicate, access services, and exchange data globally.

Today, the Internet has become an essential part of education, business, entertainment, banking, and communication.

**DEFINITION OF INTERNET**

The Internet is a global system of interconnected computer networks that uses the TCP/IP protocol suite to connect devices worldwide.

**HISTORY OF INTERNET**

**1969 – ARPANET**

The first computer network developed by the U.S. Department of Defense.

**1970s**

Development of TCP/IP protocols.

**1983**

TCP/IP became the standard communication protocol.

**1990**

World Wide Web (WWW) was introduced by **Tim Berners-Lee**.

**Present**

Billions of users worldwide use the Internet daily.

**FEATURES OF INTERNET**

1. Global Connectivity
2. Information Sharing
3. Fast Communication

4. Resource Sharing
5. Electronic Commerce
6. Online Learning
7. Social Networking

### **ADVANTAGES OF INTERNET**

#### **1. Communication**

Email, video conferencing, and instant messaging.

#### **2. Information Access**

Easy access to knowledge and educational resources.

#### **3. Online Banking**

Digital transactions and fund transfers.

#### **4. E-Commerce**

Online buying and selling of products.

#### **5. Entertainment**

Movies, music, games, and streaming services.

### **DISADVANTAGES OF INTERNET**

1. Cybercrime
2. Data Theft
3. Privacy Issues
4. Internet Addiction
5. Malware Attacks

### **INTERNET SERVICES**

The Internet provides various services:

#### **1. E-Mail**

Electronic mail used for communication.

Example:

- Gmail
- Yahoo Mail

#### **2. World Wide Web (WWW)**

Collection of interconnected web pages.

#### **3. File Transfer Protocol (FTP)**

Used for transferring files between computers.

#### **4. Remote Login**

Allows users to access remote systems.

#### **5. Video Conferencing**

Online meetings and communication.

Examples:

- Zoom
- Google Meet

### **WORLD WIDE WEB (WWW)**

The World Wide Web is a collection of web pages connected through hyperlinks and accessed using web browsers.

WWW is one of the most popular services available on the Internet.

### **COMPONENTS OF WWW**

#### **1. Web Browser**

Software used to access websites.

Examples:

- Google Chrome
- Mozilla Firefox
- Microsoft Edge

#### **2. Web Server**

Stores and delivers web pages.

Examples:

- Apache Server
- Nginx

#### **3. Web Pages**

Documents displayed in browsers.

Created using:

- HTML
- CSS
- JavaScript

### **WEB TECHNOLOGY**

Web Technology refers to the technologies used for communication between web servers and web browsers.

It includes:

- HTML
- CSS

- JavaScript
- XML
- PHP
- ASP.NET

## **CLIENT-SERVER ARCHITECTURE**

Web applications follow the Client-Server Architecture.

### **CLIENT**

The client is the user's computer or browser that requests services.

Examples:

- Chrome Browser
- Firefox Browser

### **SERVER**

The server provides requested resources and services.

Examples:

- Apache Server
- Nginx Server

## **CLIENT-SERVER MODEL**

Client (Browser)

↓ Request

Internet

↓

Server

↓ Response

Client

## **ADVANTAGES OF CLIENT-SERVER MODEL**

1. Centralized management.
2. Better security.
3. Easy maintenance.
4. Resource sharing.

## WEB BROWSER

A Web Browser is software used to access and display web pages.

### FUNCTIONS OF WEB BROWSER

1. Request web pages.
2. Interpret HTML documents.
3. Display multimedia content.
4. Execute JavaScript code.

### COMMON WEB BROWSERS

Browser	Developer
Google Chrome	Google
Firefox	Mozilla
Edge	Microsoft
Safari	Apple

## WEB SERVER

A Web Server stores websites and delivers web content to users.

### FUNCTIONS OF WEB SERVER

1. Receive requests.
2. Process requests.
3. Send responses.
4. Manage web resources.

### URL (UNIFORM RESOURCE LOCATOR)

A URL is the address of a web resource on the Internet.

### EXAMPLE

<https://www.google.com>

### COMPONENTS OF URL

<https://www.example.com/index.html>

**Protocol**

https

**Domain Name**

www.example.com

**Web Page**

index.html

**DOMAIN NAME SYSTEM (DNS)**

DNS converts domain names into IP addresses.

**EXAMPLE**

User enters:

www.google.com

DNS converts it into:

142.250.xxx.xxx

so computers can communicate.

**ADVANTAGES OF DNS**

1. Easy website access.
2. Human-readable names.
3. Efficient resource location.

**IP ADDRESS**

Every device connected to the Internet has a unique IP Address.

**TYPES OF IP ADDRESS****IPv4**

32-bit address.

Example:

192.168.1.1

**IPv6**

128-bit address.

Example:

2001:db8::1

**TCP/IP PROTOCOL SUITE**

TCP/IP is the standard protocol suite used on the Internet.

### **TCP (TRANSMISSION CONTROL PROTOCOL)**

TCP provides:

- Reliable communication
- Error detection
- Ordered delivery

### **IP (INTERNET PROTOCOL)**

IP provides:

- Addressing
- Routing

### **TCP/IP LAYERS**

Application Layer

Transport Layer

Internet Layer

Network Access Layer

### **HTTP (HYPERTEXT TRANSFER PROTOCOL)**

HTTP is the communication protocol used between browsers and web servers.

### **WORKING OF HTTP**

Browser Request



Web Server



HTTP Response



Browser Display

### **FEATURES OF HTTP**

1. Simple

2. Fast
3. Stateless Protocol

## **HTTPS**

HTTPS stands for:

### **HyperText Transfer Protocol Secure**

It is the secure version of HTTP.

### **ADVANTAGES OF HTTPS**

1. Data Encryption
2. Secure Communication
3. Data Integrity
4. Authentication

### **DIFFERENCE BETWEEN HTTP AND HTTPS**

<b>HTTP</b>	<b>HTTPS</b>
Not Secure	Secure
No Encryption	Uses Encryption
Port 80	Port 443
Less Secure	Highly Secure

### **SEARCH ENGINE**

A Search Engine helps users find information on the Internet.

Examples:

- Google
- Bing
- Yahoo

### **WORKING OF SEARCH ENGINE**

1. Crawling
2. Indexing

3. Searching
4. Ranking

## **INTERNET SECURITY THREATS**

### **1. Virus**

Malicious software that damages files.

### **2. Worm**

Self-replicating malware.

### **3. Trojan Horse**

Malicious program disguised as legitimate software.

### **4. Phishing**

Fraudulent attempts to steal user information.

## **FIREWALL**

A Firewall is a security system that monitors and controls network traffic.

## **FUNCTIONS OF FIREWALL**

1. Blocks unauthorized access.
2. Protects systems.
3. Filters network traffic.

## **APPLICATIONS OF INTERNET AND WEB TECHNOLOGY**

1. E-Commerce
2. E-Learning
3. Online Banking
4. Social Media
5. Cloud Computing
6. Digital Marketing
7. Telemedicine

PROF PRITAM NANDA

## UNIT – II

### HTML, HTML5, WEB PAGE DESIGN, HYPERLINKS, LISTS, TABLES, FORMS, FRAMES, MULTIMEDIA AND CSS:

#### **INTRODUCTION TO HTML**

HTML stands for **HyperText Markup Language**.

It is the standard markup language used to create web pages and web applications.

HTML describes the structure of a web page using tags and elements.

A web browser reads HTML documents and displays them as web pages.

#### **FEATURES OF HTML**

1. Easy to learn and use.

2. Platform independent.
3. Supports hyperlinks.
4. Supports multimedia.
5. Compatible with all web browsers.
6. Forms the foundation of web development.

## **HISTORY OF HTML**

- Developed by **Tim Berners-Lee**
- First version released in 1991.
- HTML5 is the latest major version.

## **BASIC STRUCTURE OF HTML DOCUMENT**

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>My First Web Page</title>
```

```
</head>
```

```
<body>
```

```
<h1>Welcome to HTML</h1>
```

```
<p>This is my first web page.</p>
```

```
</body>
```

```
</html>
```

## **EXPLANATION OF HTML TAGS**

**<!DOCTYPE html>**

Specifies HTML5 document.

**<html>**

Root element of HTML page.

**<head>**

Contains metadata.

**<title>**

Displays page title.

## **<body>**

Contains visible content.

## **HTML ELEMENTS**

An HTML element consists of:

`<tagname>`

Content

`</tagname>`

Example:

`<p>Hello World</p>`

## **TYPES OF TAGS**

### **1. Container Tags**

Require opening and closing tags.

Example:

`<p>Hello</p>`

### **2. Empty Tags**

Do not require closing tags.

Example:

`<br>`

`<hr>`

## **HTML HEADING TAGS**

HTML provides six heading levels.

`<h1>Heading 1</h1>`

`<h2>Heading 2</h2>`

`<h3>Heading 3</h3>`

`<h4>Heading 4</h4>`

`<h5>Heading 5</h5>`

`<h6>Heading 6</h6>`

## **PARAGRAPH TAG**

Used to create paragraphs.

<p>This is a paragraph.</p>

### LINE BREAK TAG

<br>

Moves content to next line.

### HORIZONTAL RULE

<hr>

Creates a horizontal line.

### TEXT FORMATTING TAGS

#### Bold Text

<b>Bold Text</b>

Output:

**Bold Text**

#### Italic Text

<i>Italic Text</i>

Output:

*Italic Text*

#### Underline Text

<u>Underlined Text</u>

Output:

Underlined Text

#### Superscript

x<sup>2</sup>

Output:

x<sup>2</sup>

#### Subscript

H<sub>2</sub>O

Output:

H<sub>2</sub>O

## HTML ATTRIBUTES

Attributes provide additional information about elements.

Example:

```
<a href="https://www.google.com">
```

Google

```
</a>
```

Here:

href

is an attribute.

## HYPERLINKS IN HTML

Hyperlinks connect one web page to another.

Created using:

```
<a>
```

tag.

## EXAMPLE

```
<a href="https://www.google.com">
```

Visit Google

```
</a>
```

## TYPES OF LINKS

### 1. External Link

Links to another website.

### 2. Internal Link

Links to pages within same website.

### 3. Email Link

```
<a href="mailto:abc@gmail.com">
```

Send Mail

```
</a>
```

## HTML LISTS

Lists are used to display related items.

Types:

1. Ordered List
2. Unordered List
3. Description List

### **ORDERED LIST**

Displays numbered items.

```
<ol>
  <li>HTML</li>
  <li>CSS</li>
  <li>JavaScript</li>
</ol>
```

Output:

1. HTML
2. CSS
3. JavaScript

### **UNORDERED LIST**

Displays bullet points.

```
<ul>
  <li>Apple</li>
  <li>Mango</li>
</ul>
```

Output:

- Apple
- Mango

### **DESCRIPTION LIST**

```
<dl>
  <dt>HTML</dt>
  <dd>Markup Language</dd>
</dl>
```

## HTML TABLES

Tables organize data into rows and columns.

### TABLE STRUCTURE

```
<table border="1">
```

```
<tr>
```

```
  <th>Name</th>
```

```
  <th>Age</th>
```

```
</tr>
```

```
<tr>
```

```
  <td>Alisha</td>
```

```
  <td>22</td>
```

```
</tr>
```

```
</table>
```

### TABLE TAGS

Tag	Purpose
-----	---------

<table>	Creates table
---------	---------------

<tr>	Table row
------	-----------

<th>	Table heading
------	---------------

<td>	Table data
------	------------

### OUTPUT

Name	Age
------	-----

Alisha	22
--------	----

### HTML IMAGES

Images are inserted using:

```
<img>
```

tag.

## EXAMPLE

```

```

## HTML FORMS

Forms collect user information.

Examples:

- Registration Form
- Login Form
- Feedback Form

## FORM TAG

```
<form>
```

```
</form>
```

## FORM ELEMENTS

### Text Box

```
<input type="text">
```

### Password Box

```
<input type="password">
```

### Radio Button

```
<input type="radio">
```

### Check Box

```
<input type="checkbox">
```

### Submit Button

```
<input type="submit">
```

## REGISTRATION FORM EXAMPLE

<form>

Name:

<input type="text"><br>

Password:

<input type="password"><br>

Gender:

<input type="radio">Male

<input type="radio">Female

<br>

<input type="submit">

</form>

## HTML5

HTML5 is the latest version of HTML.

Introduced to support:

- Multimedia
- Graphics
- Mobile devices
- Better web applications

## FEATURES OF HTML5

1. Audio Support
2. Video Support
3. Canvas Graphics
4. Local Storage
5. Semantic Elements

## 6. Improved Forms

### **SEMANTIC ELEMENTS**

Semantic tags describe content meaning.

Examples:

<header>

<nav>

<section>

<article>

<footer>

### **BENEFITS OF SEMANTIC TAGS**

1. Better readability.
2. SEO friendly.
3. Easy maintenance.

### **HTML5 AUDIO**

Audio files can be embedded directly.

<audio controls>

<source src="song.mp3">

</audio>

### **HTML5 VIDEO**

Video files can be embedded.

<video width="400" controls>

<source src="movie.mp4">

</video>

### **CANVAS ELEMENT**

Used for graphics and drawings.

<canvas id="myCanvas">

</canvas>

Applications:

- Games
- Charts
- Animations

## **FRAMES IN HTML**

Frames divide browser window into multiple sections.

### **FRAMESET EXAMPLE**

```
<frameset cols="30%,70%">
```

```
<frame src="menu.html">
```

```
<frame src="home.html">
```

```
</frameset>
```

### **ADVANTAGES OF FRAMES**

1. Multiple pages displayed.
2. Easy navigation.

### **DISADVANTAGES OF FRAMES**

1. SEO issues.
2. Difficult maintenance.
3. Poor mobile support.

## **INTRODUCTION TO CSS**

CSS stands for:

### **Cascading Style Sheets**

Used to style HTML documents.

CSS controls:

- Colors

- Fonts
- Layout
- Spacing
- Animations

### **ADVANTAGES OF CSS**

1. Attractive web pages.
2. Reusable styles.
3. Reduced code.
4. Faster loading.

### **TYPES OF CSS**

#### **1. Inline CSS**

Applied directly inside tag.

```
<p style="color:red;">
```

Hello

```
</p>
```

#### **2. Internal CSS**

Written inside <style> tag.

```
<style>
```

```
p{
```

```
color:red;
```

```
}
```

```
</style>
```

#### **3. External CSS**

Stored in separate file.

```
<link rel="stylesheet"
```

```
href="style.css">
```

### **CSS SYNTAX**

selector

```
{  
property:value;  
}
```

Example:

```
h1  
{  
color:blue;  
}
```

## **CSS SELECTORS**

### **Element Selector**

```
p  
{  
color:red;  
}
```

### **Class Selector**

```
.note  
{  
color:green;  
}
```

HTML:

```
<p class="note">
```

Welcome

```
</p>
```

### **ID Selector**

```
#header  
{  
color:blue;  
}
```

HTML:

```
<h1 id="header">
```

Welcome

```
</h1>
```

## CSS PROPERTIES

### Color

```
color:red;
```

### Background Color

```
background-color:yellow;
```

### Font Size

```
font-size:20px;
```

### Text Alignment

```
text-align:center;
```

### Border

```
border:2px solid black;
```

## SIMPLE WEB PAGE EXAMPLE

```
<html>
```

```
<head>
```

```
<style>
```

```
body{
```

```
background-color:lightblue;
```

```
}
```

```
h1{
```

```
color:red;
```

```
text-align:center;
```

```
}
```

</style>

</head>

<body>

<h1>Welcome to MCA</h1>

<p>Internet and Web Technology</p>

</body>

</html>

## **WEB PAGE DESIGN PRINCIPLES**

1. Simplicity
2. Consistency
3. Easy Navigation
4. Fast Loading
5. Responsive Design
6. Attractive Layout

## **APPLICATIONS OF HTML AND CSS**

1. Website Development
2. E-Commerce Websites
3. Educational Portals
4. Online Banking
5. Social Media Platforms

## UNIT – III

### JAVASCRIPT, CLIENT-SIDE SCRIPTING, DOM, EVENTS, FORM VALIDATION, COOKIES AND DYNAMIC WEB PAGES:

#### **INTRODUCTION TO JAVASCRIPT**

JavaScript is a lightweight, interpreted, object-based scripting language used to make web pages interactive and dynamic.

It was developed by **Brendan Eich** in 1995 at **Netscape Communications**.

JavaScript works mainly on the client side (browser side) and is one of the core technologies of web development along with HTML and CSS.

#### **NEED FOR JAVASCRIPT**

HTML provides structure.

CSS provides presentation.

JavaScript provides behavior and interactivity.

Without JavaScript, web pages remain static.

#### **FEATURES OF JAVASCRIPT**

1. Lightweight scripting language.
2. Platform independent.
3. Object-oriented.
4. Event-driven.
5. Supports client-side validation.
6. Easy integration with HTML and CSS.
7. Used for dynamic web applications.

#### **ADVANTAGES OF JAVASCRIPT**

1. Fast execution.
2. Reduces server load.
3. Improves user experience.
4. Creates interactive websites.
5. Supports animations and effects.

#### **LIMITATIONS OF JAVASCRIPT**

1. Browser dependency.
2. Security restrictions.
3. Different browser behaviors.
4. Can be disabled by users.

#### **CLIENT-SIDE SCRIPTING**

Client-side scripting refers to scripts executed in the user's browser rather than on the server.

Examples:

- JavaScript
- VBScript (older technology)

#### **ADVANTAGES OF CLIENT-SIDE SCRIPTING**

1. Faster response.
2. Reduced server traffic.
3. Better user interaction.
4. Immediate validation.

#### **EMBEDDING JAVASCRIPT IN HTML**

JavaScript code is written inside the <script> tag.

##### **Example**

```
<html>
```

```
<head>
```

```
<script>
```

```
document.write("Welcome to JavaScript");
```

```
</script>
```

```
</head>
```

```
<body>
```

```
</body>
```

```
</html>
```

## **OUTPUT METHODS IN JAVASCRIPT**

### **document.write()**

Displays output on web page.

```
document.write("Hello World");
```

### **alert()**

Displays popup message.

```
alert("Welcome");
```

### **console.log()**

Displays output in browser console.

```
console.log("Hello");
```

## **VARIABLES IN JAVASCRIPT**

Variables store data values.

### **Syntax**

```
var age = 22;
```

or

```
let age = 22;
```

or

```
const age = 22;
```

## **DATA TYPES IN JAVASCRIPT**

### **Number**

```
let age = 22;
```

## String

```
let name = "Alisha";
```

## Boolean

```
let result = true;
```

## Array

```
let marks = [80,90,85];
```

## Object

```
let student =  
{  
  name:"Alisha",  
  age:22  
};
```

## OPERATORS IN JAVASCRIPT

### Arithmetic Operators

```
+  
-  
*  
/  
%
```

Example:

```
let c = a + b
```

### Relational Operators

```
>  
<  
>=  
<=  
==  
!=
```

### Logical Operators

```
&&
```

```
||
```

```
!
```

## **CONTROL STATEMENTS**

Control statements determine the flow of execution.

### **IF STATEMENT**

```
let age = 20;
```

```
if(age>=18)
```

```
{
```

```
document.write("Eligible");
```

```
}
```

### **IF-ELSE STATEMENT**

```
if(age>=18)
```

```
{
```

```
document.write("Adult");
```

```
}
```

```
else
```

```
{
```

```
document.write("Minor");
```

```
}
```

### **SWITCH STATEMENT**

```
let day = 2;
```

```
switch(day)
```

```
{
```

```
case 1:
```

```
document.write("Monday");
```

```
break;
```

```
case 2:
```

```
document.write("Tuesday");
```

```
break;
```

```
default:
```

```
document.write("Invalid");
```

```
}
```

## **LOOPING STATEMENTS**

Loops execute a block repeatedly.

### **FOR LOOP**

```
for(let i=1;i<=5;i++)
```

```
{
```

```
document.write(i);
```

```
}
```

### **WHILE LOOP**

```
let i=1;
```

```
while(i<=5)
```

```
{
```

```
document.write(i);
```

```
i++;
```

```
}
```

### **DO-WHILE LOOP**

```
let i=1;
```

```
do
```

```
{
```

```
document.write(i);
```

```
i++;
```

```
}
```

```
while(i<=5);
```

## FUNCTIONS IN JAVASCRIPT

Functions are reusable blocks of code.

### Syntax

```
function display()  
{  
  document.write("Welcome");  
}
```

Calling Function:

```
display();
```

## FUNCTION WITH PARAMETERS

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

```
document.write(add(10,20));
```

Output:

30

## EVENTS IN JAVASCRIPT

An event is an action performed by the user or browser.

Examples:

- Mouse Click
- Key Press
- Mouse Hover
- Page Load

## COMMON EVENTS

Event	Description
onclick	Mouse click
onload	Page loading

Event	Description
onmouseover	Mouse over
onmouseout	Mouse leave
onchange	Value change
onsubmit	Form submission

#### EXAMPLE OF onclick EVENT

```
<button onclick="show()">
```

Click Here

```
</button>
```

```
<script>
```

```
function show()
```

```
{
```

```
  alert("Button Clicked");
```

```
}
```

```
</script>
```

#### DOCUMENT OBJECT MODEL (DOM)

DOM stands for **Document Object Model**.

It represents an HTML document as a tree structure.

JavaScript uses DOM to access and modify webpage elements dynamically.

#### DOM STRUCTURE

Document

|

HTML

/ \

Head Body

|

H1

|

Text

## DOM METHODS

### getElementById()

Accesses element using ID.

```
<p id="demo">
```

Hello

```
</p>
```

```
document.getElementById("demo");
```

### innerHTML

Changes content.

```
document.getElementById("demo").innerHTML
```

```
=
```

```
"Welcome MCA";
```

## EXAMPLE OF DOM MANIPULATION

```
<p id="msg">
```

Hello

```
</p>
```

```
<button onclick="change()">
```

Change

```
</button>
```

```
<script>
```

```
function change()
```

```
{
```

```
document.getElementById("msg").innerHTML
```

```
=
```

```
"Welcome Student";
```

```
}
```

```
</script>
```

## FORM VALIDATION

Form validation ensures users enter correct data before submission.

Validation can be:

1. Client-side
2. Server-side

JavaScript mainly performs client-side validation.

## ADVANTAGES OF VALIDATION

1. Prevents invalid data.
2. Improves security.
3. Reduces server load.
4. Enhances user experience.

## EXAMPLE OF FORM VALIDATION

```
<form onsubmit="return validate()">
```

Name:

```
<input type="text" id="name">
```

```
<input type="submit">
```

```
</form>
```

```
<script>
```

```
function validate()
```

```
{
```

```
let n =
```

```
document.getElementById("name").value;
```

```
if(n=="")
```

```
{
```

```
alert("Name Required");
```

```
return false;
```

```
}
```

```
return true;
```

```
}
```

```
</script>
```

### **EMAIL VALIDATION**

```
let email =
```

```
"abc@gmail.com";
```

```
if(email.includes("@"))
```

```
{
```

```
alert("Valid");
```

```
}
```

```
else
```

```
{
```

```
alert("Invalid");
```

```
}
```

### **COOKIES**

Cookies are small text files stored on the user's computer by a website.

They store user-related information.

## USES OF COOKIES

1. User Authentication
2. Session Management
3. User Preferences
4. Shopping Cart Data

## CREATING A COOKIE

document.cookie

=

"username=Alisha";

## READING A COOKIE

document.cookie;

## DELETING A COOKIE

document.cookie =

"username=;

expires=Thu, 01 Jan 1970";

## ADVANTAGES OF COOKIES

1. Personalized browsing.
2. Session maintenance.
3. Better user experience.

## DISADVANTAGES OF COOKIES

1. Privacy concerns.
2. Security risks.
3. Limited storage.

## DYNAMIC WEB PAGES

A Dynamic Web Page changes its content automatically based on user interaction or data.

Dynamic pages use:

- HTML
- CSS

- JavaScript
- Databases

### STATIC VS DYNAMIC WEB PAGES

Static Page	Dynamic Page
-------------	--------------

Fixed Content	Changing Content
---------------	------------------

HTML Only	HTML + JavaScript
-----------	-------------------

No Interaction	Interactive
----------------	-------------

Simple	Advanced
--------	----------

### EXAMPLE OF DYNAMIC PAGE

```
<button onclick="showTime()">
```

Show Time

```
</button>
```

```
<p id="time"></p>
```

```
<script>
```

```
function showTime()  
{  
  document.getElementById("time")  
  .innerHTML  
  =  
  new Date();  
}
```

```
</script>
```

When button is clicked, current date and time appear dynamically.

### JAVASCRIPT OBJECTS

Objects store data and functions together.

Example:

```
let student =  
{  
  name:"Alisha",  
  roll:101,  
  
  display:function()  
  {  
    return this.name;  
  }  
};
```

### **JAVASCRIPT ARRAYS**

Arrays store multiple values.

```
let marks =  
[80,85,90];
```

Accessing elements:

```
marks[0];
```

Output:

80

### **ERROR HANDLING**

JavaScript provides error handling using:

```
try  
{  
}  
catch(error)  
{  
}
```

Example:

```
try  
{  
  a=b+c;  
}
```

```
catch(error)
{
alert("Error Occurred");
}
```

#### **APPLICATIONS OF JAVASCRIPT**

1. Form Validation
2. Online Banking Systems
3. E-Commerce Websites
4. Social Media Platforms
5. Interactive Games
6. Dynamic Web Applications
7. Real-Time Chat Applications

## **UNIT – IV**

**XML, DTD, XML SCHEMA, AJAX, WEB SERVICES, PHP BASICS, SERVER-SIDE SCRIPTING AND DATABASE CONNECTIVITY:**

### **INTRODUCTION TO XML**

XML stands for **eXtensible Markup Language**.

XML is a markup language used to store, transport, and exchange data between different applications.

Unlike HTML, XML focuses on storing and describing data rather than displaying it.

XML was developed by the **World Wide Web Consortium**.

### **FEATURES OF XML**

1. Platform independent.
2. User-defined tags.
3. Self-descriptive.
4. Easy data exchange.
5. Supports hierarchical structure.
6. Human-readable and machine-readable.

### **ADVANTAGES OF XML**

1. Simplifies data sharing.
2. Easy integration between systems.
3. Supports web services.
4. Flexible and extensible.

### **XML DOCUMENT STRUCTURE**

Example:

```
<?xml version="1.0"?>
```

```
<Student>
```

```
<Name>Alisha</Name>
```

```
<Roll>101</Roll>
```

```
<Course>MCA</Course>
```

```
</Student>
```

### **PARTS OF XML DOCUMENT**

### **XML Declaration**

```
<?xml version="1.0"?>
```

Specifies XML version.

### **Root Element**

```
<Student>
```

Top-level element.

### **Child Elements**

```
<Name>
```

```
<Roll>
```

```
<Course>
```

Contained inside root element.

### **XML RULES**

1. Every opening tag must have a closing tag.
2. Tags are case-sensitive.
3. Elements must be properly nested.
4. XML document must have one root element.
5. Attribute values must be enclosed in quotes.

### **WELL-FORMED XML DOCUMENT**

A document that follows all XML syntax rules.

Example:

```
<Student>
```

```
<Name>Alisha</Name>
```

```
</Student>
```

### **INVALID XML**

<Student>

<Name>Alisha

</Student>

Reason:

Closing tag missing.

### XML ATTRIBUTES

Attributes provide additional information.

Example:

<Student id="101">

<Name>Alisha</Name>

</Student>

Here:

id="101"

is an attribute.

### XML VS HTML

#### XML

Stores data

User-defined tags

Case-sensitive

Focuses on data

#### HTML

Displays data

Predefined tags

Not case-sensitive

Focuses on presentation

### DOCUMENT TYPE DEFINITION (DTD)

DTD stands for **Document Type Definition**.

It defines the legal structure of an XML document.

DTD specifies:

- Elements
- Attributes

- Relationships

### **PURPOSE OF DTD**

1. Validates XML documents.
2. Defines structure.
3. Ensures consistency.

### **DTD EXAMPLE**

```
<!DOCTYPE Student [
```

```
<!ELEMENT Student (Name,Roll)>
```

```
<!ELEMENT Name (#PCDATA)>
```

```
<!ELEMENT Roll (#PCDATA)>
```

```
]>
```

### **XML USING DTD**

```
<Student>
```

```
<Name>Alisha</Name>
```

```
<Roll>101</Roll>
```

```
</Student>
```

### **ADVANTAGES OF DTD**

1. Simple implementation.
2. Defines document structure.
3. Improves consistency.

### **LIMITATIONS OF DTD**

1. Limited data types.
2. Less flexible.

3. Not XML-based.

### **XML SCHEMA**

XML Schema is an advanced method of defining XML structure.

It is written using XML syntax.

File extension:

.xsd

### **ADVANTAGES OF XML SCHEMA OVER DTD**

1. Supports data types.
2. Written in XML.
3. More secure.
4. Extensible.
5. Better validation.

### **XML SCHEMA EXAMPLE**

```
<xs:element  
name="student"  
type="xs:string"/>
```

### **DATA TYPES IN XML SCHEMA**

#### **String**

xs:string

#### **Integer**

xs:integer

#### **Date**

xs:date

#### **Boolean**

xs:boolean

## DTD VS XML SCHEMA

DTD	XML Schema
Not XML based	XML based
Limited data types	Rich data types
Less powerful	More powerful
Difficult validation	Better validation

## AJAX

AJAX stands for:

**Asynchronous JavaScript and XML**

AJAX allows communication with a server without reloading the entire webpage.

## NEED FOR AJAX

Traditional Web Page:

Request



Server



Reload Entire Page

AJAX:

Request



Server



Update Specific Part

## FEATURES OF AJAX

1. Faster interaction.
2. No full page reload.
3. Better user experience.
4. Reduced bandwidth usage.

## COMPONENTS OF AJAX

1. HTML
2. CSS
3. JavaScript
4. DOM
5. XML/JSON
6. XMLHttpRequest

### **WORKING OF AJAX**

Browser



JavaScript



AJAX Request



Server



Response



Update Web Page

### **AJAX EXAMPLE**

```
var xhttp =  
new XMLHttpRequest();  
  
xhttp.onreadystatechange =  
function()  
{  
if(this.readyState==4 &&  
this.status==200)  
{  
document.getElementById("demo")  
.innerHTML=this.responseText;  
}  
};
```

```
xhttp.open("GET",  
"data.txt",  
true);
```

```
xhttp.send();
```

### **ADVANTAGES OF AJAX**

1. Faster applications.
2. Better responsiveness.
3. Reduced server load.
4. Improved user experience.

### **APPLICATIONS OF AJAX**

1. Gmail
2. Online Search Suggestions
3. Social Media Updates
4. Chat Applications

### **WEB SERVICES**

Web Services enable communication between different applications over a network.

They allow data exchange irrespective of programming language or platform.

### **FEATURES OF WEB SERVICES**

1. Platform independent.
2. Language independent.
3. Reusable.
4. Supports interoperability.

### **COMPONENTS OF WEB SERVICES**

#### **SOAP**

Simple Object Access Protocol

#### **WSDL**

Web Service Description Language

#### **UDDI**

## **TYPES OF WEB SERVICES**

### **1. SOAP Web Services**

Uses XML messages.

More secure and standardized.

### **2. RESTful Web Services**

Uses HTTP methods.

Popular in modern web development.

## **REST ARCHITECTURE**

REST stands for:

### **Representational State Transfer**

Uses:

GET

POST

PUT

DELETE

methods.

## **EXAMPLE OF REST API**

GET /students

Retrieve students.

POST /students

Add student.

## **ADVANTAGES OF REST**

1. Lightweight.
2. Fast.
3. Easy integration.
4. Widely used.

PHP stands for:

## **Hypertext Preprocessor**

PHP is a server-side scripting language used for web development.

### **FEATURES OF PHP**

1. Open source.
2. Platform independent.
3. Easy database connectivity.
4. Fast execution.
5. Supports dynamic web pages.

### **ADVANTAGES OF PHP**

1. Easy to learn.
2. Free to use.
3. Large community support.
4. Supports multiple databases.

### **BASIC PHP SYNTAX**

```
<?php
```

```
echo "Welcome to PHP";
```

```
?>
```

Output:

Welcome to PHP

### **VARIABLES IN PHP**

Variables start with:

\$

Example:

```
<?php
```

```
$name="Alisha";
```

```
echo $name;
```

```
?>
```

## PHP DATA TYPES

1. Integer
2. Float
3. String
4. Boolean
5. Array
6. Object

## PHP OPERATORS

### Arithmetic Operators

+

-

\*

/

%

### Relational Operators

==

!=

>

<

## CONDITIONAL STATEMENTS IN PHP

### IF STATEMENT

```
<?php
```

```
$age=20;
```

```
if($age>=18)
{
echo "Adult";
}
```

```
?>
```

## **LOOPS IN PHP**

### **FOR LOOP**

```
<?php
```

```
for($i=1;$i<=5;$i++)
{
echo $i;
}
```

```
?>
```

### **WHILE LOOP**

```
<?php
```

```
$i=1;
```

```
while($i<=5)
{
echo $i;
$i++;
}
```

```
?>
```

### **FUNCTIONS IN PHP**

```
<?php
```

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

```
echo add(10,20);
```

```
?>
```

Output:

30

### **SERVER-SIDE SCRIPTING**

Server-side scripting means code executes on the server before sending results to the client.

Examples:

- PHP
- ASP.NET
- JSP
- Node.js

### **ADVANTAGES OF SERVER-SIDE SCRIPTING**

1. Better security.
2. Database access.
3. Dynamic content generation.
4. Authentication support.

### **DATABASE CONNECTIVITY IN PHP**

PHP commonly connects with MySQL databases.

### **STEPS OF DATABASE CONNECTIVITY**

1. Create Connection.
2. Select Database.
3. Execute Query.
4. Display Result.
5. Close Connection.

## PHP MYSQL CONNECTION

```
<?php
```

```
$conn =
```

```
mysqli_connect(
```

```
"localhost",
```

```
"root",
```

```
"",
```

```
"studentdb"
```

```
);
```

```
if(!$conn)
```

```
{
```

```
die("Connection Failed");
```

```
}
```

```
echo "Connected Successfully";
```

```
?>
```

## INSERT QUERY

```
<?php
```

```
$sql =
```

```
"INSERT INTO student
```

```
(name,course)
```

```
VALUES
```

```
('Alisha','MCA');"
```

```
mysqli_query($conn,$sql);
```

```
?>
```

## SELECT QUERY

```
<?php
```

```
$sql =
```

```
"SELECT * FROM student";
```

```
$result =
```

```
mysqli_query($conn,$sql);
```

```
?>
```

## **CRUD OPERATIONS**

CRUD stands for:

### **Create**

Insert data.

### **Read**

Retrieve data.

### **Update**

Modify data.

### **Delete**

Remove data.

## **MODERN WEB APPLICATIONS**

Modern web applications combine:

- HTML
- CSS
- JavaScript
- AJAX
- PHP
- Databases
- APIs

Examples:

- E-Commerce Websites
- Banking Systems

- Social Media Platforms
- Learning Management Systems

## **APPLICATIONS OF XML, AJAX AND PHP**

### **XML**

- Data Exchange
- Configuration Files

### **AJAX**

- Search Suggestions
- Live Updates

### **PHP**

- Dynamic Websites
- E-Commerce Portals

### **Web Services**

- Mobile Applications
- Cloud Services

PROF PRITAM NANDA

## **UNIT – V**

### **E-COMMERCE, WEB SECURITY, DIGITAL SIGNATURE, SSL, CYBER SECURITY, SEO, CLOUD COMPUTING AND EMERGING WEB TECHNOLOGIES:**

#### **INTRODUCTION TO E-COMMERCE**

E-Commerce (Electronic Commerce) refers to the buying and selling of goods and services through electronic networks, mainly the Internet.

It enables businesses and customers to conduct transactions online without physical interaction.

Today, E-Commerce has become an essential part of modern business due to its convenience, speed, and global reach.

#### **DEFINITION OF E-COMMERCE**

E-Commerce is the process of conducting commercial transactions electronically over the Internet.

#### **FEATURES OF E-COMMERCE**

1. Global Reach
2. 24x7 Availability
3. Fast Transactions
4. Reduced Operational Cost
5. Online Payment Facility
6. Better Customer Service
7. Easy Product Comparison

#### **TYPES OF E-COMMERCE**

##### **1. Business to Business (B2B)**

Transactions occur between businesses.

**Example**

A manufacturer sells products to a wholesaler.

**2. Business to Consumer (B2C)**

Business directly sells products to customers.

**Example**

Amazon

**3. Consumer to Consumer (C2C)**

Consumers sell products to other consumers.

**Example**

Online marketplace platforms.

**4. Consumer to Business (C2B)**

Consumers provide services to businesses.

**Example**

Freelancing websites.

**ADVANTAGES OF E-COMMERCE**

1. Global market access.
2. Reduced business cost.
3. Faster transactions.
4. Better customer satisfaction.
5. Easy inventory management.

**DISADVANTAGES OF E-COMMERCE**

1. Security risks.
2. Internet dependency.
3. Lack of physical inspection.
4. Delivery delays.

**E-COMMERCE ARCHITECTURE**

Customer



Web Browser



Web Server



Application Server



Database Server

## **ELECTRONIC PAYMENT SYSTEM**

An Electronic Payment System allows online transfer of money between customers and merchants.

### **TYPES OF ELECTRONIC PAYMENT**

#### **Credit Card**

Allows payment using borrowed funds.

#### **Debit Card**

Payment directly from bank account.

#### **Internet Banking**

Online banking services.

#### **Mobile Wallets**

Examples:

- Google Pay
- PhonePe

#### **UPI (Unified Payments Interface)**

Instant fund transfer system.

## **WEB SECURITY**

Web Security refers to protecting websites, web applications, and data from unauthorized access, attacks, and damage.

### **NEED FOR WEB SECURITY**

1. Protect user information.
2. Prevent cyber attacks.
3. Maintain confidentiality.
4. Ensure integrity of data.

5. Protect organizational assets.

## **SECURITY GOALS**

### **Confidentiality**

Information should be accessible only to authorized users.

#### **Example**

Bank account passwords.

### **Integrity**

Data should not be modified without permission.

#### **Example**

Exam result records.

### **Availability**

Resources should be available whenever required.

#### **Example**

Online banking systems.

## **CYBER SECURITY**

Cyber Security is the practice of protecting computers, networks, applications, and data from cyber threats.

## **TYPES OF CYBER THREATS**

### **Virus**

A malicious program that infects files and programs.

#### **Characteristics**

- Replicates itself.
- Damages system files.
- Slows system performance.

### **Worm**

A self-replicating malware that spreads automatically.

#### **Characteristics**

- No user interaction required.
- Consumes network resources.

### **Trojan Horse**

Malicious software disguised as legitimate software.

#### **Example**

Fake software downloads.

### **Phishing**

Fraudulent attempts to steal sensitive information.

#### **Example**

Fake banking emails requesting passwords.

### **Ransomware**

Malware that encrypts files and demands payment.

#### **Example**

Locking user data until ransom is paid.

### **FIREWALL**

A Firewall is a network security device that monitors and controls incoming and outgoing traffic.

#### **FUNCTIONS OF FIREWALL**

1. Blocks unauthorized access.
2. Filters malicious traffic.
3. Protects internal networks.
4. Prevents cyber attacks.

### **DIGITAL SIGNATURE**

A Digital Signature is an electronic technique used to verify the authenticity and integrity of digital documents.

It acts like a handwritten signature in electronic form.

#### **OBJECTIVES OF DIGITAL SIGNATURE**

1. Authentication
2. Integrity
3. Non-Repudiation

## **WORKING OF DIGITAL SIGNATURE**

Step 1:

Create document.

Step 2:

Generate hash value.

Step 3:

Encrypt hash using sender's private key.

Step 4:

Send document and signature.

Step 5:

Receiver verifies signature using public key.

## **DIGITAL SIGNATURE PROCESS**

Document



Hash Function



Hash Value



Private Key Encryption



Digital Signature

## **ADVANTAGES OF DIGITAL SIGNATURE**

1. Secure authentication.
2. Data integrity.
3. Legal validity.
4. Prevents forgery.

## **SSL (SECURE SOCKETS LAYER)**

SSL is a security protocol used to establish encrypted communication between a web browser and a web server.

Modern systems use TLS (Transport Layer Security), which evolved from SSL.

## **PURPOSE OF SSL**

1. Data Encryption
  2. Authentication
  3. Secure Communication
- 

## **SSL WORKING**

Browser



SSL Request



Server Certificate



Encrypted Session



Secure Communication

## **SSL CERTIFICATE**

An SSL Certificate contains:

1. Website Identity
2. Public Key
3. Certificate Authority Information

## **BENEFITS OF SSL**

1. Secure transactions.
2. User trust.
3. Data protection.
4. HTTPS support.

## **HTTPS**

HTTPS stands for:

**HyperText Transfer Protocol Secure**

It combines:

HTTP + SSL/TLS

## **DIFFERENCE BETWEEN HTTP AND HTTPS**

<b>HTTP</b>	<b>HTTPS</b>
Not secure	Secure
No encryption	Encrypted
Port 80	Port 443
Less reliable	Highly secure

## **SEARCH ENGINE OPTIMIZATION (SEO)**

SEO is the process of improving a website's visibility in search engine results.

It helps websites rank higher and attract more visitors.

### **OBJECTIVES OF SEO**

1. Increase website traffic.
2. Improve search ranking.
3. Enhance user experience.
4. Increase business opportunities.

### **TYPES OF SEO**

#### **On-Page SEO**

Optimization performed within the website.

Examples:

- Keywords
- Meta Tags
- Headings
- Content Optimization

#### **Off-Page SEO**

Activities performed outside the website.

Examples:

- Backlinks
- Social Media Promotion
- Guest Blogging

## **SEO TECHNIQUES**

### **Keyword Optimization**

Using relevant keywords.

### **Quality Content**

Creating valuable content.

### **Mobile Friendly Design**

Responsive websites.

### **Fast Loading Pages**

Improved performance.

## **BENEFITS OF SEO**

1. Higher search rankings.
2. Increased traffic.
3. Better brand visibility.
4. Long-term growth.

## **CLOUD COMPUTING**

Cloud Computing is the delivery of computing services such as storage, servers, databases, networking, and software over the Internet.

Instead of owning physical infrastructure, users access resources on demand.

## **CHARACTERISTICS OF CLOUD COMPUTING**

1. On-Demand Service
2. Broad Network Access
3. Resource Pooling
4. Rapid Elasticity
5. Measured Service

## **SERVICE MODELS OF CLOUD COMPUTING**

### **Infrastructure as a Service (IaaS)**

Provides virtualized hardware resources.

Example:

Amazon Web Services

### **Platform as a Service (PaaS)**

Provides development platforms.

Example:

Google Cloud

### **Software as a Service (SaaS)**

Provides software through Internet.

Examples:

- Google Docs
- Microsoft 365

### **ADVANTAGES OF CLOUD COMPUTING**

1. Cost reduction.
2. Scalability.
3. Flexibility.
4. Data backup.
5. High availability.

### **DISADVANTAGES OF CLOUD COMPUTING**

1. Security concerns.
2. Internet dependency.
3. Limited control.
4. Downtime risks.

### **EMERGING WEB TECHNOLOGIES**

Emerging Web Technologies are modern technologies shaping the future of web development.

### **INTERNET OF THINGS (IoT)**

IoT refers to interconnected smart devices communicating through the Internet.

Examples:

- Smart Home Systems
- Smart Watches
- Smart Sensors

## **ARTIFICIAL INTELLIGENCE (AI)**

AI enables machines to perform tasks requiring human intelligence.

Applications:

- Chatbots
- Virtual Assistants
- Recommendation Systems

## **BIG DATA**

Big Data refers to extremely large datasets requiring advanced processing.

Applications:

- Business Analytics
- Healthcare
- Banking

## **BLOCKCHAIN TECHNOLOGY**

Blockchain is a distributed digital ledger technology.

Features:

1. Transparency
2. Security
3. Decentralization

Applications:

- Digital Transactions
- Supply Chain Management

## **WEB 3.0**

Web 3.0 represents the next generation of the web focusing on:

1. Decentralization
2. Artificial Intelligence
3. Semantic Data
4. Enhanced User Control

## **APPLICATIONS OF INTERNET AND WEB TECHNOLOGY**

1. Online Banking
2. E-Commerce

3. Social Networking
4. Digital Marketing
5. Cloud Services
6. E-Learning
7. Telemedicine
8. Smart Cities

PROF PRITAM NANDA