



Web Technologies

SEC103

Bachelor of Computer Applications

I Year II Semester

G Prembihari Saran



Berhampur University

Unit-1

Introduction to www, Protocols and Programs, Applications and development tools, Web Browsers, DNS, Web Hosting Provider, Setting up Windows/Linux/Unix web servers, Webhosting in cloud, Types of Web hosting.

Introduction to HTML, history of HTML, Objective, basic Structures of HTML, Header Tags, body tags, Paragraph Tags.

Tags for FORM Creation, TABLE, FORM, TEXTAREA, SELECT, IMG, IFRAME FIELDSET, ANCHOR. Lists in HTML, Introduction to DIV tag, NAVBAR Design.

Introduction to CSS, types, Selectors, and Responsiveness of a web page.

Introduction to Bootstrap, downloads/linking, using classes of Bootstrap, understanding the Grid System in Bootstrap

Introduction to www

Introduction to the World Wide Web (WWW)

The World Wide Web (WWW or simply the Web) is a vast information space where documents and other digital resources are:

- Identified by Uniform Resource Locators (URLs)
- Connected through hypertext links
- Accessed over the Internet using web browsers

Origin and Development

The Web was invented by Sir Tim Berners-Lee, a British scientist, in 1989 while working at CERN (European Organization for Nuclear Research) in Switzerland.

In 1990, he developed the first web browser and editor, named World Wide Web.

- The browser was made available:
- To research institutions in January 1991
- To the general public in August 1991
- This innovation laid the foundation for the modern Internet experience.

Significance in the Information Age

The World Wide Web has played a pivotal role in shaping the Information Age, becoming the primary tool for billions of people to access, share, and interact with content globally.

Structure and Components

Web pages are primarily text-based documents formatted using HTML (Hypertext Markup Language).

In addition to text, web pages may incorporate:

- Images
- Audio
- Video
- Interactive software components

These elements are rendered by web browsers into unified, interactive multimedia experiences.

Hyperlinks and Navigation

Embedded hyperlinks within web pages allow seamless navigation across the web.

A collection of interconnected web pages under a single domain constitutes a website.

Website Classifications

Websites may be:

- Informational – e.g., news sites, blogs
- Interactive – e.g., social networks, forums
- Transactional – e.g., e-commerce platforms
- Organizational – used by governments, NGOs, or corporations

Content can be static (publisher-controlled) or dynamic, influenced by user interaction or generated in real-time.

Client-Server Architecture

The Web operates on a client/server model:

- The client (web browser) sends a request for a document or resource
- The server (web server) processes the request and sends back the content
- This communication uses the HTTP (Hypertext Transfer Protocol)

Web documents are created using HTML, which structures and formats content for browser display.



The functioning of the World Wide Web relies heavily on specific protocols and software programs that facilitate communication, content creation, and content access over the Internet.

1. Protocols Used in the Web

Protocols are standardized rules that allow computers to communicate over networks. Key protocols that power the Web include:

a. HTTP (Hypertext Transfer Protocol)

- The **foundation of data communication** on the Web.
- It defines how messages are formatted and transmitted between **web browsers (clients)** and **web servers**.
- The **secure version, HTTPS (HTTP Secure)**, uses encryption (SSL/TLS) to protect data.

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

- The **basic communication language** of the Internet.
- **TCP** ensures data is transmitted reliably.
- **IP** handles addressing and routing between devices on the network.

c. DNS (Domain Name System)

- Translates **human-readable domain names** (like www.example.com) into **IP addresses** (like 192.0.2.1).
- Acts like a phonebook of the Internet.

d. FTP (File Transfer Protocol)

- Used to **upload or download files** between a client and server.
- Often used for **website management and deployment**.

e. SSL/TLS (Secure Sockets Layer / Transport Layer Security)

- Protocols that provide **data encryption and secure communication** over the Web.
- Widely used with HTTPS to protect sensitive data like passwords and payment information.

2. Programs Used in the Web

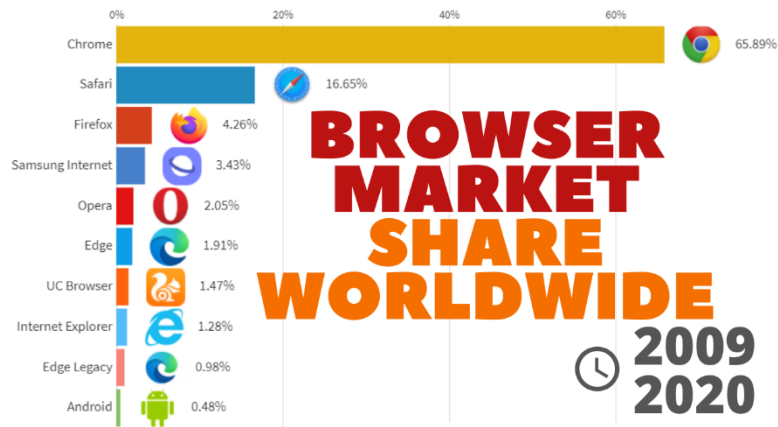
Various software programs make the Web accessible, usable, and functional:

a. Web Browsers (Clients)

These programs allow users to view and interact with web content.

Examples:

- **Google Chrome**
- **Mozilla Firefox**
- **Microsoft Edge**
- **Safari**
- **Opera**



Browsers interpret HTML, CSS, JavaScript, and render websites into interactive pages.

b. Web Servers

Servers **store, process, and deliver web content** to clients upon request.

Popular web servers:

- **Apache HTTP Server**
- **Nginx**
- **Microsoft IIS (Internet Information Services)**
- **LiteSpeed**

c. Web Development Tools

Used to create and maintain websites.

Examples:

- **HTML/CSS/JavaScript editors** (e.g., VS Code, Sublime Text)
- **Frameworks** (e.g., React, Angular, Django)
- **Content Management Systems (CMS)** like **WordPress, Drupal, Joomla**

d. Protocol Handlers & Plugins

- Programs or browser extensions that handle specific types of content or network protocols (e.g., PDF viewers, media players, Flash, Java applets—although many are now deprecated).

Applications and development tools

I. Applications of the World Wide Web

The World Wide Web serves as a platform for a wide range of applications that impact nearly every aspect of modern life.

1. Education

- **E-learning platforms** (e.g., Coursera, Khan Academy, NPTEL)
- Online libraries, tutorials, and MOOCs
- Virtual classrooms and webinars

2. E-commerce

- Online shopping portals (e.g., Amazon, Flipkart)
- Digital payments, product reviews, and customer support
- B2B, B2C, and C2C commerce platforms

3. Communication

- Web-based **email** services (e.g., Gmail, Outlook)
- **Instant messaging**, video calls (e.g., WhatsApp Web, Zoom, Google Meet)
- Social networking sites (e.g., Facebook, Instagram, LinkedIn)

4. Information Sharing

- News websites, blogs, wikis (e.g., Wikipedia)
- Online journals, research papers
- Search engines like Google, Bing

5. Entertainment

- **Streaming services** (e.g., YouTube, Netflix, Spotify)
- Online gaming platforms
- Digital art, memes, and fan fiction

6. Government and Public Services

- E-governance portals for citizen services
- Online tax filing, license renewal, and public records
- Digital ID verification and voting portals

7. Business and Professional Services

- **Remote work tools** (e.g., Google Workspace, Microsoft 365)
- Cloud computing services (e.g., AWS, Azure)

- Online job portals (e.g., Naukri, Indeed)

II. Web Development Tools

Web development tools help in designing, coding, testing, and deploying web applications efficiently.

1. Markup and Styling Tools

- **HTML** – Structures web content
- **CSS** – Styles and formats web pages
- **Sass/Less** – CSS preprocessors for modular styling

2. Programming Languages

- **JavaScript** – Client-side scripting for interactivity
- **PHP, Python, Ruby, Java** – Server-side scripting
- **TypeScript** – Superset of JavaScript with static typing

3. Frameworks and Libraries

- **Frontend:**
 - **React.js, Angular, Vue.js**
- **Backend:**
 - **Node.js, Django, Flask, Express.js**

4. Development Environments

- **Code Editors/IDEs:** VS Code, Sublime Text, Atom, IntelliJ IDEA
- **Version Control:** Git, GitHub, GitLab, Bitbucket

5. Testing and Debugging Tools

- **Browser DevTools** (built into Chrome, Firefox, etc.)
- **Jest, Mocha, Selenium** – Testing frameworks
- **Postman** – API testing

6. Deployment and Hosting Platforms

- **Cloud Providers:** AWS, Azure, Google Cloud
- **Web Hosts:** Netlify, Vercel, Heroku, GitHub Pages
- **CI/CD Tools:** Jenkins, Travis CI, GitHub Actions

7. Content Management Systems (CMS)

- WordPress, Drupal, Joomla
- Useful for creating websites without writing much code

8. Security Tools

- SSL/TLS certificates
- OWASP ZAP, Burp Suite – for vulnerability testing

Web Browsers

What is a Web Browser?

A **web browser** is a **software application** that enables users to access, retrieve, and view content on the **World Wide Web**. It acts as a **client** in the client-server model, sending requests to web servers and displaying the returned documents (typically written in HTML, CSS, and JavaScript).

Key Functions of a Web Browser

- Interprets and renders **HTML documents** into visual and interactive web pages
- Sends **HTTP/HTTPS requests** to web servers
- Displays **multimedia content** (text, images, video, audio)
- Stores **data temporarily** via caching, cookies, and local storage
- Supports **extensions and plugins** for additional functionality

Popular Web Browsers

Browser	Developer	Key Features
Google Chrome	Google	Fast, secure, supports extensions, widely used
Mozilla Firefox	Mozilla Foundation	Open-source, privacy-focused, flexible extensions
Safari	Apple	Optimized for macOS and iOS, energy-efficient
Microsoft Edge	Microsoft	Built on Chromium, integrates with Windows services
Opera	Opera Software	Built-in VPN, ad blocker, sidebar messengers
Brave	Brave Software	Blocks ads and trackers by default, privacy-first
Tor Browser	Tor Project	Routes traffic through Tor network for anonymity

DNS

DNS (Domain Name System) is a fundamental component of the internet that acts like a phonebook for the web.

◆ What is DNS?

DNS translates human-readable domain names (like www.google.com) into IP addresses (like 142.250.195.4) that computers use to identify each other on the network.

◆ Common DNS Records

Record Type	Description	Example
A	Maps a domain to IPv4 address	example.com → 192.0.2.1
AAAA	Maps a domain to IPv6 address	example.com → 2001:db8::1
CNAME	Alias of one name to another	www.example.com → example.com
MX	Mail exchange – directs emails	Mail server for example.com
NS	Name servers for a domain	Tells who is authoritative
TXT	Text record (e.g., SPF for email auth)	Security, verification, etc.

Web Hosting Provider

🌐 What is a Web Hosting Provider?

A **Web Hosting Provider** is a **service that stores your website's files** (HTML, CSS, images, databases, etc.) on a **server** and makes them **accessible on the internet**.

When someone types your domain (e.g., www.example.com), the hosting provider delivers your website to their browser.

◆ Key Roles of a Web Hosting Provider

Function Description

Store website files Keeps all your website's data on secure servers.

Ensure uptime Keeps your website live and accessible 24/7.

Domain support Often includes domain registration and DNS management.

Function	Description
Email services	Lets you create custom emails like you@example.com.
Security	Offers SSL certificates, DDoS protection, firewalls.
Support	Provides technical help (chat, ticket, or phone).

◆ Types of Web Hosting

Hosting Type	Description	Suitable For
Shared Hosting	Multiple sites share one server. Cheapest.	Beginners, small websites
VPS Hosting	Virtual private server with more control.	Growing websites
Dedicated Hosting	One server = one client. Full control.	Large businesses
Cloud Hosting	Uses multiple servers. Scalable.	High-traffic websites
Managed Hosting	Hosting provider handles maintenance.	WordPress users, non-tech users

◆ Popular Web Hosting Providers (2025)

Provider	Known For
Bluehost	Beginner-friendly, WordPress recommended
HostGator	Affordable and reliable
SiteGround	Fast support, security-focused
A2 Hosting	Speed and developer tools
GoDaddy	Domain + hosting bundles
AWS (Amazon)	Enterprise-grade cloud hosting
DigitalOcean	Developer-focused, cloud VMs
Namecheap	Affordable domains + hosting

Setting up Windows/Linux/Unix web

⚙️ Step-by-Step Guide: Setting Up Web Servers

● 1. On Windows (Using IIS or WAMP/XAMPP)

Option A: Using IIS (Internet Information Services)

✓ **Best for:** Windows Server users, enterprise use

Steps:

1. Enable IIS:

- Go to *Control Panel > Programs > Turn Windows features on/off*
- Check **Internet Information Services (IIS)** and click OK

2. Access IIS Manager:

- Search for **IIS Manager** in the Start Menu

3. Place Your Website Files:

- Copy your files to `C:\inetpub\wwwroot`

4. Test:

- Open browser and go to `http://localhost`

Option B: Using WAMP/XAMPP

✓ **Best for:** PHP + MySQL development

Steps (XAMPP example):

1. Download and install XAMPP.
2. Launch **XAMPP Control Panel** → Start **Apache** and **MySQL**.
3. Place your website files in `C:\xampp\htdocs`.
4. Visit `http://localhost/yourfoldername` in your browser.

● 2. On Linux (Ubuntu/Debian)

✓ **Best for:** Developers, scalable production deployment

Option A: Using Apache (LAMP Stack)

Install Apache:

bash

CopyEdit

sudo apt update

```
sudo apt install apache2
```

Start the server:

```
bash
```

```
CopyEdit
```

```
sudo systemctl start apache2
```

Check status:

```
bash
```

```
CopyEdit
```

```
sudo systemctl status apache2
```

Web root directory:

```
bash
```

```
CopyEdit
```

```
/var/www/html
```

Test:

```
Open browser → http://localhost
```

Option B: Using Nginx (LEMP Stack)

Install Nginx:

```
bash
```

```
CopyEdit
```

```
sudo apt install nginx
```

Start and enable:

```
bash
```

```
CopyEdit
```

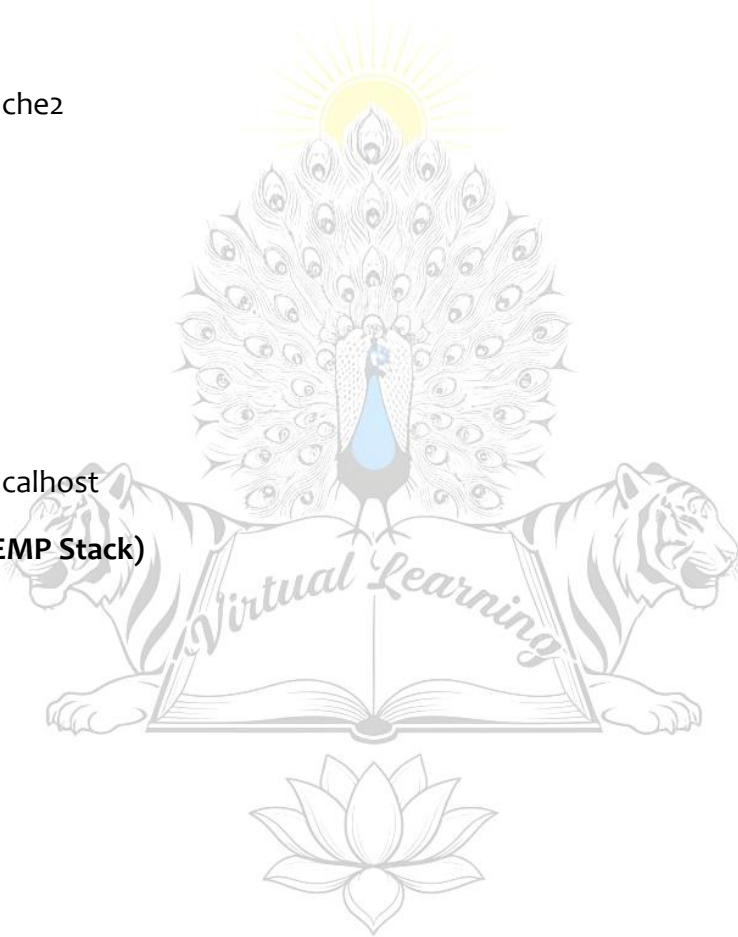
```
sudo systemctl start nginx
```

```
sudo systemctl enable nginx
```

Place files in:

```
bash
```

```
CopyEdit
```



/var/www/html

Test:

Visit <http://localhost>

3. On Unix (General Instructions)

Unix systems work similarly to Linux. Use either **Apache** or **Nginx**:

Install Apache:

bash

CopyEdit

```
sudo pkg install apache24 # On FreeBSD, for example
```

```
sudo service apache24 start
```

Configure:

- Website files: /usr/local/www/apache24/data
- Config file: /usr/local/etc/apache24/httpd.conf

Start Server:

bash

CopyEdit

```
sudo service apache24 start
```

Hosting Static and Dynamic Sites

Content Type Tools Required

Static HTML Just Apache/Nginx/IIS

PHP/MySQL LAMP/WAMP stack (PHP + DB)

Python/Flask Install Flask + WSGI + Nginx

Node.js App Install Node.js, use PM2 + Nginx



Webhosting in cloud servers

Web Hosting in the Cloud – Explained

Cloud web hosting is a type of web hosting that uses a **network of virtual servers (cloud)** to host websites or applications, rather than a single physical server.

It's scalable, flexible, and reliable — making it ideal for modern websites, apps, and services.

◆ How Cloud Hosting Works

- Your website's files are stored **across multiple servers**.
- If one server goes down, another instantly takes over — **high availability**.
- Resources (RAM, CPU, storage) are **dynamically allocated** as needed.
- You pay for **what you use** — like electricity or water.

◆ Cloud Hosting vs Traditional Hosting

Feature	Traditional Hosting	Cloud Hosting
Server Location	One physical server	Multiple virtual servers
Scalability	Limited	High (on-demand scaling)
Reliability	Single point of failure	Redundant (self-healing infrastructure)
Pricing	Fixed monthly/yearly plans	Pay-as-you-go (usage-based)
Performance	May vary	High and consistent

◆ Popular Cloud Web Hosting Providers

Provider	Key Highlights
Amazon Web Services (AWS)	Widely used, scalable, robust. Ideal for enterprise-grade apps.
Google Cloud Platform (GCP)	Fast, secure, integrated with Google's AI & analytics tools.
Microsoft Azure	Ideal for enterprise and hybrid cloud deployments.
DigitalOcean	Simple, developer-friendly. Great for startups and small teams.
Linode	Affordable and powerful VPS hosting.
Vultr	Low-cost, SSD-based servers worldwide.
Cloudways	Managed hosting over AWS, GCP, DigitalOcean, etc.
Kamatera	Customizable servers with global datacenters.

Types of Web Hosting

Types of Web Hosting – Explained

Web hosting comes in different forms depending on the **size**, **traffic**, and **technical needs** of your website. Below are the most common types:

◆ 1. Shared Hosting

✔ **Best For:** Beginners, small websites, blogs

💰 **Cost:** Low

How it works:

Your website shares the same server resources (CPU, RAM, storage) with many other websites.

Pros:

- Cheapest option
- Easy to set up
- Beginner-friendly control panel (like cPanel)

Cons:

- Slower performance during high traffic
- Limited control and customization

◆ 2. VPS Hosting (Virtual Private Server)

✔ **Best For:** Growing websites, moderate traffic

💰 **Cost:** Moderate

How it works:

You still share a physical server, but each user gets **isolated resources** (virtualized).

Pros:

- Better performance than shared hosting
- More control and customization
- Scalable

Cons:

- Requires some technical knowledge
- More expensive than shared hosting

◆ 3. Dedicated Hosting

✔ **Best For:** Large businesses, high-traffic sites

💎 **Cost:** High

How it works:

You rent an **entire server** dedicated solely to your website.

Pros:

- Full control over the server
- High performance and reliability
- Custom security configurations

Cons:

- Very expensive
- Needs server management skills or a system admin

◆ 4. Cloud Hosting

✔ **Best For:** Scalable applications, startups, enterprises

💎 **Cost:** Pay-as-you-go

How it works:

Your site is hosted on a **cluster of servers**. If one server fails, others take over.

Pros:

- Extremely scalable and reliable
- Good for unpredictable traffic spikes
- Pay only for what you use

Cons:

- Can be more complex
- Pricing may be harder to predict

◆ 5. Managed Hosting

✔ **Best For:** WordPress users, non-tech users

💎 **Cost:** Varies (moderate to high)

How it works:

The hosting provider **handles all technical tasks:** updates, backups, security, performance tuning.

Pros:

- Hassle-free
- Secure and optimized
- 24/7 expert support

Cons:

- Less control
- More expensive than self-managed options

◆ **6. Reseller Hosting**

✔ **Best For:** Web agencies, freelancers

◆ **Cost:** Varies

How it works:

You buy hosting space in bulk and **resell it to your clients** under your brand.

Pros:

- Create your own hosting business
- White-label branding
- Centralized client management

Cons:

- Requires support and business skills

◆ **7. Colocation Hosting**

✔ **Best For:** Enterprises with own servers

◆ **Cost:** High (initial + maintenance)

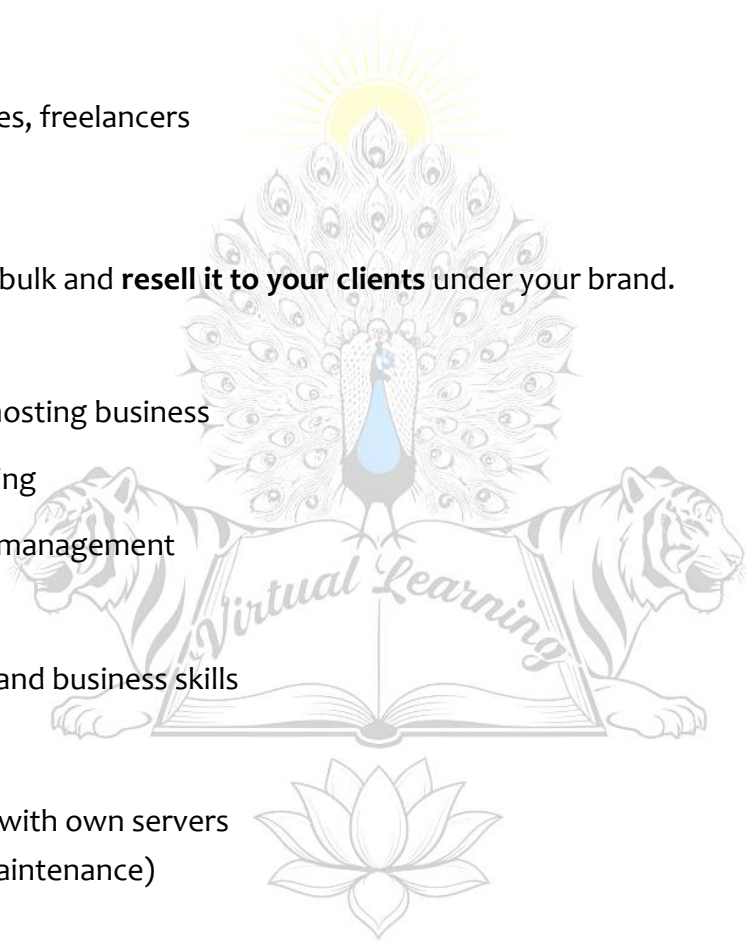
How it works:

You **own the hardware**, but rent space in a **data center** to house it.

Pros:

- Full control over your server
- High security and redundancy

Cons:



- High cost
- You manage everything (hardware/software)

Introduction to HTML

🌐 Introduction to HTML (Hyper Text Markup Language)

◆ What is HTML?

HTML stands for **Hyper Text Markup Language**.

It is the **standard language** used to **create and structure webpages** on the internet.

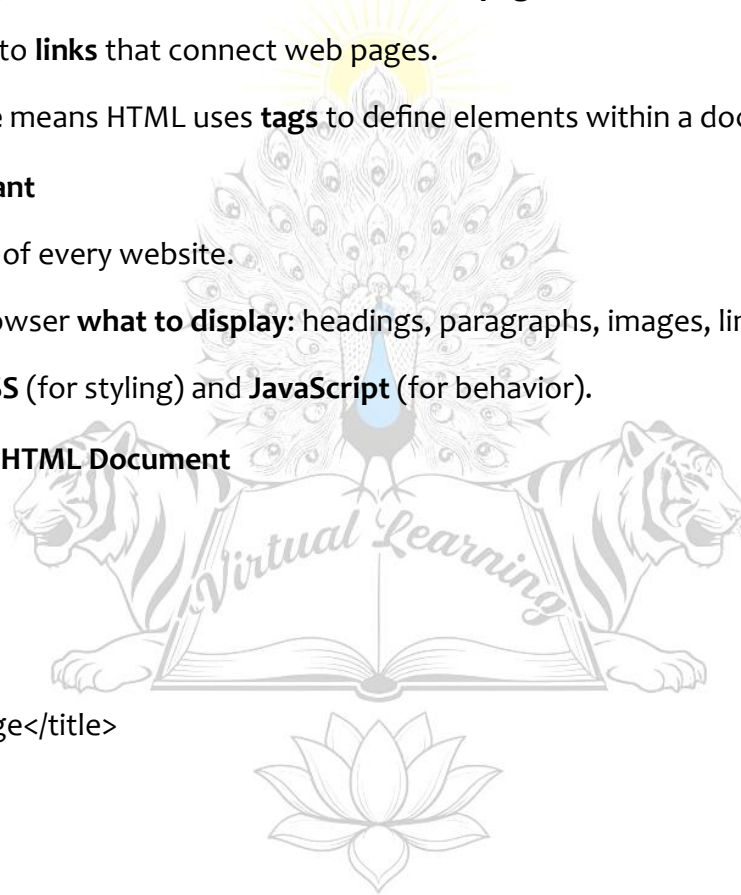
- **Hyper Text** refers to **links** that connect web pages.
- **Markup Language** means HTML uses **tags** to define elements within a document.

◆ Why HTML is Important

- It is the **backbone** of every website.
- HTML tells the browser **what to display**: headings, paragraphs, images, links, etc.
- Used alongside **CSS** (for styling) and **JavaScript** (for behavior).

◆ Basic Structure of an HTML Document

```
<!DOCTYPE html>
<html>
<head>
  <title>My First Web Page</title>
</head>
<body>
  <h1>Welcome to HTML</h1>
  <p>This is a paragraph.</p>
</body>
</html>
```



History of HTML

History of HTML (Hyper Text Markup Language)

HTML has evolved significantly since its invention, becoming the **foundation of the modern web**. Here's a concise timeline and overview of its development:

Year	Version	Key Milestones
1991	HTML 1.0	First release by Tim Berners-Lee
1995	HTML 2.0	Standardized early HTML
1997	HTML 3.2	Tables, scripting support
1999	HTML 4.01	More structure, styling
2000	XHTML 1.0	Stricter XML-based HTML
2014	HTML5	Rich media, modern web applications
Now	Living Std.	HTML continuously evolves via WHATWG

Objective

Objective of HTML

The **primary objective of HTML (Hyper Text Markup Language)** is to **structure content on the web** and enable browsers to display **text, images, multimedia, and links** in a readable and organized format.

◆ Main Objectives of HTML

1. Structure Web Content

- Define headings, paragraphs, lists, tables, forms, etc.
- Provide semantic meaning using tags like `<article>`, `<section>`, `<nav>`, etc.

2. Create Links (Hypertext)

- Use `<a>` tags to connect web pages and documents.
- Enable navigation through websites via internal and external links.

3. Embed Multimedia Elements

- Display images (``), videos (`<video>`), and audio (`<audio>`) in web pages.

4. Form User Interfaces

- Build forms using `<form>`, `<input>`, `<select>`, `<textarea>`, etc.
- Enable user interaction and data submission.

5. Support Styling and Scripting

- Work alongside **CSS** for design and layout.
- Integrate **JavaScript** for interactivity and dynamic behavior.

6. Ensure Accessibility & SEO

- Use semantic tags to improve **accessibility for screen readers** and **search engine indexing**.

7. Standardize Web Content

- Provide a consistent way to create and display websites across different browsers and platforms.

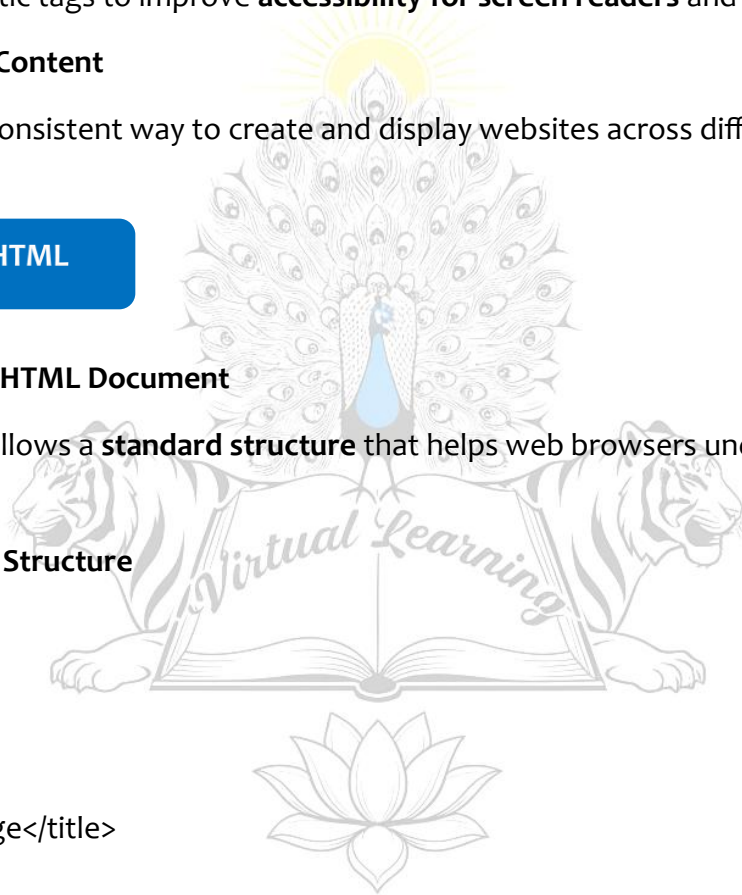
Basic Structures of HTML

Basic Structure of an HTML Document

Every HTML document follows a **standard structure** that helps web browsers understand and render the page correctly.

◆ Standard HTML Page Structure

```
<!DOCTYPE html>
<html>
<head>
  <title>My First Web Page</title>
</head>
<body>
  <h1>Welcome to HTML</h1>
  <p>This is a paragraph.</p>
</body>
</html>
```



◆ Explanation of Each Part

Element	Description
---------	-------------

<!DOCTYPE html>	Declares the document type as HTML5 . Helps browser render correctly.
-----------------	--

<html>	Root element that wraps the entire HTML document .
--------	---

<head>	Contains meta-information : title, styles, scripts, etc.
--------	---

<title>	Sets the title of the page , displayed in the browser tab.
---------	---

<body>	Contains the visible content of the web page (text, images, links, etc.)
--------	---

<h1>, <p>	Elements inside <body> that display content on the screen.
-----------	---

◆ Additional Optional Elements

You can enhance the structure with other optional elements:





- <meta charset="UTF-8"> – Defines character encoding
- <link rel="stylesheet"> – Link to CSS file
- <script src="app.js"> – Link to JavaScript file
- <footer>, <nav>, <section>, <article> – Semantic structure



Header Tags

📌 HTML Header Tags

HTML provides **six levels of header tags** to define headings and subheadings on a web page. These are essential for organizing content hierarchically and improving readability and accessibility.

◆ List of Header Tags

Tag	Description	Example Output
<h1>	Main heading (most important)	 Large and bold
<h2>	Subheading under <h1>	 Slightly smaller
<h3>	Subheading under <h2>	 Smaller
<h4>	Subheading under <h3>	 Even smaller

<code><h5></code>	Subheading under <code><h4></code>	 Smaller still
<code><h6></code>	Least important heading	 Smallest size

◆ Example Usage

`<h1>`This is a Main Heading`</h1>`

`<h2>`This is a Subheading`</h2>`

`<h3>`This is a Section Heading`</h3>`

`<h4>`This is a Subsection Heading`</h4>`

`<h5>`This is a Smaller Subsection`</h5>`

`<h6>`This is the Smallest Heading`</h6>`

Body Tags

HTML `<body>` Tag – Explained

The `<body>` tag in HTML represents the **main content** of an HTML document that is **visible to users in the browser**. Everything you see on a webpage—text, images, videos, links, forms—is placed inside the `<body>` tag.

◆ Syntax

`<body>`

`<!-- All visible content goes here -->`

`</body>`

◆ Location in HTML Structure

`<!DOCTYPE html>`

`<html>`

`<head>`

`<title>`Page Title`</title>`

`</head>`

`<body>`

`<h1>`Hello, World!`</h1>`

```
<p>This is a paragraph inside the body.</p>
</body>
</html>
```

- The <body> tag is placed **after <head>** and is enclosed within the <html> tag.

◆ **What You Can Include in <body>**

Element Type	Examples
Headings	<h1> to <h6>
Paragraphs	<p>
Links	Link
Images	
Lists	, ,
Tables	<table>, <tr>, <td>, <th>
Forms	<form>, <input>, <textarea>
Media	<audio>, <video>, <iframe>
Divisions & Layouts	<div>, , <section>, etc.

◆ **Attributes of <body> (HTML5+)**

While modern best practices recommend CSS for styling, some **deprecated attributes** used earlier include:

Attribute	Purpose	Status
bgcolor	Background color	Deprecated
text	Text color	Deprecated
link	Link color	Deprecated
onload	Executes JS when page loads	Still used
onunload	Executes JS when page unloads	Still used

Paragraph Tags

HTML <p> Tag – Paragraph Tag

The <p> tag in HTML defines a **paragraph** of text. It is a **block-level element**, meaning it starts on a new line and adds space above and below the content automatically.

◆ Syntax

```
<p>This is a paragraph of text.</p>
```

◆ Characteristics of <p> Tag

Feature	Description
Block-level Element	Takes up the full width and starts on a new line.
Automatic Spacing	Adds space before and after the paragraph.
Can Contain	Inline elements like , <i>, <a>, etc.
Nesting	Cannot contain block-level elements inside.

Form

HTML Form Tags – Elements for Form Creation

HTML provides a set of specialized **form tags** to collect user input such as text, selections, checkboxes, radio buttons, and submit actions. These are widely used in login pages, registrations, contact forms, etc.

◆ Basic Form Structure

```
<form action="/submit" method="post">
```

```
<!-- Form elements go here -->
```

```
</form>
```

Attribute Description

action URL where form data will be sent

method HTTP method: get or post

◆ Common Form Tags & Their Uses

Tag	Purpose	Example Code
<code><input></code>	Generic input field (text, email, etc.)	<code><input type="text" name="username"></code>
<code><textarea></code>	Multi-line text input	<code><textarea name="message"></textarea></code>
<code><select></code>	Dropdown selection	<code><select><option>Option</option></select></code>
<code><option></code>	Options inside <code><select></code>	<code><option value="1">One</option></code>
<code><button></code>	Clickable button (submit/reset/custom)	<code><button type="submit">Send</button></code>
<code><label></code>	Label for form controls	<code><label for="email">Email</label></code>
<code><fieldset></code>	Group related fields together	<code><fieldset>...</fieldset></code>
<code><legend></code>	Caption for <code><fieldset></code>	<code><legend>User Info</legend></code>
<code><input type="radio"></code>	Radio button for single-choice	<code><input type="radio" name="gender"></code>
<code><input type="checkbox"></code>	Checkbox for multiple-choice	<code><input type="checkbox" name="agree"></code>
<code><input type="submit"></code>	Submit button	<code><input type="submit" value="Submit"></code>

◆ Example Form

```

<form action="/submit" method="post">
<label for="name">Name:</label>
<input type="text" id="name" name="name"><br><br>

<label for="email">Email:</label>
<input type="email" id="email" name="email"><br><br>

<label for="message">Message:</label><br>
<textarea id="message" name="message"></textarea><br><br>
<input type="submit" value="Send">
</form>

```

Table

HTML <table> Tag – Creating Tables in HTML

The <table> tag in HTML is used to **display data in a tabular (row-column) format**. Tables are ideal for organizing information like schedules, reports, pricing, and any structured data.

◆ Basic Table Structure

```
<table>
<tr>
  <th>Header 1</th>
  <th>Header 2</th>
</tr>
<tr>
  <td>Data 1</td>
  <td>Data 2</td>
</tr>
</table>
```

◆ Important Table Tags

Tag	Description
<table>	Defines the table
<tr>	Table row
<th>	Table header cell (bold, centered)
<td>	Table data cell
<caption>	Title of the table
<thead>	Groups the header content
<tbody>	Groups the body content
<tfoot>	Groups the footer content
colspan	Merges cells horizontally



rowspan	Merges cells vertically
----------------	-------------------------

◆ Example with Caption and Span

```
<table border="1">
  <caption>Student Grades</caption>
  <tr>
    <th>Student</th>
    <th colspan="2">Marks</th>
  </tr>
  <tr>
    <td>Alice</td>
    <td>Math: 90</td>
    <td>Science: 88</td>
  </tr>
  <tr>
    <td>Bob</td>
    <td colspan="2">Math & Science: 85</td>
  </tr>
</table>
```



Introduction to CSS

CSS (Cascading Style Sheets) controls the presentation and layout of HTML elements. It separates content from design, making websites more maintainable and flexible.

Types of CSS

1. **Inline CSS:** Applied directly to HTML elements

```
<p style="color: blue; font-size: 16px;">Styled text</p>
```

2. **Internal CSS:** Defined within the HTML document's head section

```
<head>
  <style>
    p { color: blue; font-size: 16px; }
  </style>
</head>
```

3. **External CSS:** Separate CSS file linked to HTML

```
<link rel="stylesheet" href="styles.css">
```

CSS Selectors

Basic Selectors

- **Element Selector:** Targets HTML elements

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

- **Class Selector:** Targets elements with specific class

```
.highlight { background-color: yellow; }
```

- **ID Selector:** Targets elements with specific ID

```
#header { font-size: 24px; }
```

- **Universal Selector:** Targets all elements

```
* { margin: 0; padding: 0; }
```

Attribute Selectors

```
input[type="text"] { border: 1px solid blue; }
[class^="nav"] { font-weight: bold; }
```

Responsiveness in Web Design

Responsive design ensures websites work across all devices and screen sizes using:

- **Media Queries:** Apply styles based on device characteristics

```
@media (max-width: 768px) {  
  body { font-size: 14px; }  
}
```

- **Flexible Grids:** Use percentages instead of fixed units

```
.container { width: 100%; max-width: 1200px; }  
.column { width: 33.33%; }
```

- **Fluid Images:** Scale images proportionally

```
img { max-width: 100%; height: auto; }
```

Introduction to Bootstrap

Bootstrap is a popular CSS framework that provides pre-built components and a responsive grid system for rapid web development.

Downloads and Linking

CDN Method (Recommended)

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"  
rel="stylesheet">  
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
```

Local Installation

Download Bootstrap files and link locally:

```
<link rel="stylesheet" href="css/bootstrap.min.css">  
<script src="js/bootstrap.bundle.min.js"></script>
```

Bootstrap Classes

Common Bootstrap utility classes:

- **Text:** `.text-center`, `.text-left`, `.text-primary`
- **Spacing:** `.m-3` (margin), `.p-4` (padding)
- **Display:** `.d-block`, `.d-none`, `.d-flex`
- **Colors:** `.bg-primary`, `.text-success`, `.btn-danger`

Bootstrap Grid System

Bootstrap uses a 12-column grid system that's fully responsive.

Basic Structure

```
<div class="container">
  <div class="row">
    <div class="col-md-4">Column 1</div>
    <div class="col-md-4">Column 2</div>
    <div class="col-md-4">Column 3</div>
  </div>
</div>
```

Grid Classes

- **xs:** Phones (< 768px)
- **sm:** Tablets (≥ 768px)
- **md:** Small laptops (≥ 992px)
- **lg:** Laptops & desktops (≥ 1200px)
- **xl:** Large desktops (≥ 1400px)



Grid Examples

```
<!-- Equal columns -->
<div class="row">
  <div class="col">Column</div>
  <div class="col">Column</div>
  <div class="col">Column</div>
</div>
```

```
<!-- Specific widths -->
<div class="row">
```

```
<div class="col-md-8">Main content</div>
<div class="col-md-4">Sidebar</div>
</div>

<!-- Responsive columns -->
<div class="row">
  <div class="col-12 col-md-6 col-lg-4">Responsive column</div>
</div>
```

World Wide Web (WWW)

Introduction to WWW

The **World Wide Web** is an information system that uses the internet to access interlinked hypertext documents. It was invented by Tim Berners-Lee in 1989 at CERN.

Protocols and Programs

HTTP/HTTPS Protocol

- **HTTP:** HyperText Transfer Protocol for transferring web pages
- **HTTPS:** Secure version using SSL/TLS encryption
- **Port 80:** Default for HTTP
- **Port 443:** Default for HTTPS

Other Web Protocols

- **FTP:** File Transfer Protocol for file uploads/downloads
- **TCP/IP:** Foundation protocol suite for internet communication
- **DNS:** Domain Name System for resolving domain names to IP addresses

Applications and Development Tools

Web Development Tools

- **Text Editors:** VS Code, Sublime Text, Atom
- **IDEs:** WebStorm, Eclipse, NetBeans

- **Frameworks:** React, Angular, Vue.js
- **Version Control:** Git, GitHub, GitLab
- **Build Tools:** Webpack, Gulp, Grunt

Web Browsers

Popular web browsers and their engines:

- **Chrome:** Blink engine (Google)
- **Firefox:** Gecko engine (Mozilla)
- **Safari:** WebKit engine (Apple)
- **Edge:** Blink engine (Microsoft)
- **Opera:** Blink engine

Domain Name System (DNS)

DNS translates human-readable domain names into IP addresses.

DNS Process

1. User enters domain name in browser
2. Browser checks local cache
3. DNS resolver queries root servers
4. Root server directs to TLD server
5. TLD server points to authoritative server
6. Authoritative server returns IP address
7. Browser connects to web server

DNS Record Types

- **A Record:** Points domain to IPv4 address
- **AAAA Record:** Points domain to IPv6 address
- **CNAME:** Creates alias for another domain
- **MX:** Specifies mail servers

- **NS:** Specifies name servers

Web Hosting Providers

Web hosting services provide server space and resources for websites.

Types of Web Hosting

1. **Shared Hosting:** Multiple websites share server resources
2. **VPS Hosting:** Virtual private server with dedicated resources
3. **Dedicated Hosting:** Entire server dedicated to one client
4. **Cloud Hosting:** Resources distributed across multiple servers
5. **Managed Hosting:** Provider handles server management

Popular Hosting Providers

- **Shared:** Bluehost, HostGator, GoDaddy
- **Cloud:** AWS, Azure, Google Cloud Platform
- **Managed:** WP Engine, Kinsta, Pantheon

Setting Up Web Servers

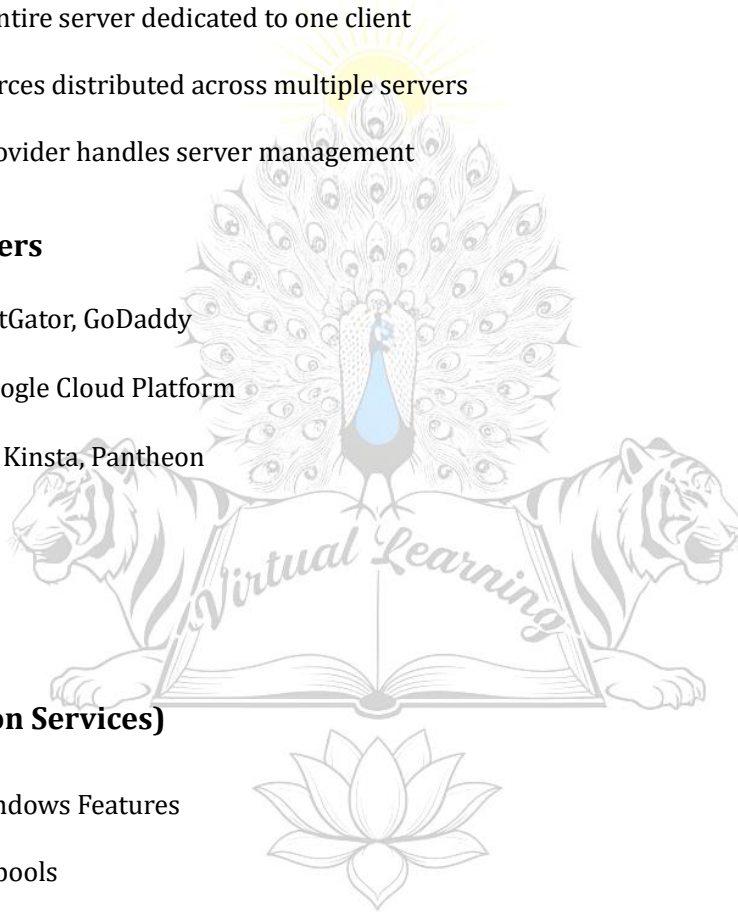
Windows Web Servers

IIS (Internet Information Services)

1. Enable IIS through Windows Features
2. Configure application pools
3. Create virtual directories
4. Set up SSL certificates
5. Configure authentication methods

Linux/Unix Web Servers

Apache HTTP Server



```
# Install Apache
sudo apt update
sudo apt install apache2

# Start Apache
sudo systemctl start apache2
sudo systemctl enable apache2

# Configure virtual hosts
sudo nano /etc/apache2/sites-available/example.com.conf
```

Nginx

```
# Install Nginx
sudo apt update
sudo apt install nginx

# Start Nginx
sudo systemctl start nginx
sudo systemctl enable nginx

# Configure server blocks
sudo nano /etc/nginx/sites-available/example.com
```

Web Hosting in Cloud

Cloud Hosting Benefits

- **Scalability:** Resources can be scaled up or down
- **Reliability:** Multiple server redundancy
- **Cost-effective:** Pay only for resources used
- **Global reach:** Data centers worldwide
- **Automatic backups:** Built-in backup solutions

Major Cloud Providers

Amazon Web Services (AWS)

- EC2: Virtual servers
- S3: Object storage
- CloudFront: Content delivery network
- Route 53: DNS service

Microsoft Azure

- Virtual Machines: Compute instances
- Blob Storage: Object storage
- CDN: Content delivery
- DNS Zone: Domain name services

Google Cloud Platform (GCP)

- Compute Engine: Virtual machines
- Cloud Storage: Object storage
- Cloud CDN: Content delivery
- Cloud DNS: Domain name system

Types of Web Hosting

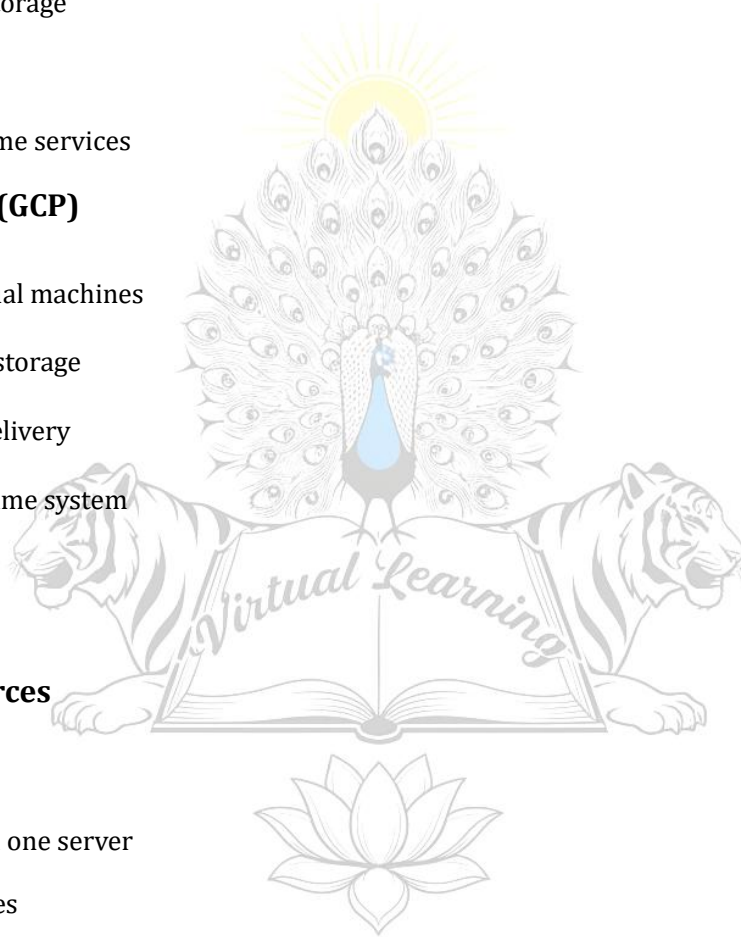
Classification by Resources

1. Shared Hosting

- Multiple sites on one server
- Limited resources
- Most affordable option
- Suitable for small websites

2. Virtual Private Server (VPS)

- Dedicated portion of server resources
- More control and customization
- Better performance than shared
- Moderate pricing



3. Dedicated Server

- Entire server for one client
- Full control and resources
- Highest performance
- Most expensive option

4. Cloud Hosting

- Resources from multiple servers
- High availability and scalability
- Pay-per-usage model
- Suitable for varying traffic

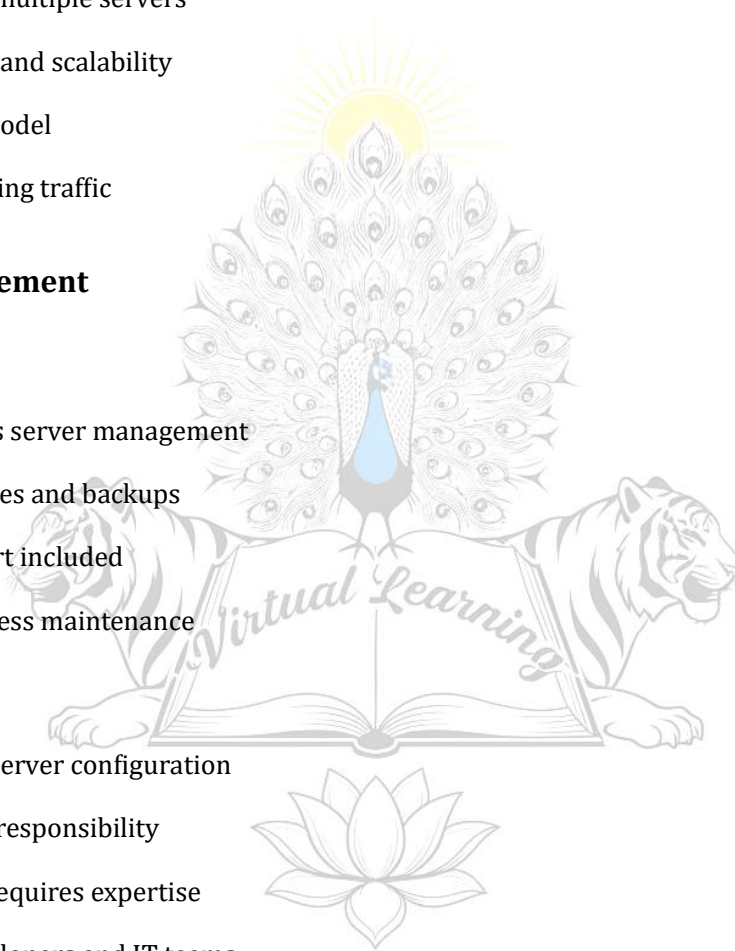
Classification by Management

1. Managed Hosting

- Provider handles server management
- Automatic updates and backups
- Technical support included
- Higher cost but less maintenance

2. Unmanaged Hosting

- Client manages server configuration
- Full control and responsibility
- Lower cost but requires expertise
- Suitable for developers and IT teams



Specialized Hosting Types

1. WordPress Hosting

- Optimized for WordPress sites
- Pre-installed WordPress
- Automatic updates and backups
- WordPress-specific support

2. E-commerce Hosting

- SSL certificates included
- Shopping cart software
- Payment gateway integration
- Enhanced security features

3. Reseller Hosting

- Hosting packages to resell
- White-label solutions
- Multiple cPanel accounts
- Revenue sharing opportunities

This comprehensive guide covers all fundamental aspects of web development, from HTML structure to modern hosting solutions. Each topic provides practical examples and real-world applications essential for building modern websites.

