CMR INSTITUTE OF TECHNOLOGY	USN										

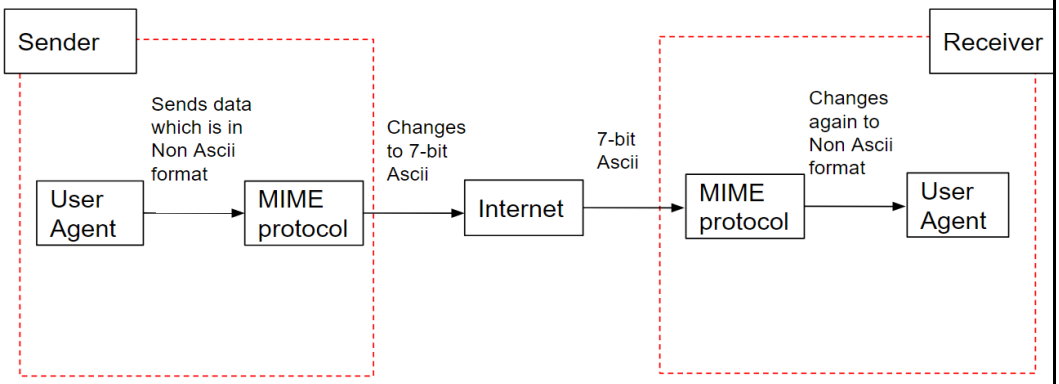
Internal Assessment Test –I, February 2025				
Sub:	**Web Technologies**	**Code:**	**MMC105**	
	Answer Key	**Marks**	**OBE**	
			CO	RBT
1	Briefly explain the following a). URL b). MIME c). Web server d). Web Browser **a) URL** A URL (Uniform Resource Locator) is the unique address used to identify and access resources on the internet, such as web pages, files, or images. It specifies where a resource is located and how to retrieve it. Components of a URL 1. Protocol: Indicates the communication protocol to use (e.g., HTTP, HTTPS, FTP). 2. Domain: Identifies the server hosting the requested resource. Can be a human-readable name (e.g., example.com) or an IP address (e.g., 192.168.1.1). Case insensitive. 3. Port (Optional): Specifies the port number to connect to on the server. Default ports are determined by the protocol: HTTP: Port 80. HTTPS: Port 443. Non-default ports can be specified using a colon after the domain, e.g., http://example.com:888/. 4. Path (Optional): Represents the location of a file or directory on the server. Follows the domain, e.g., http://example.com/**files/image.jpg.** Case-sensitive on most servers (except some Windows-based servers). If not specified, the server serves the default file (e.g., index.html or default.html). 5. Query String (Optional):	[10]	CO1	L1

Provides key-value pairs for additional information, often from user input or form submissions.
Begins with a ? symbol, with key-value pairs separated by &.
Example:
http://example.com/page?username=john&password=abc123.

b) MIME

What is MIME?

MIME stands for Multipurpose Internet Mail Extensions, a standard that extends the format of email to support text in different character sets, attachments such as images, audio, video, and application files, and other multimedia formats. Although originally developed for email, MIME types are now widely used in the context of the Web, where they describe the nature and format of a file or data.



Key Features of MIME

1. Content Description:
   Specifies the type of data being sent.
   Examples: Text, image, video, audio, etc.
2. Encodings:
   Allows non-text data to be encoded in text-based formats for transmission (e.g., Base64).
3. Multipart Messages:
   Supports messages with multiple parts (e.g., an email with both text and an attachment).
4. Cross-Application Usage:
   Used by web browsers, servers, and email clients to handle and interpret file formats correctly.

MIME in HTTP(Hyper Text Transfer Protocol)

In the context of the web, MIME types are used to indicate the type of content being transferred over HTTP. They are sent via the Content-Type header in HTTP responses.

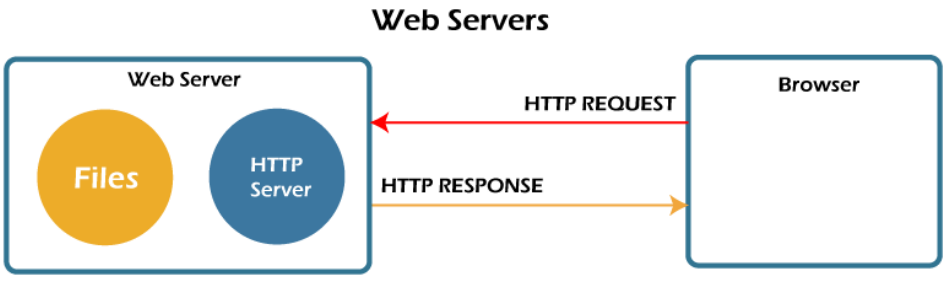
Examples of MIME Types:

MIME Type	Description
text/html	HTML documents
text/css	Cascading Style Sheets (CSS)
application/json	JSON data
application/javascript	JavaScript files
image/jpeg	JPEG images
image/png	PNG images
audio/mpeg	MP3 audio files
video/mp4	MP4 video files
application/pdf	PDF documents
application/octet-stream	Arbitrary binary data

c) Web server

What is a Web Server?

A web server is a computer system or software application that serves content to users over the internet. It processes requests made through the Hypertext Transfer Protocol (HTTP) or its secure version, HTTPS, and delivers the requested resources, such as web pages, images, or files.



How a Web Server Works:

1. Request Handling:

   A user types a URL in their browser or clicks a link.

   The browser sends an HTTP request to the web server.

2. Response Generation:

   The web server processes the request.

   If the resource exists, it sends back the content (e.g., HTML, CSS, JavaScript files).

If not, the server returns an error code like 404 Not Found.

3. Browser Rendering:
   The browser displays the returned data to the user.
   Web Servers and the LAMP Stack: Overview
   A web server is essentially a computer that responds to HTTP requests and delivers web content. It can range from a simple personal computer (like Tim Berners-Lee's first web server) to powerful servers in large-scale web farms.

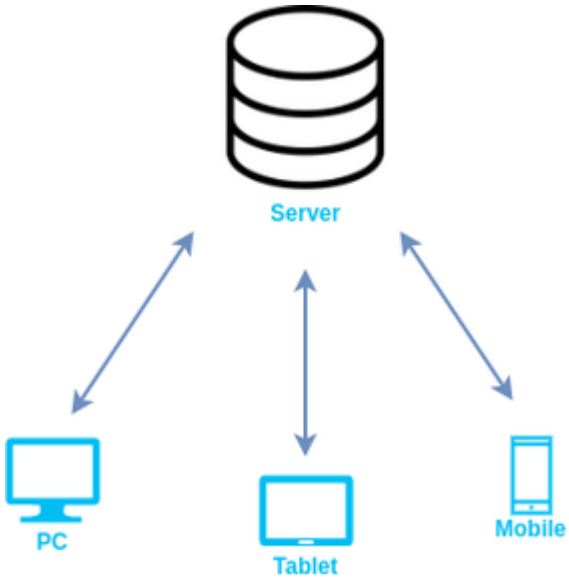
Key Components of a Web Server:

1. Operating Systems (OS):
   Commonly used OS: Linux (preferred for its uptime, lower memory usage, and remote management).
   Windows is also widely used, especially in enterprises adopting Microsoft tools.
2. Web Server Software:
   Apache: Open-source, widely used, supports Linux, Windows, and Mac.
   IIS (Internet Information Services): Microsoft's web server software, tightly integrated with the .NET framework.
3. Database Software:
   For dynamic websites, databases are essential.
   Common open-source options: MySQL, SQLite.
   Proprietary choices: Microsoft SQL Server, Oracle, IBM DB2.
4. Scripting/Server-Side Software:
   LAMP stack often uses PHP, but other options include Python, Ruby on Rails, or ASP.NET.
   PHP is popular for its ease of use, widespread support, and compatibility with HTTP.
   The LAMP Stack:

- Linux (Operating System)
- Apache (Web Server)
- MySQL (Database Management System)
- PHP (Server-Side Scripting Language)

d) Web Browser

A web browser is a software application that facilitates access to the World Wide Web (WWW) by acting as an intermediary between the client (user) and the server. It enables users to request web documents and services from servers, interprets the received data (usually in HTML), and renders it as a user-friendly web page containing text, images, links, and interactive elements. Common web browsers include Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge.



History of Web Browsers

1. WorldWideWeb (1990):
   Invented by Tim Berners-Lee; later renamed Nexus.
   First web browser and editor.
2. Netscape Navigator (1994):
   An advanced version of Mosaic, developed by Marc Andreessen.
   Played a major role in the browser wars of the 1990s.
3. Internet Explorer (1995):
   Launched by Microsoft as the default browser for Windows OS.
   Dominated the market for years.
4. Modern Browsers:
   Mozilla Firefox, Google Chrome, Safari, Opera, and others followed, each offering unique features like speed, privacy, and integration.

2	List and Explain different types of lists in HTML with an example. ● **Lists** In HTML5, lists are used to group related items. There are three main types of lists, each with different attributes and use cases: 1. Unordered List ( ) 2. Ordered List ( ) 3. Description List (<dl>) Each type of list can have various attributes, although the list element itself has only a few that directly impact its appearance or behavior. Let's explore these lists and attributes with examples.	[10]	CO2	L2

1. Unordered List (<ul>)

An unordered list is used when the order of items doesn't matter. By default, items in an unordered list are displayed with bullets.

Attributes:

- type: The type attribute specifies the bullet style. It's applicable only to unordered lists.
  - type="disc" (default, filled circle)
  - type="circle" (hollow circle)
  - type="square" (square bullet)

Example:

```
<ul type="square">
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ul>
```

- Result: A list with square bullets.

---

2. Ordered List (<ol>)

An ordered list is used when the sequence of items is important. It is automatically numbered by default, but the numbering style can be modified.

Attributes:

type: Specifies the numbering style.
- type="1" (default, decimal numbering)
- type="A" (uppercase letters)
- type="a" (lowercase letters)
- type="I" (uppercase Roman numerals)
- type="i" (lowercase Roman numerals)

start: Specifies the starting value for the list. By default, it starts at 1.
reversed: If present, the list will display in reverse order.

Example:

```
<ol type="A" start="5" reversed>
  <li>Item 5</li>
  <li>Item 6</li>
  <li>Item 7</li>
</ol>
```

- Result: An ordered list starting at 5, using uppercase letters, and in reverse order (7, 6, 5).

---

3. Description List (<dl>)

A description list is used to group terms and their definitions. It doesn't have the same numbering or bullet attributes as the other list types.

Attributes:

- <dt>: Specifies the term (item) being defined.
- <dd>: Specifies the description of the term.

Example:

```
<dl>
  <dt>HTML</dt>
  <dd>HyperText Markup Language, used to structure web content.</dd>

  <dt>CSS</dt>
  <dd>Cascading Style Sheets, used to style web content.</dd>

  <dt>JavaScript</dt>
  <dd>A programming language used for dynamic web content.</dd>
</dl>
```

Result: A list of terms with their descriptions.

Nested Lists

You can create nested lists (a list inside another list) to represent more complex structures.

Example (Ordered list nested inside an Unordered list):

```
<ul>
  <li>Groceries
    <ol>
      <li>Milk</li>
      <li>Eggs</li>
      <li>Bread</li>
    </ol>
  </li>
  <li>Household Items
    <ol>
      <li>Toilet Paper</li>
      <li>Soap</li>
    </ol>
```

	     Example (Unordered list inside an Ordered list): - First item - Sub-item 1 - Sub-item 2 - Second item 			
3	Create a Registration form to accept name, gender, date of birth, qualification,address and provide Reset and Submit buttons. <!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>Registration Form</title> </head> <body> Registration Form <form> <label for="name">Name:</label> <input type="text" id="name" name="name" required> <label for="gender">Gender:</label> <input type="radio" id="male" name="gender" value="Male"> Male <input type="radio" id="female" name="gender" value="Female"> Female <input type="radio" id="other" name="gender" value="Other"> Other <label for="dob">Date of Birth:</label> <input type="date" id="dob" name="dob" required> <label for="qualification">Qualification:</label> <input type="text" id="qualification" name="qualification" required> <label for="address">Address:</label> <textarea id="address" name="address" rows="4" cols="30" required></textarea> <input type="reset" value="Reset"> <input type="submit" value="Submit"> </form> </body> </html>	[10]	CO1	L2
4	Explain Arithmetic operators with examples.	[10]	CO2	L2

Java Arithmetic Operators

Arithmetic operators in Java are used to perform mathematical operations such as addition, subtraction, multiplication, division, and modulus.

Operator	Description	Example
`+`	Addition	`a + b`
`-`	Subtraction	`a - b`
`*`	Multiplication	`a * b`
`/`	Division	`a / b`
`%`	Modulus (Remainder)	`a % b`

Examples in Java

```
public class ArithmeticOperators {
    public static void main(String[] args) {
        int a = 20, b = 10;

        // Addition
        System.out.println("Addition: " + (a + b)); // Output: 30

        // Subtraction
        System.out.println("Subtraction: " + (a - b)); // Output: 10

        // Multiplication
        System.out.println("Multiplication: " + (a * b)); // Output: 200

        // Division
        System.out.println("Division: " + (a / b)); // Output: 2

        // Modulus (Remainder)
        System.out.println("Modulus: " + (a % b)); // Output: 0
    }
}
```

5	Explain CSS Margin, padding properties in detail with an example. 1. **Padding** The space between the content and the border. **Size controlled by**: padding property.	[10]	CO2	L1

Padding increases the overall size of the box.

Property:

```
padding-top
padding-right
padding-bottom
padding-left
```

padding: 10px;

Example:-

```
div {
        padding: 10px; /* Adds 10px padding on all sides */
        padding-left: 20px; /* Adds 20px padding to the left only */
}
```

2. **Margin**

The space between the element and its neighboring elements.

Size controlled by: margin property.

Property:-

```
margin-top

margin-right

margin-bottom

margin-left
```

margin: 20px;

Example:

```
div {

margin: 15px; /* Adds 15px margin on all sides */

margin-top: 20px; /* Adds 20px margin to the top only */
```

	}			
6	Discuss on the different ways of including CSS style information to a HTML document. **Levels of CSS** CSS can be added to HTML documents in 3 ways: ● Inline - by using the style attribute inside HTML elements ● Internal - by using a <style> element in the <head> section ● External - by using a <link> element to link to an external CSS file **Inline CSS** An inline CSS is used to apply a unique style to a single HTML element. An inline CSS uses the style attribute of an HTML element. The following example sets the text color of the <h1> element to blue, and the text color of the element to red: **Example:-** <h1 style="color:blue;">A Blue Heading</h1> A red paragraph. **Internal CSS** An internal CSS is used to define a style for a single HTML page. An internal CSS is defined in the <head> section of an HTML page, within a <style> element. The following example sets the text color of ALL the <h1> elements (on that page) to blue, and the text color of ALL the elements to red. In addition, the page will be displayed with a "blue" background color: **Example**:- <!DOCTYPE html> <html> <head> <style>	[10]	CO2	L2

```
body {background-color: blue;}
h1   {color: blue;}
p    {color: red;}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

External CSS

An external style sheet is used to define the style for many HTML pages.

To use an external style sheet, add a link to it in the <head> section of each HTML page:

Example:-

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

The external style sheet can be written in any text editor. The file must not contain any HTML code, and must be saved with a .css extension.

Here is what the "styles.css" file looks like:

styles.css

```
body {
  background-color: powder blue;
}
```

	h1 {
	color: blue;
	}
	p {
	color: red;
	}			

7	Write a HTML document that defines a table with columns and rows. There must be at least three rows. Include cellpadding and cellspacing attributes. <!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>HTML Table Example</title> </head> <body> Sample Table <table border="1" cellpadding="10" cellspacing="5"> <tr> <th>Name</th> <th>Age</th> <th>City</th> </tr> <tr> <td>Alice</td> <td>25</td> <td>New York</td> </tr> <tr> <td>Bob</td> <td>30</td> <td>Los Angeles</td> </tr> <tr> <td>Charlie</td> <td>28</td> <td>Chicago</td> </tr> </table> </body> </html>	[10]	CO1	L3
8	Explain the different form widgets created using <input> tag. • **Text Input (<input> element)**	[10]	CO2	L1

The <input> element is the most versatile and commonly used form element. The type attribute specifies what kind of data the input should accept.
Examples:

- Text Input (type="text"): A single-line text box for the user to enter short text.

<input type="text" id="name" name="name" placeholder="Enter your name" required>

- **Password Input (type="password")**: Similar to text input, but hides the input text (for sensitive data like passwords).

<input type="password" id="password" name="password" placeholder="Enter your password" required>

- **Email Input (type="email")**: Validates that the input matches the format of an email address (e.g., user@example.com).

<input type="email" id="email" name="email" placeholder="Enter your email" required>

- **Number Input (type="number")**: Restricts input to numerical values.

<input type="number" id="age" name="age" min="18" max="100" required>

- **min and max:** Specify the minimum and maximum acceptable values for number inputs.

2. Textarea (<textarea> element)

The <textarea> element allows users to input multi-line text, making it suitable for messages, comments, or descriptions.
<textarea id="message" name="message" rows="4" cols="50" placeholder="Enter your message here" required></textarea>

- rows: Specifies the visible number of lines in the text box.
- cols: Specifies the visible width of the text box in characters.

3. Checkbox (<input type="checkbox">)

Checkboxes allow users to select one or more options. Each checkbox has a unique name attribute, but they can share the same name if they are part of a group.
<input type="checkbox" id="subscribe" name="subscribe" value="yes">
<label for="subscribe">Subscribe to newsletter</label>

4. Radio Button (<input type="radio">)

	Radio buttons are used to allow users to select one option from a group. Radio buttons with the same name are grouped together, so only one option can be selected at a time. <input type="radio" id="male" name="gender" value="male"> <label for="male">Male</label> <input type="radio" id="female" name="gender" value="female"> <label for="female">Female</label> ● name: All radio buttons in the same group must share the same name to ensure only one option is selected.			

9	**Explain CSS Font properties with examples.**
	• **Font**
	CSS font properties allow you to style text on web pages by controlling aspects like font family, size, weight, style, and spacing. Here's a detailed overview of the font-related properties in CSS:
	1. font-family
	Specifies the typeface of the text. It can include multiple fonts as a fallback mechanism. If the browser doesn't support the first font, it will try the next.
	
	p {
	font-family: "Arial", "Helvetica", sans-serif;
	}
	• **Generic Families**:
	○ serif (e.g., Times New Roman)
	○ sans-serif (e.g., Arial)
	○ monospace (e.g., Courier)
	○ cursive (e.g., Comic Sans)
	
	2. font-size
	Defines the size of the text. It can be specified in:
	• Absolute units: px, pt, cm, etc.
	• Keywords: small, medium, large, etc.
	
	h1 {
	font-size: 24px;
	}
	
	3. font-weight
	Controls the boldness of text. Acceptable values:
	• Keywords: normal, bold, lighter, bolder
	• Numerical values: 100 (thin) to 900 (extra-bold)
	p {
	font-weight: bold;
	}
	h2 {
	font-weight: 300; /* Light */
	}
	
	4. font-style
	Defines the style of the font. Options:
	• normal: Default text style
		[10]	CO2	L2

	• italic: Italicized text
	h1 {
	font-style: italic;
	}
	
	5. font-variant
	Used to enable or disable small-caps text (small uppercase letters).
	p {
	font-variant: small-caps;
	}
	
	6. font-shorthand
	The font property allows you to define multiple font properties in one line. The order is important:
	font: font-style font-variant font-weight font-size/line-height font-family;
	p {
	font: italic small-caps bold 16px/1.5 "Georgia", serif;
	}			

10	**Explain the following tags with examples.** **1.<hr> 2. <small> 3. <u> 4. <font> 5. ** **1. <hr> — Horizontal Rule / Line** Inserts a horizontal line on the page to visually separate sections of content. It's an empty tag (no closing tag). Example Section One <hr> Section Two **2. <small> - Small Text** The <small> tag is used to display text in a smaller font size. It can be used to indicate footnotes, fine print, or side information. Example: This is <small>smaller</small> text.	[10]	CO2	L2

3. <u> - Underlined Text

The <u> tag is used to underline text. It is typically used for links but can also be used for general emphasis in text.

Example:

```
<p>This is <u>underlined</u> text.</p>
```

4. <font>

The <font> tag is obsolete in HTML5 — that means it's outdated and you should use CSS instead for styling text. But it still works in older HTML.

Set *font face*, *size*, and *color* of text.

Example (old style)

```
<font size="5" color="green" face="Arial">
  This text is styled with the font tag.
</font>
```

5. <p> — Paragraph Tag

Defines a paragraph of text. It's a *block-level* element — the browser typically adds space *above and below.*

Example

```
<p>This is the first paragraph.</p>
<p>This is the second paragraph.</p>
```