**Module-2 : HTML ASSIGNMENT**

***HTML Basics***

(Theory Assignment)

Question 1: Define HTML. What is the purpose of HTML in web

development?

* HTML(HyperText Markup Language), is the standard markup language used to create and design documents on the World Wide Web. It provides the structure for web pages by using a series of elements or tags that define the content and layout of a webpage.
* It provides the basic building blocks of web pages, such as text, images, links, lists, tables, forms, and more.

**Purpose of HTML:**

* **Structure**: HTML defines the structure of a webpage (e.g., headings, paragraphs, sections).
* **Content Display**: It tells web browsers what content to display and how to display it.
* **Linking**: HTML enables linking between web pages using hyperlinks.
* **Multimedia Integration**: It allows embedding of images, videos, and audio.
* **Form Handling**: HTML provides elements like forms and inputs to collect user data.
* **Foundation for CSS and JavaScript**: HTML is the base layer, which works with CSS (for styling) and JavaScript (for interactivity).

Question 2: Explain the basic structure of an HTML document. Identify the

mandatory tags and their purposes.

* The basic structure of an HTML document follows a standard format that ensures the browser can correctly interpret and display the webpage.

<html>

<head>

<title>Document</title>

</head>

<body>

<h1>This is Heading1</h1>

<p>This is Paragraph</p>

</body>

</html>

**Explanation of Tags:**

* **<!DOCTYPE html>:**
* Declares the document type.
* Tells the browser to use HTML5 standards.
* Must be the first line in the HTML document.
* **<html>:**
* The root element of the HTML document.
* Wraps all the content of the entire webpage.
* **<head>:**
* Contains meta-information about the page (not displayed directly).
* Can include <title>, <meta>, <link>, <style>, and <script>.
* **<title>:**
* Sets the title of the web page (shown in the browser tab).
* Located inside the <head> section.
* **<body>:**
* Contains all the visible content of the web page (text, images, links, etc.).
* Everything inside <body> is what users see on the browser.

Question 3: What is the difference between block-level elements and inline

elements In HTML? Provide examples of each.

* In HTML, elements are categorized as block-level or inline based on how they behave in the document layout.
* **Block-Level Elements:**
* **Definition:** Block-level elements start on a new line and take up the full width available (by default).
* **Behavior:** They stack vertically on top of each other.
* **Used for:** Creating sections, paragraphs, containers, etc.
* **Examples:**
* <div> – generic container
* <p> – paragraph
* <h1> to <h6> – headings
* <ul>,<ol> – lists
* <li> – list items
* <section>,<article>,<header>,<foote> etc.
* **Inline Elements:**
* **Definition:** Inline elements do not start on a newline and take up only as much width as needed.
* **Behavior:** They flow horizontally with surrounding content.
* **Used for:** Styling or formatting small parts within block elements.
* **Examples:**
* <span> – generic inline container
* <a> – hyperlink
* <strong> – bold text
* <em> – italicized text
* <img> – image etc.

Question 4: Discuss the role of semantic HTML. Why is it important for

accessibility and SEO? Provide examples of semantic

elements.

* Semantic HTML uses HTML tags that clearly describe their meaning and purpose both to the browser and developers.

Instead of generic tags like<div> or <span>,semantic tags use meaningful names, such as <header>,<article> , or <footer>

indicate the role of the content they enclose.

* **Why Semantic HTML Matters:**
* **Improves Accessibility:**
* Helps screen readers and assistive technologies understand page structure.
* Improves navigation for users with disabilities.
* Helps visually impaired users navigate content easily (e.g., identifying navigation menus, main content, headings).
* **Boosts SEO (Search Engine Optimization):**
* Search engines use semantic elements to better understand the content and structure of web pages.
* Improves indexing and ranking in search results.
* Enhances content relevance detection by search engines like Google.
* **Better Code Readability & Maintainability:**
* Semantic HTML makes the code more understandable for developers.
* Easier to maintain and debug.
* **Examples of Semantic HTML Elements:**
* <header> - Defines the header of a document or

Section.

* <nav> - Defines navigation links.
* <main> - Defines the main content area.
* <section> - Defines a section in a document.
* <article> - Represents self-contained content.
* <aside> - Content indirectly related to main content.
* <footer> - Defines the footer of a document or

section.

* <figure> - Groups media content (like images).

***HTML Forms***

(Theory Assignment)

Question 1: What are HTML forms used for? Describe the purpose of the

input, textarea, select, and button elements.

* HTML forms are used to collect user input and send it to a server for processing. They are essential in web development for features like user registration, login, searching, feedback submission, and more.
* **Purpose of Common HTML Form Elements:**
* **<input>:**
* Used to get single-line input from the user.
* Common types include:
* text: for plain text input.
* email: for email addresses.
* password: for hidden password input.
* checkbox, radio, number, date, etc.
* Example:
  + <input type="text" name="username" placeholder="Enter your name">

#### **<textarea>:**

#### Used for multi-line text input, like comments or messages.

#### Unlike <input>, it doesn't have a type attribute.

#### Example:

* + <textarea name="message" rows="4" cols="50">Type your message here...</textarea>

#### **<select>:**

#### Creates a dropdown list for users to choose from.

#### Inside it, <option> tags define the available choices.

#### Example:

* + <select name="country">

<option value="in">India</option>

<option value="us">USA</option>

</select>

* **<button>:**
* Represents a clickable button used to submit forms or perform actions.
* It can also contain text or HTML like icons.
* Types include:
  + submit: submits the form.
  + reset: clears form fields.
  + button: performs custom actions with JavaScript.
* Example:
  + <button type="submit">Submit</button>

Question 2: Explain the difference between the GETand POSTmethods / inform submission. When should each be used?

* In HTML forms we can submit data using two main HTTP methods: GET and POST.
* They determine how the data is sent to the server.
* **GET METHOD:**
* How it works:
* Data is appended to the URL as query parameters.
* Example URL after submission:  
  (https://example.com/form?name=John&age=25)
* Characteristics:
* Visible in the browser's address bar.
* Limited in size (due to URL length restrictions).
* Can be bookmarked or shared.
* Cached by browsers.

#### When to use:

#### For retrieving data (read-only actions).

#### When you don’t need to send sensitive information.

#### Example: Search forms, filter options.

#### Example:

<form method="get" action="/search">

<input type="text" name="query">

<button type="submit">Search</button>

</form>

* **POST METHOD:**

#### How it works:

* Data is sent in the body of the HTTP request, not visible in the URL.

#### Characteristics:

#### More secure than GET (data isn’t visible in the URL).

#### No size limitations for most practical use cases.

#### Not cached or bookmarked.

#### When to use:

#### For sending sensitive or large amounts of data.

#### When the form changes data on the server (e.g., creating or updating).

#### Example: Login forms, contact forms, registration

* Example:

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

<input type="text" name="username">

<button type="submit">Submit</button>

</form>

Question 3: What is the purpose of the label element in a form, and how

does it improve accessibility?

* **Purpose of the <label> Element:**
* The <label> element is used to define a caption or description for a form control like <input>, <select>, or <textarea>. It helps users understand what kind of data should be entered.
* Example:

<label for="email">Email Address:</label>

<input type="email" id="email" name="email">

* **Accessibility Benefits:**
* Screen Reader Support:
* Screen readers announce the label when the user focuses on the input.
* Helps visually impaired users understand what each input field is for.
* Larger Click Area:
* Clicking the label will focus or activate the associated input, making forms easier to use, especially on mobile devices.
* Clear Instructions:
* Ensures that each form field is clearly identified, reducing user errors.

***HTML Tables***

(Theory Assignment)

Question 1: Explain the structure of an HTML table and the purpose of

each of the following elements:<table>,<tr>,<th>,<td> and

<thead>.

* HTML tables are used to organize data into rows and columns. Each part of the table has specific elements that define how the table is structured and displayed.
* **Basic Structure of an HTML Table:**

<table>

<thead>

<tr>

<th>Header 1</th>

<th>Header 2</th>

</tr>

</thead>

<tr>

<td>Data 1</td>

<td>Data 2</td>

</tr>

</table>

### **Explanation of Each Element:**

#### **<table>:**

#### This is the main container for the entire table.

#### All other table elements must be placed inside it.

#### It represents a grid of rows and columns.

#### **<tr> (**Table Row**):**

#### Stands for "table row".

#### Used to define a single row in the table.

#### It can contain one or more <td> or <th> elements.

#### **<th> (**Table Header Cell**):**

#### Stands for "table header".

#### Used to create a header cell in a row.

#### Text is bold and centered by default.

#### Helps identify the meaning of the data in a column.

#### **<td> (**Table Data Cell**):**

#### Stands for "table data".

#### Used to create a standard data cell in the table.

#### Each <td> is placed inside a <tr>.

#### **<thead> (**Table Head**):**

#### Groups the header section of the table.

#### Contains one or more <tr> rows with <th> cells.

#### Helps with styling and accessibility.

Question 2: What is the difference between colspan and rowspan in tables?

Provide examples.

* colspan and rowspan are attributes used in HTML tables to merge table cells across columns or rows.
* **colspan (Column Span):**
  + Purpose: Merges multiple columns into one cell.
  + Used in: <th>or <td> elements.
  + Example Use Case: A cell that needs to span across 2 or more columns.
  + Example:

<table border="1">

<tr>

<th colspan="2">Name & Contact</th>

</tr>

<tr>

<td>John</td>

<td>123-456-7890</td>

</tr>

</table>

* **rowspan (Row Span):**
  + Purpose: Merges multiple rows into one cell.
  + Used in: or elements.
  + Example Use Case: A category or label that applies to multiple rows.
  + Example:

<table border="1">

<tr>

<th rowspan="2">Name</th>

<td>John</td>

</tr>

<tr>

<td>Alice</td>

</tr>

</table>

Question 3: Why should tables be used sparingly for layout purposes?

What is a better alternative?

* Tables should be used sparingly for layout purposes because they can lead to several issues:

#### **1. Poor Accessibility:**

#### Screen readers may misinterpret layout tables as data tables, confusing users with disabilities.

#### **2. Difficult to Maintain**

#### Table-based layouts are harder to read and update in the long term.Changing a layout may require editing multiple nested <table> elements.

#### **3. Not Responsive:**

#### Tables do not adapt well to different screen sizes, especially on mobile devices. It becomes hard to implement responsive design (like flexible layouts, stacking, etc.).

* **4. Semantic Meaning:**

Using tables for layout can confuse the semantic meaning of HTML.

Tables are intended for displaying tabular data, and using them for layout can mislead both developers and search engines about the content structure.

* A better alternative for layout purposes is to use CSS (Cascading Style Sheets). CSS provides a flexible and powerful way to control the layout of web pages without compromising accessibility or semantic structure. Techniques such as Flexbox and CSS Grid allow for responsive designs that adapt to various screen sizes while maintaining a clear and meaningful HTML structure.
* **Better Alternative: Use CSS for Layout**
* Modern web design uses CSS layout techniques like:
  + Flexbox:

Great for 1D layouts(rows or columns).

Simple and powerful.

* + Css Grid:

Ideal for 2D layouts(both rows and columns).

Offer full control over layouts.

* + Media Queries:

Help build responsive layouts for all screen sizes.

* Example using css flex instead of table:

<div style="display: flex;">

<div>Left</div>

<div>Right</div>

</div>