MODULE 2 :- HTML IN FRONT-END

1 :- HTML Basic Theory

QUE. 1 DEFINE HTML. What is the purpose of HTML in web development ?

ANS. HTML (Hyper Text Markup Language) is the standard language used to create and structure content on the web. It defines the structure of web pages using a system of tags and elements.

These tags describe the content, such as headings, paragraphs, links, images, and other types of media, and how they should be displayed in a web browser.

✞ **PURPOSE OF HTML :-**

The purpose of HTML (Hyper Text Markup Language) in web development is to structure the content and define the layout of a webpage. It serves as the foundation for all web pages, providing the basic framework for displaying text, images, links, forms, and other elements that make up the content of a website. HTML allows browsers to interpret and render the content in a readable and accessible format for users.

✞ **Key purposes of HTML in web development:-**

- **Structure Content:** HTML defines how the content of a webpage is organized and arranged. It tells the browser how to display text, images, videos, and other elements.
- **Create Links:** HTML is used to create hyperlinks (<a> tags), allowing users to navigate between different pages or external websites.
- **Embed Media:** HTML allows for the embedding of media like images, videos, audio, and animations within a webpage using tags such as <img>, <video>, and <audio>.
- **Form Elements:** HTML is used to create forms (e.g., contact forms, login forms, search bars) using tags like <form>, <input>, <textarea>, and <button>, enabling user interaction with the website.
- **Semantic Structure:** HTML helps to organize content semantically (using tags like <header>, <footer>, <article>, and <section>), making websites more accessible and easier to navigate, especially for search engines and users with disabilities.
- **Compatibility with Other Technologies**: HTML works alongside CSS for design (styling) and JavaScript for dynamic interactions, enabling the creation of fully functional, interactive, and visually appealing websites.

QUE. 2: Explain the basic structure of an HTML document. Identify the mandatory tags and their purposes.

ANS. :- The basic structure of an HTML document consists of a set of elements that define the content and organization of the webpage. Here's an overview of the main components:

✞ **STRUCTURE OF HTML DOCUMENT:-**

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Document</title>
</head> <body>

</body>
</html>
```

✞ **Identify the mandatory tags and their purposes :-**

1. **<!DOCTYPE html>**
   - **Purpose**: This declaration tells the browser that the document is written in HTML5. It ensures the document is interpreted using the correct version of HTML.
   - **Why it's mandatory**: Without it, browsers may fall back to older rendering modes (quirks mode), which can lead to inconsistent display across different browsers.
2. **<html>**
   - **Purpose**: This is the root element of an HTML document, and all other elements must be nested inside it. It represents the start of the HTML document.
   - **Why it's mandatory**: It indicates the start of an HTML document and serves as a container for all content in the document.
3. **<head>**
   - **Purpose**: The <head> element contains metadata about the document, such as the title, character encoding, and links to external resources like stylesheets or scripts.
   - **Why it's mandatory**: While not all content in the <head> is visible to users, it contains critical information for the browser (such as the page title) and for SEO and accessibility purposes.
4. **<title>**

- **Purpose**: Defines the title of the document, which appears in the browser's title bar or tab.
- **Why it's mandatory**: The title is important for accessibility, SEO, and user experience. It helps users identify the page in their browser.

5. **<body>**
    - **Purpose**: The <body> tag contains all the visible content of the webpage, such as text, images, links, and other elements that are rendered by the browser.
    - **Why it's mandatory**: Without the <body>, there would be no content visible on the page. It's the primary section where all user-visible content is placed.

Que 3. : What is the difference between block-level elements and inline elements in HTML? Provide examples of each.

ANS. : Introduction :

[1] Block Level : These are the elements, which structure main part of webpage, by dividing a page into coherent blocks. A block-level element always start with new line and takes the full width of web page, from left to right. These elements can contain block-level as well as inline elements.

[2] Inline : Inline elements are those elements, which differentiate the part of a give text and provide it a particular function. These elements does not start with new line and take width as per requirement. The Inline elements are mostly used with other elements.

- **Difference:-**

Subject	Block Level	Inline
1. Definition	It occupies the full width available, starting on a new line.	It only takes up as much width as necessary, staying in the same line.
2.Nesting Behaviour	It can contain both block and inline elements.	It usually contains only text or other inline elements.
3. Common Usage	It is used to create sections, paragraphs, and containers	It is used to create sections, paragraphs, and containers
4. Width Control	You can have width and height set using CSS.	In this width and height cannot be controlled explicitly (only padding and margins apply).

5. Example	***<h1>…<h6>, <img>, , etc.***	***<td>, <label>, <i>, <b>, <mark>, etc.***

QUE. 4 : Discuss the role of semantic HTML. Why is it important for accessibility and SEO? Provide examples of semantic elements.

ANS. : Meaning/role: Semantic HTML refers to using HTML elements that convey meaning and structure to both browsers and developers. These elements clearly define their purpose, making web content easier to understand, maintain, and optimize for accessibility and SEO. Instead of using generic ** or ** elements, semantic tags like **<article>, <section>,** and **<nav>** provide structure and meaning to a webpage. **A. <u>Accessibility:</u>**

- **Improving User Experience for Everyone:** Screen readers and assistive technologies can better interpret and navigate content. It helps visually impaired users understand page sections, improving user experience. It also provides clear landmarks (<nav>, <header>, <footer>) for easier browsing.
- **Enhances Meaning and Context:** The **<article>** and **<section>** help define content roles, allowing screen readers to announce them properly**.** Tags like **<button>, <submit>, <reset>** tells assistive tools that the element is interactive.
- **Supports Keyboard Navigation:** Semantic elements ensure better tab order and focus states for users who navigate with keyboards. The **<label>** tags improve accessibility for forms by linking text with input fields. **B. <u>SEO:</u>**
- **Boosting Search Engine Ranking:** Search engines like Google prioritize wellstructured, meaningful content. Semantic HTML improves indexing, ranking, and readability. The **<article>, <section>,** and **<aside>** help search engines categorize content properly.
- **Enhances Rich Snippets and SERP Features:** Input like **<time>** and **<address>** provide additional metadata for events and locations. The **<meta>** tags in the **<head>** section improve search visibility.
- **Reduces Unnecessary Usage:** Cleaner and well-structured HTML leads to better indexing and ranking which helps search engine crawlers find relevant content faster. A structured page with proper sections and headings keeps users engaged.

2 :- HTML Form

QUE. 1 : What are HTML forms used for? Describe the purpose of the input, text area, select, and button elements.

ANS. : What is a form and what are HTML forms used for?

- Form is a structure which is made up of label & input to collect data. We can know more about it by dividing it into two parts. The first part is the form that you see on the page itself. Forms are made up of buttons, text fields, and pull-down menus (collectively known as form controls) used to collect information from the user. Forms may also contain text and other elements. The other component of a web form is an application or script on the server that processes the information collected by the form and returns an appropriate response. It's what makes the form work.

- **Elements of form and their purpose:**
- Forms are added to web pages using the <form> element. The form element is a container for all the content of the form, including some number of form controls, such as text entry fields and buttons. It may also contain block elements, (h1, p, and lists, for example), however, it may not contain another form element.

1) **<Input>:** The majority of controls are added to a form using the input element. The functionality and appearance of the input element changes based on the type attribute. The input in web form is used to collect user-input. It can collect text, radio, email, checkbox, password etc.

2) **<textarea> :** At times, you'll want your users to be able enter more than just one line of text. For these instances, use the textarea element that is replaced by a multi-line, scrollable text entry box when displayed by the browser. Unlike the empty input element, the textarea element has content between its opening and closing tags. The content of the textarea element is the initial content of the text box when the form is displayed in the browser. In addition to the required name attribute, the textarea element uses the following attributes:

   Rows: Specifies the number of lines of text the area should display. Scrollbars will be provided if the user types more text than fits in the allotted space.

   Cols: Specifies the width of the text area measured in number of characters.The textarea is also called "Multiline text entry control".

3) **<select>:** The select element displays as a pull-down menu by default when no size is specified or if the size attribute is set to 1. In pull-down menus, only one item may be selected. You add both pull-down and scrolling menus to a form with the select element. Whether the menu pulls down or scrolls is the result of how you specify its size and whether you allow more than one option to be selected. Let's take a look at both menu types. The select element is just a container for a number of option elements. The content of the chosen option element is what gets passed to the web application when the form is submitted. If for some reason you want to send different value than what appears in the menu, use the value attribute to provide an overriding value.

4) **<button>:** There are a number of different kinds of buttons that can be added to web forms. The most fundamental is the submit button. When clicked, the submit button immediately sends the collected form data to the server for processing. The reset button returns the form controls to the state they were in when the form loaded. Both submit and reset buttons are added using the input element. As mentioned earlier, because these buttons have specific functions that do not include the entry of data, they are the only form control elements that do not require the name attribute the form data to the server the form controls to their default settings Submit and reset buttons are straightforward to use. Just place them in the appropriate place in the form, in most cases, at the very end. By default, the submit button displays with the label "Submit" and the reset button is labelled "Reset." Change the text on the button using the value.

QUE. 2 : Explain the difference between the GET and POST methods in form submission. When should each be used?

ANS. : The method attribute the method attribute specifies how the information should be sent to the server. There are only two methods for sending this encoded data to the server: POST or GET indicated using the method attribute in the form element.

We'll look at the difference between the two methods and when should each be used in the following:

✞ **Difference between GET and POST :-**

No.	Get Method	Post Method
(1)	It is by default.	You have to manually select this.
(2)	With the GET method, the encoded form data gets tacked right onto the URL sent to the server. A question mark character separates the URL from the following data.	When the form's method is set to POST, the browser sends a separate server request containing some special headers followed by the data.
(3)	Because the content of the form is in plain sight, GET is not appropriate for forms with private personal or financial information.	Only the server sees the content of this request, thus it is the best method for sending secure information such as credit card or other personal information.
(4)	The GET method is appropriate if you want users to be able to bookmark the results of a form submission (such as a list of search results).	The POST method doesn't provide such a thing because of being secure.
(5)	In addition, because there is a 256 character limit on what can be appended to a URL, GET may not be used for sending a lot of data or when the form is used to upload a file.	The POST method is also preferable for sending a lot of data, such as a lengthy text entry, because there is no character limit as there is for GET.

QUE. 3 : What is the purpose of the label element in a form, and how does it improve accessibility?

ANS. : <Label>: The label element is used to associate descriptive text with its respective form field. This provides important context for users with speech-based browsers.

Purpose : It enhances usability, clarity, and accessibility by explicitly defining what information the user needs to enter. It clearly describes the purpose of an input field, reducing confusion. It helps organize form elements, creating a better user experience.

✞ **How <label> Improves Accessibility :**

1. **Helps Screen Readers Identify Input Fields :** The screen readers announce the **<label>** text when users navigate the form. A properly associated **<label>** ensures the screen reader reads the field name**,** helping users understand what data is expected.
2. **Expands Clickable Area for Users with Motor Disabilities:** The users with limited mobility or tremors may struggle to click on small checkboxes or radio

buttons. When a **<label>** is linked to an input field, clicking on the label also activates the input. This reduces frustration and improves usability**,** especially on touchscreens and small screens so it supports keyboard navigation for users who cannot use a mouse.

3. **Enhances Form Validation & Error Handling:** The Labels help users understand what to enter, reducing mistakes. The screen readers can provide better feedback when errors occur in labelled fields.

3 :- HTML TABLE

QUE. 1 : Explain the structure of an HTML table and the purpose of each of the following elements: <table>, <tr>, <td> and <thead>.

ANS. : Table: HTML tables were created for instances when you need to add tabular material [data arranged into rows and columns] to a web page. Tables may be used to organize calendars, schedules, statistics, or other types of information.

The elements that identify the table [<table>], rows [<tr> for "table row"], and cells [<th> for "table headers," and <td> for "table data"]. Cells are the heart of the table, because that's where the actual content goes. The other elements just hold things together. This is how structure of table works. ✞ **Structure:**

```
<table>
    <tr>
        <th>1 header</th>
    </tr>

    <tr>
        <th>header 2</th>
        <td>data1</td>
        <td>data2</td>
    </tr>
</table>
```

✞ **Purpose of the elements:**

1. **<table>:** The **<table>** element in HTML is used to organize and display structured data in a tabular format consisting of rows **(<tr>)** and columns **(<td>).** Tables improve data readability, especially when combined with headers **(<th>)** for better organization. Additionally, they enhance accessibility when properly formatted with attributes like caption.

2. **<tr>:** The **<tr>** (Table Row) element in HTML is used to define a row within a **<table>.** It acts as a container for table data **(<td>)** and table header **(<th>)** cells, grouping them into a horizontal structure. Each **<tr>** represents a single row of data, and multiple **<tr>** elements are used to create multiple rows in a table.

3. **<td>:** The **<td>** (Table Data) element in HTML defines a single cell within a **<tr>** (table row). It is used to hold data inside a table, making up the body of the table along with other **<td>** elements. Each **<td>** corresponds to a column within a row and can contain text, images, links, or other HTML elements.

4. **<thead>:** The **<thead>** (Table Head) element in HTML is used to group the header content of a table. It typically contains one or more **<tr>** (table row) elements, which in turn hold **<th>** (table header) cells. The **<thead>** section helps organize and separate the table's header from the body **(<tbody>),** improving readability and accessibility.

QUE. 2 : What is the difference between colspan and rowspan in tables? Provide examples.

ANS. : <u>Difference between colspan and rowspan</u>

Colspan	Rowspan
Column spans or colspan is an attribute of <td> & <th>	Row span works in same elements like colspan
It is “1” by default	It is also “1” by default.
It stretches a cell to the right to span over the subsequent columns.	It causes the cell to span downward over several rows.

<table> <tr> <td colspan=”3 Merged Columns</td> <tr> </table>	<table> <tr> <td rowspan="3">Merged Rows</td> <tr> </table>

QUE. 3 : Why should tables be used sparingly for layout purposes? What is a better alternative?

ANS. : The tables are created to display the data, not to create layout for webpage and they should be used sparingly for layout purposes because of following points:

1. **The issue of poor accessibility:** The tables are designed to **structure tabular data,** not for layouts. When it is used for design, **screen readers** misinterpret the content, making navigation difficult for visually impaired users. This creates so much confusing experience, as assistive technologies assume table elements represent structured data rather than layout components.
2. **Table being not responsive:** They do not adapt well to screen sizes which differs from devices to devices, especially on mobile devices. The table-based designs often require horizontal scrolling, making the content difficult to view on smaller screens. This leads to poor user experience and readability issues.
3. **SEO and Semantic Issues:** The search engines prioritize semantic HTML to understand web pages better. When the tables are being misused for layout, it reduces the SEO friendliness of the site, making it harder for search engines to properly index content. This can also negatively impact ranking and discoverability of a web page**.**

✞ **Better Alternative :-**

HTML5 :HTML5 introduced new semantic elements and improved CSS layout techniques, making it a superior alternative to table-based layouts. Unlike the tables, which were originally designed for displaying tabular data**,** HTML5 provides a more flexible, responsive, and accessible way to structure web pages. Here are some points in detail:

1. **Use of Semantic Elements:** HTML5 introduces semantic tags like **<header>, <nav>, <section>, <article>, <aside>** and **<footer>.** These elements provide a structure**,** making web pages more accessible, SEO-friendly, and easier to understand for both developers and search engines.
    A. **Header**: - The header element represents a group of introductory or navigational aids. A header element is intended to usually contain the section’s heading (an h1-h6 element or an hgroup element), but this is not required.

B. Nav: - The nav element represents a section of a page that links to other pages or to parts within the page a section with navigation links. Not all links of a document must be in an element. The element is intended only for major block of navigation links. Browsers, such as screen readers for disabled users, can use this element to determine whether to omit the initial rendering of this content.

C. Section: - The section element represents a generic section of a document application. A section in this context is a thematic grouping content typically with a heading.

D. Article:- This article element represents a self- contained composition in a document, page, application, or site and that is intended to be independently distributable or reusable, e.g. In syndication

E. Aside: - The aside element represents a section of a page that consists of content that is tangentially related to the content around the aside element, and which could be considered separate from that content.

F. Footer:-The footer element represents a footer for its nearest ancestor sectioning content or sectioning root element. A footer typically contains information about its section such an s who wrote it. Links to related documents. Copyright data and the like.

2. Improved Responsiveness: HTML5 allow layouts to be fully responsive, adapting automatically to different screen sizes using media queries. This makes the web pages mobile-friendly without requiring any complex table structures that break on small screens. For ex. <meta name="viewport" content="width=device-width, initial-scale=1.0">

3. Better Accessibility & SEO: Semantic HTML improves accessibility for screen readers and makes content easier for search engines to index. Unlike tables, which confuse assistive technologies, HTML5 elements clearly define sections, improving navigation and usability**.**