Module 2 – Frontend – HTML

HTML Basics

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

- HTML (HyperText Markup Language) is the standard language used to create and structure content on the web.
- HTML is a **markup language** that uses **tags** to define the structure and layout of a webpage, such as headings, paragraphs, links, images, tables, forms, and more.

Purpose of HTML in Web Development:

- **Structure Content**: HTML organizes content into elements like headings (<h1>), paragraphs (<p>), lists (<ul>, <li>), etc.
- **Display Information**: It tells the browser how to display text, images, and multimedia.
- **Create Links (Hypertext)**: HTML enables linking to other pages or websites using <a> tags.
- **Embed Multimedia**: You can add images, videos, audio, and other media using tags like <img>, <video>, and <audio>.
- **Form Handling**: HTML provides form elements (<form>, <input>, <textarea>, etc.) to collect user input.
- **Foundation for CSS and JavaScript**: HTML provides the base structure that CSS styles and JavaScript makes interactive.

- **HTML is the backbone of every webpage**, helping browsers understand and present web content correctly.

2.Explain the basic structure of an HTML document. Identify the mandatory tagsand their purposes.

- An HTML document follows a specific structure that helps browsers understand and render the content correctly.

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

Tag	Purpose
<!DOCTYPE html>	Declares the document type and HTML version (HTML5 here).
<html>	Root element of the HTML page; wraps all content.
<head>	Contains meta information, title, links to CSS, scripts, etc.
<title>	Sets the title of the webpage (shown in browser tab).
<body>	Contains all the visible content of the webpage (text, images, etc.).

Optional but Common Elements in <head>:

- <meta>: Defines metadata (character set, viewport, description, etc.).
- <link>: Links to external stylesheets.
- <script>: Adds JavaScript functionality.

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

- HTML elements are mainly categorized into two types based on how they behave in the document layout:

1. **Block-Level Elements**

Definition: Block-level elements take up the **full width** available and always start on a **new line**.

They create "blocks" of content.

Examples:

<p>
<h1> to <h6>
, <ol>,
<table>
<section>, <article>

2. **Inline Elements**

Definition: Inline elements only take up **as much width as necessary** and do **not** start on a new line.

They appear within lines of text.
Examples:

<a>
,
<img>
<label>, <input> (though input has some special behavior)

4.Discuss the role of semantic HTML. Why is it important for accessibility andSEO? Provide examples of semantic elements.

Role of Semantic HTML

- **Semantic HTML** uses **meaningful tags** to clearly describe the structure and content of a webpage. These tags tell both the browser and developers **what each section of the page means**, not just how it looks.

Why Semantic HTML Is Important

1. Accessibility
- Screen readers and assistive technologies use semantic tags to **understand page structure**.
- Helps users with disabilities **navigate content** more easily.
- Example: <nav> lets screen readers jump directly to navigation links.

2. SEO (Search Engine Optimization)
- Search engines use semantic tags to better **index and rank** content.
- It improves how your content is understood by Google and other search engines.

- Example: <article> or <header> helps bots understand main content sections.

3. Readability and Maintainability

- Makes code easier to read and maintain for developers.
- Clearer layout structure helps teams collaborate better.

Examples of Semantic Elements

Tag	Meaning / Purpose
<header>	Intro section or heading of a page/section
<nav>	Navigation links area
<main>	Main content of the page
<section>	A standalone section of content
<article>	Independent content (e.g., blog post, news)
<aside>	Sidebar or related content
<footer>	Footer area with copyright or links
<figure> / <figcaption>	Image with a caption

Non-Semantic Tags (for comparison)

Tag	Purpose
	Generic container
	Generic inline container

HTML Forms

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 commonly used in:
  - User registration
  - Login pages
  - Contact forms
  - Surveys
  - Search bars
  - File uploads
- Forms are created using the <form> tag, and include various input elements to collect different types of data.

Common Form Elements and Their Purposes

Element	Purpose
<input>	Used for single-line user input, such as text, email, password, checkbox, radio, etc.
<textarea>	Used for multi-line text input, such as comments or messages.
<select>	Creates a dropdown menu to select one or more options.
<button>	Creates a clickable button, often used to submit the form or perform actions.

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

Feature	GET	POST
Data in URL	Yes (visible)	No (hidden in body)
Security	Less secure	More secure

Bookmarkable	Yes	No
Max Data Size	Limited	Larger (depends on server)
Use Case	Search, filters, simple forms	Login, contact, file upload, secure forms

- Use GET for actions that do not change data (like search).
- Use POST for actions that change or submit data (like registration or login).

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

- The <label> element is used to define a text description for a form control (like an <input>, <select>, or <textarea>).

Why Use <label>?

- **Identifies Inputs Clearly**
  It shows the user what information to enter in each field.
- **Improves Accessibility**
  Screen readers **read the label text** when focusing on a form input, helping visually impaired users understand what the field is for.
  Clicking on the label automatically **focuses the associated input**, improving usability for all users.

- The <label> element makes forms more **accessible**, **user-friendly**, and **semantically correct**. It ensures that every form control is clearly identified, especially for users relying on assistive technologies.

HTML Tables

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

- An HTML table is used to display data in rows and columns. The structure is made up of several key elements, each with a specific purpose.

- **Key Table Elements and Their Purposes**

Tag	Purpose
`<table>`	Defines the start and end of the table. It wraps all table content.
`<tr>`	Table row: groups one row of cells (header or data).
`<th>`	Table header cell: used for **column or row headings**, usually bold and centered.
`<td>`	Table data cell: contains **regular data** in the table.
`<thead>`	Groups the **header section** (usually rows with `<th>` elements). Helps with styling and accessibility.

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

- colspan and rowspan are attributes used in <td> or <th> elements to merge cells across columns or rows, respectively.

Attribute	Merges Across	Used In	Layout Effect
colspan	Columns	Horizontal	Makes cell wider
rowspan	Rows	Vertical	Makes cell taller

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

- In the early days of web development, developers used <table> elements to design page layouts (like headers, sidebars, and footers). However, this is not recommended today.

- **Reasons to Avoid Using Tables for Layout:**

Issue	Explanation
❌ **Not Semantic**	Tables are meant for **tabular data**, not for layout. Using them wrongly confuses screen readers and browsers.
❌ **Poor Accessibility**	Screen readers may misinterpret the structure, making it harder for users with disabilities to navigate.
❌ **Hard to Maintain**	Table layouts are **complex and messy** to edit, especially for responsive design.
❌ **Not Mobile-Friendly**	Tables don't adapt well to different screen sizes, making layouts **break on small devices**.
❌ **Longer Load Times**	More HTML code is needed compared to modern layout methods.

- **Better Alternative: Use CSS with Semantic HTML**

Use elements + CSS layout techniques:

1. **Flexbox** – for **1D layouts** (rows or columns)
2. **CSS Grid** – for **2D layouts** (rows **and** columns)