Module 2 – Mernstack - HTML

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

HTML is a markup language used to structure content on the web. It tells the web browser how to display text, images, links, and other elements on a webpage.

The main purpose of **HTML (HyperText Markup Language)** is to **create the structure and content of webpages**. It acts as the **foundation** of every website.

Key Roles of HTML:

1. **Defines the Structure of a Webpage**
   HTML uses tags (like <h1>, <p>, <div>, etc.) to organize content into headings, paragraphs, lists, sections, and more.
2. **Displays Content in the Browser**
   Web browsers read HTML files and render the content visually for users.
3. **Embeds Multimedia**
   HTML allows adding images (<img>), audio (<audio>), video (<video>), and more.
4. **Creates Links Between Pages**
   Using <a> tags, HTML enables navigation between pages or websites (hyperlinks).
5. **Forms for User Input**
   HTML provides form elements (<form>, <input>, <button>) for user interaction and data submission.
6. **Works with CSS and JavaScript**
   - HTML defines the content
   - CSS styles the content

- JavaScript adds interactivity

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

Basic HTML Structure:

```
<!DOCTYPE html>
<html>
 <head>
        <title>My Webpage</title>
 </head>
 <body>
        <h1>Hello, World!</h1>
         <p>This is a paragraph.</p>
</body>
</html>
```

Mandatory Tags and Their Purposes:

Tag	Purpose
<!DOCTYPE html>	Declares the document type and version of HTML (HTML5 in this case).
<html>	Root tag of the HTML document; all content is enclosed inside it.
<head>	Contains metadata (info about the page), such as title, links to CSS, etc.
<title>	Sets the title of the webpage (shown in browser tab).
<body>	Contains visible content that appears on the webpage (text, images, etc.).

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

1. Block-level Elements

Block-level elements always start on a new line and take up the full width available, stretching across the container.

Key Features:

- Begin on a new line
- Occupy the entire width of the parent container
- Can contain other block-level and inline elements

Examples:

```
<h1>This is a heading</h1>

<p>This is a paragraph</p>

<div>This is a block element</div>

<ul>

        <li>List item</li>

</ul>
```

2. Inline Elements

Inline elements do not start on a new line. They appear within a line, only taking up as much space as needed.

Key Features:

- Do not start on a new line
- Only take up the width of their content
- Usually used inside block elements

Examples:

```
<p>This is <strong>important</strong> text.</p>
```

```
<a href="#">Click here</a>
<span>This is inline</span>
```

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

What is Semantic HTML?

Semantic HTML uses meaningful tags that clearly describe the purpose of the content they contain.
Instead of using generic tags like <div> or <span>, semantic elements use tags like <header>, <article>, <nav>, etc., which describe their meaning both to the browser and to developers.

Why is Semantic HTML Important?

1. Accessibility

- Helps screen readers and assistive technologies understand the page structure.
- Improves navigation for visually impaired users.
- Example: <nav> tells screen readers this section is for navigation.

2. SEO (Search Engine Optimization)

- Search engines can better understand and index the content.
- Semantic tags highlight important content, improving ranking in search results.
- Example: <article> helps search engines detect main content**.**

3. Maintainability & Readability

- Easier for developers to read, update, and collaborate on code.
- Enhances the logical structure of HTML.

- **Examples of Semantic HTML Elements**

Semantic Tag	**Purpose**
<header>	Defines a page or section header
<nav>	Defines navigation links
<main>	Indicates the main content of the page
<article>	Represents a self-contained piece of content
<section>	Defines a section in the document
<aside>	Represents side content (ads, sidebars)
<footer>	Defines a footer for a section or page
<figure> & <figcaption>	For images with captions

5. What are HTML forms used for? Describe the purpose of the input, textarea, select, and button elements.

HTML Forms: Purpose

HTML forms are used to collect user input and send data to a server for processing. They are commonly used in:

- User registrations
- Login pages
- Feedback or contact forms
- Search fields
- Surveys, and more

A form typically uses the <form> tag and includes various form controls like input fields, text areas, dropdown menus, and buttons.

Key HTML Form Elements

1. <input>

- Purpose: Collect single-line input such as text, numbers, passwords, emails, etc.
- Types include:
    - text – for plain text input
    - password – hides the characters
    - email – expects an email address
    - number – allows numeric input
    - checkbox and radio – for selection options
    - submit – to submit the form

Example:

```
<input type="text" name="username" placeholder="Enter your name">
```

2. <textarea>

- Purpose: Collect multi-line text input (e.g., comments, descriptions).
- Customizable in terms of rows and columns.

Example:

```
<textarea name="message" rows="5" cols="30">Enter your message
here</textarea>
```

3. <select>

- Purpose: Create a drop-down list of options.
- Used when users must choose from a set of predefined choices.

Example:

```
<select name="country">
```

```
  <option value="india">India</option>
  <option value="usa">USA</option>
  <option value="uk">UK</option>
</select>
```

4. <button>

- Purpose: Trigger actions, typically to submit the form or reset the inputs.
- More flexible than <input type="submit"> because it can contain HTML and scripts.

Example:

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

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

Feature	GET Method	POST Method
Data Location	Appended to the URL as query string	Sent in the body of the HTTP request
Visibility	Visible in the browser address bar	Hidden from the URL
Security	Less secure (not suitable for sensitive data)	More secure (suitable for sensitive data)
Data Length	Limited (approx. 2000 characters)	No significant limit (can handle large data)
Caching	Can be cached	Not cached by default
Bookmarking	Can be bookmarked	Cannot be bookmarked

Feature	GET Method	POST Method
Use Case	When data retrieval is needed (e.g., search forms)	When data needs to be submitted securely (e.g., login forms)
Form Default?	Yes (if method is not specified in <form>)	No (must be explicitly defined)

When to Use

- **Use GET:**
  When submitting non-sensitive data and you want the data to be visible in the URL (e.g., search queries, filters).
- **Use POST:**
  When submitting sensitive data (e.g., passwords), large inputs (like messages), or modifying data on the server (e.g., register, login).

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

Purpose of the <label> Element in a Form

The <label> element is used to identify and describe a form control (like an <input>, <textarea>, or <select>), helping users understand what kind of data is expected.

Main Purposes:

1. **Describes the Input Field**
   It provides a text label for form fields, making the form easier to read and understand.
2. **Links Text to Input Control**
   When properly associated (using for="inputID"), clicking on the label will automatically focus or activate the related input field.

3. **Improves Accessibility**
   Screen readers read the label aloud when users focus on the input, which helps visually impaired users understand the purpose of each form element.

How It Improves Accessibility:

Feature	**Benefit**
Screen reader support	Helps users who rely on screen readers understand form fields clearly
Clickable labels	Easier for all users to activate inputs by clicking the label
Better form structure	Semantically connects inputs with their purpose

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

Structure of an HTML Table

An HTML table is used to display data in rows and columns. It is structured using a set of specific tags that define the table and its content.

Key HTML Table Elements and Their Purpose

Element	**Description & Purpose**	**Example**
<table>	The container element that defines the entire table structure.	<table> ... </table>
<tr>	Stands for table row; groups table cells into a row.	<tr> ... </tr>
<th>	Stands for table header; used for column or row headings.	<th>Student Name</th>

Element	Description & Purpose	Example
<td>	Stands for table data; holds the actual data in the cell.	<td>John Doe</td>
<thead>	Groups the header content of the table (usually contains <th>). Improves structure and accessibility.	<thead> ... </thead>

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

Attribute	Purpose	Direction	Used On	Example
colspan	Merges **multiple columns** into one	Horizontal	<td> / <th>	<th colspan="2">Name</th>
rowspan	Merges **multiple rows** into one	Vertical	<td> / <th>	<td rowspan="2">Alice</td>

Example of colspan

```
<table border="1">
     <tr>
            <th colspan="2">Full Name</th>
     </tr>
      <tr>
             <td>First Name</td>
           <td>Last Name</td>
     </tr>
sss</table>
```

Example of rowspan

```
<table border="1">
    <tr>
        <th rowspan="2">Student</th>
        <td>Alice</td>
    </tr>
    <tr>
        <td>Bob</td>
    </tr>
</table>
```

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

HTML tables were once used to control the layout of web pages, but today, this practice is considered outdated and problematic.

Reasons to Avoid Using Tables for Layout

Problem	Explanation
Not Semantic	Tables are meant for displaying tabular data, not for layout design.
Accessibility Issues	Screen readers may misinterpret layout tables as data, confusing users.
Hard to Maintain	Layout tables are complicated to edit, especially with nested structures.
Not Mobile-Friendly	Tables do not adapt well to different screen sizes or responsive design.
Mixes Content and Design	Using tables for layout mixes structure (HTML) with style, which is bad practice.

Better Alternative: CSS (Cascading Style Sheets)

Modern websites use CSS for layout instead of tables. CSS separates content from design and allows for flexible, accessible, and responsive layouts.

Recommended CSS Techniques:

CSS Method	Purpose	Best Used For
Flexbox	One-dimensional layouts (row/column)	Navbars, forms, alignment
Grid	Two-dimensional layouts	Complex page structures, dashboards
Media Queries	Responsive design for devices	Mobile, tablet, and desktop views