Module 2 – MERN stack – HTML

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

HTML stands for Hyper Text Markup Language**,** which is the core language used to structure content on the web.

- It is a markup language, not a programming language.
- It is a **static language**, meaning it does not inherently provide interactive features but can be combined with CSS for styling and JavaScript for interactivity.

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

Structure :

```
<!DOCTYPE html>
<html>
<head>
   <title>My First Webpage</title>
</head>
<body>
   <h1>Welcome to My Webpage</h1>
   <p>This is my first paragraph of text!</p>
</body>
</html>
```

Explaination:

Tag	Purpose
<!DOCTYPE html>	Declares the document type and HTML version (HTML5 here).
<html>	Root element that wraps the entire HTML document.
<head>	Contains meta-information (not visible to users), like title, charset.
<title>	Sets the page title (shown in browser tab). Must be inside <head>.
<body>	Contains the visible content of the web page (text, images, etc.).

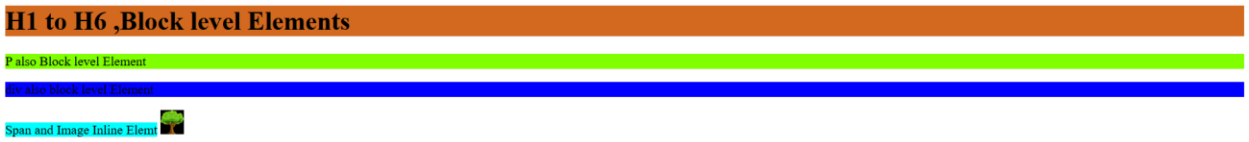
3)What is the difference between block-level elements and inline elements in HTML? Provide examples of each.

Inline Elements	Block Elements
Inline elements occupy only sufficient width required.	Block Elements occupy the full width irrespective of their sufficiency.
Inline elements don't start in a new line.	Block elements always start in a line.
Inline elements allow other inline elements to sit behind.	Block elements doesn't allow other elements to sit behind
Inline elements don't have top and bottom margin	Block elements have top and bottom margin.

EX:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Assignment Purpose</title>
  </head>
  <body>
    <h1 style="background-color: chocolate">H1 to H6, Block level Elements</h1>
    <p style="background-color: chartreuse">P also Block level Element</p>
    <div style="background-color: blue">
      <p>div also block level Element</p>
    </div>

    <span style="background-color: aqua">Span and Image Inline Element</span>
    <img
      src="tree-2027899_960_720.webp"
      alt="IMG"
      height="30px"
      style="background-color: black"
    />
  </body>
</html>
```

Output:



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

Semantic elements = elements with a meaning.

A semantic element clearly describes its meaning to both the browser and the developer.

Examples of **non-semantic** elements: and - Tells nothing about its content.

Examples of **semantic** elements: <img>, <table>, and <article> - Clearly defines its content.

◆ 1. Importance for Accessibility

Point	Explanation
Screen Readers	Semantic tags help screen readers understand page structure. For example, <nav> tells it this is the navigation section.
Keyboard Navigation	Users who rely on keyboard navigation can skip to relevant sections using semantic elements.
Logical Document Structure	Tags like <header>, <main>, and <footer> help assistive tools present content in a clear, structured way.

◈ 2. Importance for SEO (Search Engine Optimization)

Point	Explanation
Better Indexing by Search Engines	Semantic tags help search engines understand what the page is about (e.g., <article> = content block).
Improves Ranking	Pages with clear structure and relevant keywords in semantic tags can rank better in search results.
Rich Snippets	Tags like <article>, <time>, and <address> can contribute to enhanced search listings like featured snippets.

Elements:

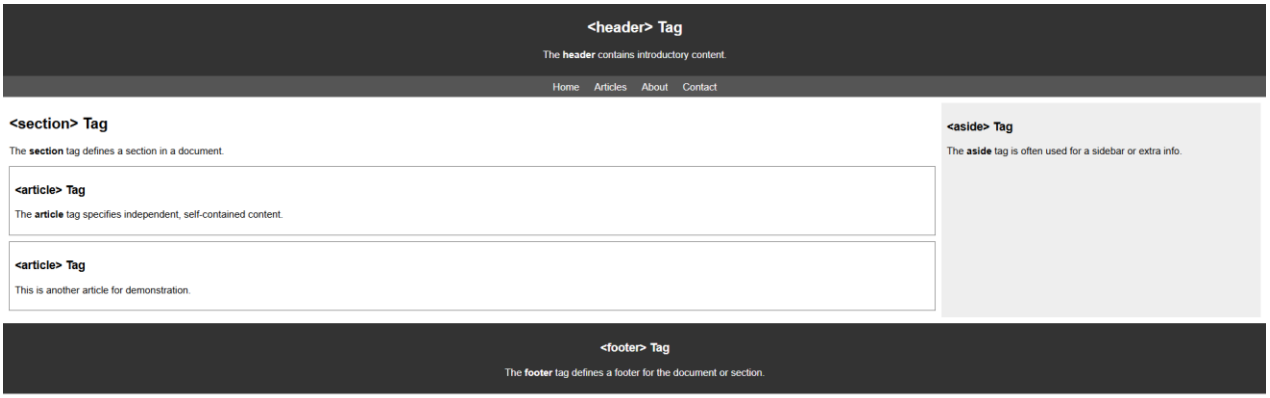
A) The <article> tag specifies independent, self-contained content.
B) The <aside> content is often placed as a sidebar in a document.
C) The <footer> tag defines a footer for a document or section.
D) The <header> element represents a container for introductory content or a set of navigational links.
E) The <main> tag specifies the main content of a document.
F) The <section> tag defines a section in a document.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Semantic Tags Example</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
    }
    header, footer {
      background: #333;
      color: white;
      padding: 10px;
      text-align: center;
    }
    nav {
      background: #555;
      padding: 8px;
      text-align: center;
    }
    nav a {
```

```
      color: white;
      text-decoration: none;
      margin: 0 10px;
    }
    main {
      display: flex;
      padding: 10px;
    }
    section {
      flex: 3;
      margin-right: 10px;
    }
    article {
      border: 1px solid gray;
      padding: 8px;
      margin-bottom: 10px;
    }
    aside {
      flex: 1;
      background: #eee;
      padding: 8px;
    }
  </style>
</head>
<body>

  <header>
    <h2>&lt;header&gt; Tag</h2>
    <p>The <strong>header</strong> contains introductory content.</p>
  </header>

  <!-- nav added here -->
  <nav>
    <a href="#">Home</a>
    <a href="#">Articles</a>
    <a href="#">About</a>
    <a href="#">Contact</a>
  </nav>

  <main>
    <section>
      <h2>&lt;section&gt; Tag</h2>
      <p>The <strong>section</strong> tag defines a section in a document.</p>

      <article>
```

```
        <h3>&lt;article&gt; Tag</h3>
        <p>The <strong>article</strong> tag specifies independent, self-contained
content.</p>
      </article>

      <article>
        <h3>&lt;article&gt; Tag</h3>
        <p>This is another article for demonstration.</p>
      </article>
    </section>

    <aside>
      <h3>&lt;aside&gt; Tag</h3>
      <p>The <strong>aside</strong> tag is often used for a sidebar or extra
info.</p>
    </aside>
  </main>

  <footer>
    <h3>&lt;footer&gt; Tag</h3>
    <p>The <strong>footer</strong> tag defines a footer for the document or
section.</p>
  </footer>

</body>
</html>
```

Output:



HTML TABLE

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

- HTML tables allow web developers to arrange data into rows and columns.
- Each table cell is defined by a `<td>` and a `</td>` tag.
- Each table row starts with a `<tr>` and ends with a `</tr>` tag.
- Sometimes you want your cells to be table header cells. In those cases use the `<th>` tag instead of the `<td>` tag
- The `<thead>` tag is used to group header content in an HTML table

```
<!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>
    <table border="1px solid black" cellspacing="0" cellpadding="10">
      <thead style="color: green;">
        <tr>
          <th>Month</th>
          <th>Savings</th>
        </tr>
      </thead>
      <tbody style="color: blueviolet;">
        <tr>
            <td>January</td>
            <td>80$</td>
        </tr>
        <tr>
            <td>February</td>
            <td>100$</td>
        </tr>
      </tbody>
      <tfoot style="color: brown;">
        <tr>
            <td>Sum</td>
            <td>180$</td>
        </tr>
      </tfoot>
    </table>
  </body>
</html>
```

Output:

Month	Savings
January	80$
February	100$
Sum	180$

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

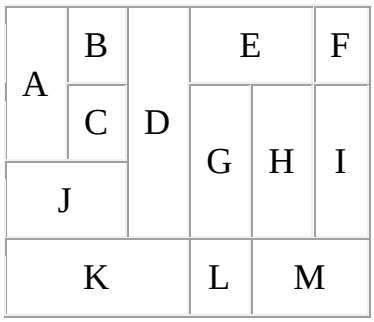
- Colspan=Allows a single table cell to span the width of more than one cell or column.
- Rowspan=Allows a single table cell to span the height of more than one cell or row.

Attribute	Purpose	Example Use Case
rowspan	Merges **rows** vertically	To make a cell span across multiple **rows**
colspan	Merges **columns** horizontally	To make a cell span across multiple **columns**

```
<!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>
    <table border="1" cellspacing="0" cellpadding="10">
      <tr align="center">
        <td rowspan="2" >A</td>
        <td>B</td>
        <td rowspan="3">D</td>
        <td colspan="2">E</td>
        <td>F</td>
      </tr>
      <tr align="center">
        <td>C</td>
        <td rowspan="2">G</td>
        <td rowspan="2">H</td>
        <td rowspan="2">I</td>
      </tr>
      <tr align="center">
```

```
        <td colspan="2">J</td>
      </tr>
      <tr align="center">
        <td colspan="3">K</td>
        <td>L</td>
        <td colspan="2">M</td>
      </tr>
    </table>
  </body>
</html>
```

Output:



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

Using HTML **tables for layout** (like placing elements side by side or aligning content) is discouraged for several reasons:

- Tables are meant to show **data**, like in rows and columns (like marksheets, schedules).
- Using them to design page layout (like placing images or buttons) makes the code **messy and hard to change**.
- It becomes **difficult for blind people using screen readers** to understand the page.
- Tables **don't work well on mobile** — the layout may break or look bad.
- The website may load **slower** if too many tables are used.

Better alternative:

Use **CSS** for layout. It's a better and modern way to design pages.

Some CSS tools:

- **Flexbox** – Easy for placing things side by side.
- **Grid** – Good for creating complex layouts.
- **Media queries** – Helps the layout change based on screen size (mobile, tablet, desktop).

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 on web pages and send it to a server for processing.

Example Uses of Forms:

- Login or Signup forms
- Contact us pages
- Product orders or payments
- Surveys or quizzes

Basic Form Elements:

- <input> – for text, password, numbers, etc.
- <textarea> – for long messages
- <select> – dropdown menus
- <button> or <input type="submit"> – to send the form
- <form> – the container that holds all these

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

Feature	GET Method	POST Method
Data visibility	Data is visible in the URL	Data is hidden from the URL
Security	Less secure (not for sensitive data)	More secure (better for passwords, personal data)
Data limit	Limited (usually up to 2048 characters)	No practical limit (can send large data)
Speed	Faster (used for quick requests)	Slightly slower (due to extra data processing)
Bookmarkable URL	Yes – URL can be bookmarked with data	No – data is not stored in URL
Use case	Search forms, filters, non-sensitive data	Login forms, registration, secure data submission
Caching	Can be cached	Cannot be cached
Idempotent (safe)?	Yes (does not change data)	No (used to create or update data)
Data sent via	URL query string (e.g., ?name=John)	Message body of HTTP request

- **Use GET** when:
    - You want to **fetch or search data**
    - The data is **not sensitive**
    - You want to **bookmark/share the result**
- **Use POST** when:
    - You want to **submit, save, or update** data
    - The data is **private or secure**
    - The form contains **a lot of data** (e.g., file uploads)

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 caption for form inputs like textboxes, checkboxes, radio buttons, etc.

Feature	Explanation
Improves usability	When users click on the label, the related input field gets focused (easier to use)
Better accessibility	Screen readers read the label with the input, helping visually impaired users
Clear identification	Labels tell users what to enter in each field, avoiding confusion

EX:

```
<label for="email">Email:</label>
<input type="email" id="email" name="email">
```

Output:

Email:

Email: