Module 2 – Mernstack – HTML

• HTML Basics

Theory Assignment:

Question 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.

Purpose:
HTML provides the basic framework of a webpage by organizing text, images, links, and other elements. It allows browsers to interpret and display content in a structured format.

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

The basic structure of an HTML document includes the following mandatory tags:

1. **`<!DOCTYPE html>`**: Declares the document type as HTML5.
2. **`<html>`**: The root element that contains all the content of the page.
3. **`<head>`**: Contains metadata (e.g., title, links to stylesheets, etc.) for the document.
4. **`<title>`**: Specifies the title of the webpage, displayed on the browser tab.
5. **`<body>`**: Contains the visible content of the webpage, such as text, images, and links.

Purpose of each tag:

- `<!DOCTYPE html>` ensures compatibility with HTML5.
- `<html>` wraps the entire HTML content.
- `<head>` provides information about the webpage, not visible to users.
- `<title>` sets the browser tab title.
- `<body>` defines what users see on the webpage.

Question 3: What is the difference between block-level elements and inline elements inHTML? Provide examples of each.

Block-level elements:

- These occupy the full width of their parent container and start on a new line.
- Examples: `<div>`, `<p>`, `<h1>`, `<ul>`, `<section>`

Inline elements:

- These only occupy the width of their content and do not start on a new line.
- Examples: `<span>`, `<a>`, `<img>`, `<strong>`, `<em>`

Key difference: Block level elements occupys the whole block even if the space is not needed, and inline elements are precise which only occupys the part which it needs and not more than that

Question 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 define the structure and content of a webpage, making it easier for browsers, search engines, and assistive technologies to understand.

Importance for Accessibility and SEO:

- **Accessibility**: Helps screen readers and assistive tools interpret content, improving the user experience for people with disabilities.
- **SEO**: Enhances search engines' ability to index and rank webpages by providing clear context and structure.

Examples of Semantic Elements:

- `<header>`: Represents the page's header or a section's heading.
- `<nav>`: Defines navigation links.
- `<article>`: Represents independent content.
- `<footer>`: Contains footer information.
- `<section>`: Groups related content together.

Lab Assignment:

Task: Create a simple HTML webpage that includes:

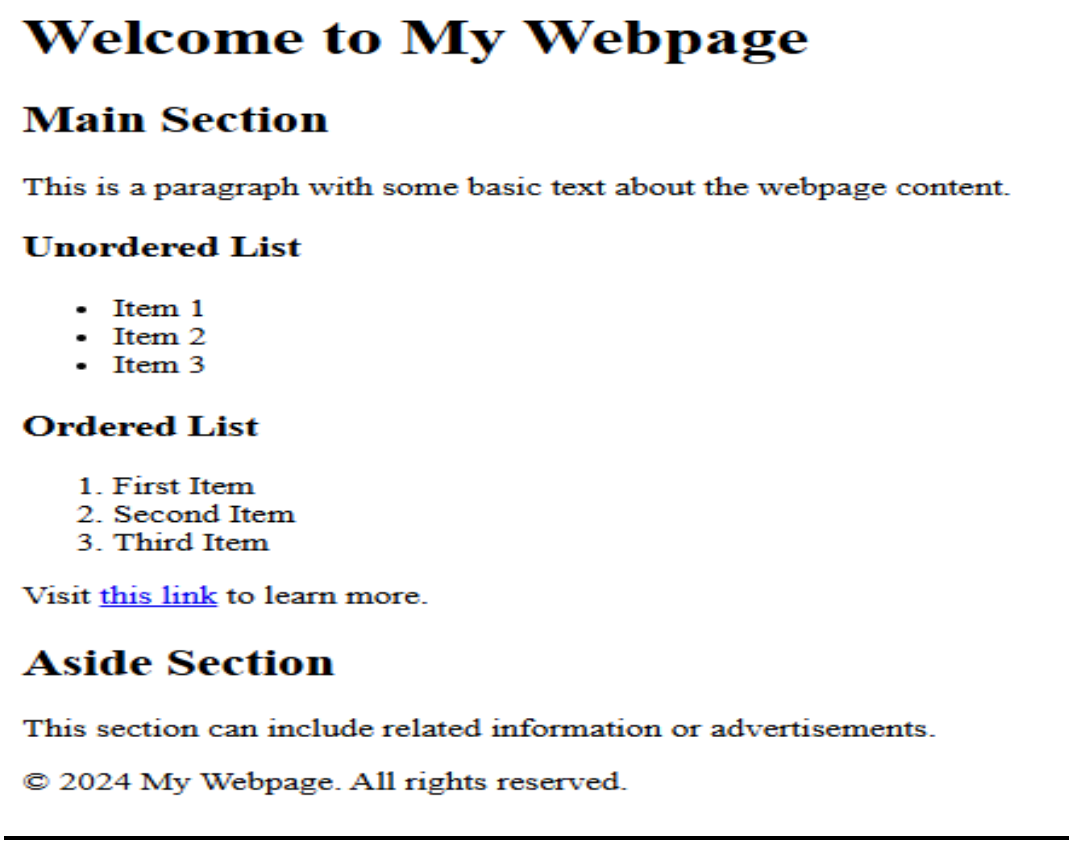
• A header (), footer (), main section (), and aside section ().

• A paragraph with some basic text.

• A list (both ordered and unordered).

• A link that opens in a new tab.

```html
<!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>
    <header>
        <h1>Welcome to My Webpage</h1>
    </header>

    <main>
        <h2>Main Section</h2>
        <p>This is a paragraph with some basic text about the webpage content.</p>

        <h3>Unordered List</h3>
        <ul>
            <li>Item 1</li>
            <li>Item 2</li>
            <li>Item 3</li>
        </ul>

        <h3>Ordered List</h3>
        <ol>
            <li>First Item</li>
            <li>Second Item</li>
            <li>Third Item</li>
        </ol>

        <p>
            Visit <a href="https://www.example.com" target="_blank">this link</a> to learn more.
        </p>
    </main>

    <aside>
        <h2>Aside Section</h2>
        <p>This section can include related information or advertisements.</p>
    </aside>

    <footer>
        <p>&copy; 2024 My Webpage. All rights reserved.</p>
    </footer>
</body>
</html>
```



Welcome to My Webpage

Main Section

This is a paragraph with some basic text about the webpage content.

Unordered List

- Item 1
- Item 2
- Item 3

Ordered List

1. First Item
2. Second Item
3. Third Item

Visit this link to learn more.

Aside Section

This section can include related information or advertisements.

© 2024 My Webpage. All rights reserved.

• 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:
HTML forms are used to collect user input and send it to a server for processing. They are essential for creating interactive webpages, such as login pages, search bars, and feedback forms.

Purpose of Elements:

1. **`<input>`**:
   - Used to collect user data like text, email, password, etc.
   - Example: `<input type="text" placeholder="Enter your name">`.
2. **`<textarea>`**:

- Used for multi-line text input, such as comments or feedback, it also provide attributes like rows and cols
- Example: `<textarea placeholder="Enter your message" rows=”5” cols=”5”></textarea>`.

3. **`<select>`**:
   - Creates a dropdown menu to choose from multiple options.
   - Example:

```
<select>
    <option value="1">Option 1</option>
    <option value="2">Option 2</option>
</select>
```

4. **`<button>`**:
   - Triggers form submission or other actions when clicked.
   - Example: `<button type="submit">Submit</button>`.

Question 2: Explain the difference between the GETand POST methods in form submission. When should each be used?

Difference between GET and POST methods in form submission:

1. **GET Method**:
   - Data is sent via the URL as query parameters.
   - Not secure, as data is visible in the URL.
   - Limited data can be sent due to URL length restrictions.
   - Best for retrieving or requesting data (e.g., search queries).
2. **POST Method**:
   - Data is sent in the request body, not visible in the URL.
   - More secure for sensitive information (e.g., passwords).
   - Allows sending large amounts of data.
   - Best for submitting or updating data (e.g., forms, login).

When to Use:

- **GET**: Use for non-sensitive data retrieval, like search forms or filters.
- **POST**: Use for sensitive data or actions that modify server-side data, like login or registration.

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 label for form controls (e.g., `<input>`, `<textarea>`, `<select>`). It links descriptive text to a specific form element, making the form easier to understand and interact with.

How it improves accessibility:

1. **Screen Reader Support**: Screen readers can identify the purpose of the form control based on the associated label, improving usability for visually impaired users.
2. **Clickable Area**: When a `<label>` is properly linked to a form element, clicking the label also activates the form control, enhancing usability.
3. **Clear Context**: Provides a clear and descriptive context for form fields, reducing confusion for all users.

Lab Assignment:

Task: Create a contact form with the following fields:

• Full name (text input)

• Email (email input)

• Phone number (tel input)

• Subject (dropdown menu)

• Message (textarea)

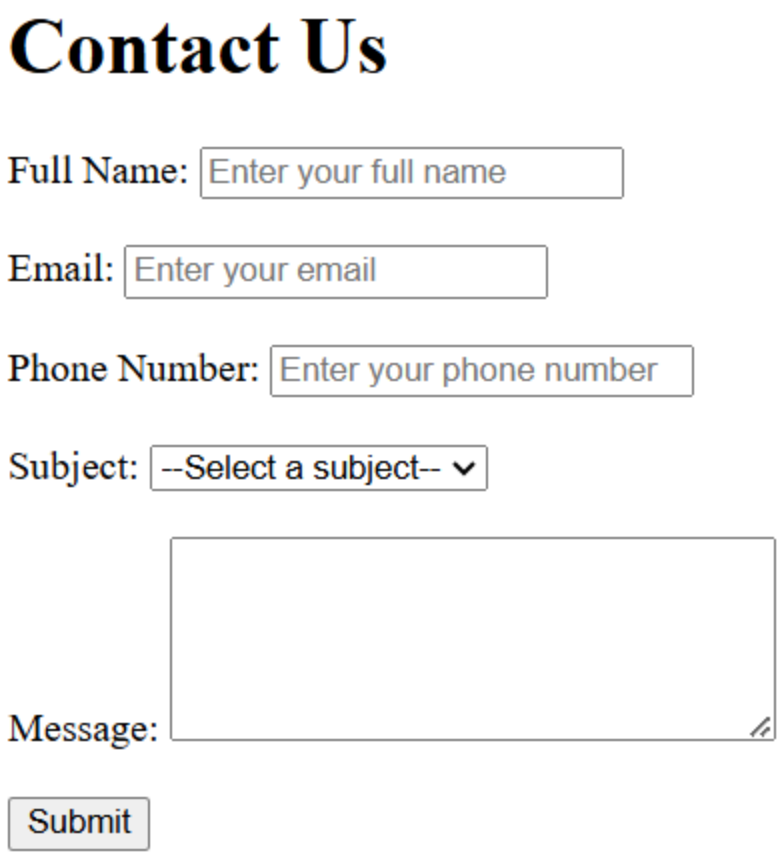
• Submit button

Additional Requirements:

• Use appropriate form validation using required, minlength, maxlength, and pattern.

• Link form labels with their corresponding inputs using the for attribute

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Contact Form</title>
</head>
<body>
    <h1>Contact Us</h1>
    <form action="#" method="post">

        <label for="full-name">Full Name:</label>
        <input type="text" id="full-name" name="full-name" placeholder="Enter your full name" required minlength="3" maxlength="50">
        <br><br>

        <label for="email">Email:</label>
        <input type="email" id="email" name="email" placeholder="Enter your email" required>
        <br><br>

        <label for="phone">Phone Number:</label>
        <input type="tel" id="phone" name="phone" placeholder="Enter your phone number" pattern="[0-9]{10}" title="Enter a valid 10-digit phone number" required>
        <br><br>

        <label for="subject">Subject:</label>
        <select id="subject" name="subject" required>
            <option value="" disabled selected>--Select a subject--</option>
            <option value="general">General Inquiry</option>
            <option value="support">Support</option>
            <option value="feedback">Feedback</option>
        </select>
        <br><br>

        <label for="message">Message:</label>
        <textarea id="message" name="message" placeholder="Enter your message" rows="5" cols="30" required minlength="10" maxlength="500">
        </textarea>
        <br><br>

        <button type="submit">Submit</button>
    </form>
</body>
</html>
```



• 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>`,`<thead>`.**

Structure of an HTML Table

An HTML table is used to organize data into rows and columns. It consists of several elements that define its structure.

Purpose of Elements:

1. **`<table>`**:

- Defines the entire table structure.
- Acts as a container for all rows, columns, and headers.

2. **<tr>** (Table Row):
   - Represents a single row in the table.
   - Contains cells (<th> or <td>) arranged horizontally.
3. **<th>** (Table Header):
   - Defines a header cell in a table.
   - Typically used in the first row or column to describe data.
   - Text inside <th> is bold and centered by default.
4. **<td>** (Table Data):
   - Represents a single data cell in a row.
   - Contains the actual data for the table.
5. **<thead>**:
   - Groups the header rows (<tr> with <th>) together.
   - Helps separate the table's header from its body (<tbody>).

Example Table:

```
<table border="1">
    <thead>
        <tr>
            <th>Product</th>
            <th>Price</th>
            <th>Quantity</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>Apple</td>
            <td>$1</td>
            <td>5</td>
        </tr>
        <tr>
            <td>Banana</td>
            <td>$0.5</td>
            <td>10</td>
        </tr>
    </tbody>
</table>
```

Explanation of Example:

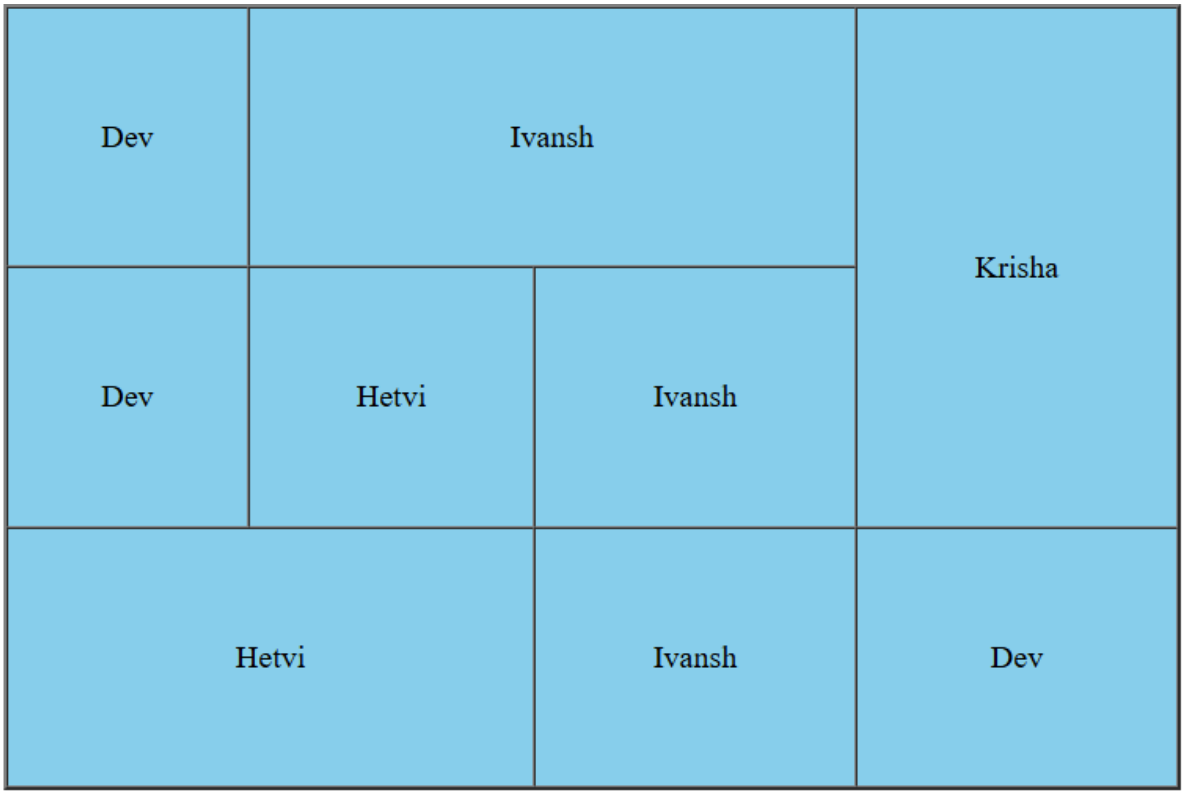
- `<table>`: Defines the table.
- `<thead>`: Groups the header row (`<tr>` with `<th>` elements).
- `<tr>`: Creates rows.
- `<th>`: Creates bold, centered header cells ("Product," "Price," "Quantity").
- `<td>`: Defines the data cells ("Apple," "$1," "5").

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

Difference Between `colspan` and `rowspan`:

1. **`colspan`**:
   - Merges multiple columns into a single cell.
   - Used when a cell needs to span across multiple columns.
2. **`rowspan`**:
   - Merges multiple rows into a single cell.
   - Used when a cell needs to span across multiple rows.

```html
<!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="2" align="center" cellspacing="0" cellpadding="10" bgcolor="skyblue" height="400px" width="600px">
        <tr align="center">
            <td>Dev</td>
            <td colspan="2">Ivansh</td>
            <td rowspan="2">Krisha</td>
        </tr>

        <tr align="center">
            <td>Dev</td>
            <td>Hetvi</td>
            <td>Ivansh</td>
        </tr>

        <tr align="center">
            <td colspan="2">Hetvi</td>
            <td>Ivansh</td>
            <td>Dev</td>
        </tr>
    </table>
</body>
</html>
```



Summary:

- **`colspan`**: Combines multiple columns into one cell.
- **`rowspan`**: Combines multiple rows into one cell.

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

Why Tables Should Be Used Sparingly for Layout Purposes:

1. **Lack of Flexibility**:
   Tables are rigid and not responsive, making it hard to adapt to different screen sizes and devices.
2. **Accessibility Issues**:
   Using tables for layout confuses screen readers, which are designed to interpret tables as data structures. This can hinder accessibility for users with disabilities.
3. **Code Complexity**:
   Tables for layout require excessive HTML, making the code harder to read, maintain, and debug.
4. **Poor SEO**:
   Search engines may misinterpret table layouts, negatively impacting rankings.

Better Alternative: CSS for Layout

CSS (Cascading Style Sheets) is a more efficient and modern way to create layouts. It provides flexibility, responsiveness, and accessibility.

Recommended CSS Techniques:

1. **Flexbox**:
   - Ideal for creating one-dimensional layouts (e.g., rows or columns).
   - Example:

```
<div style="display: flex;">
    <div>Column 1</div>
    <div>Column 2</div>
</div>
```

2. **CSS Grid**:
   - Best for two-dimensional layouts, such as grids or complex designs.
   - Example:

```
<div style="display: grid; grid-template-columns:
1fr 1fr;">
    <div>Item 1</div>
    <div>Item 2</div>
</div>
```

Conclusion:

Instead of tables, use **CSS Flexbox** or **CSS Grid** for creating layouts. They are more responsive, accessible, and maintainable while adhering to modern web development standards.

Lab Assignment:

Task: Create a product catalog table that includes the following columns:

• Product Name

• Product Image (use placeholder image URLs)

• Price

• Description

• Availability (in stock, out ofstock)

Additional Requirements:

• Use thead for the table header.

• Add a border and some basic styling using inline CSS.

• Use colspan or rowspan to merge cells where applicable.

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Product Catalog</title>
</head>
<body>
    <h1>Product Catalog</h1>
    <table border="1" cellspacing="0" cellpadding="10" style="border-collapse: collapse; width: 100%; text-align: left;">

        <thead style="background-color: #f4f4f4;">
            <tr>
                <th>Product Name</th>
                <th>Product Image</th>
                <th>Price</th>
                <th>Description</th>
                <th>Availability</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td rowspan="2">Product A</td>
                <td rowspan="2" align="center">
                    <img src="https://images.unsplash.com/photo-1560343090-f0409e92791a?q=80&w=1964&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D" alt="Product A" style="width: 200px; height:200px;">
                </td>
                <td>$10</td>
                <td>High-quality product with durable materials.</td>
                <td>In Stock</td>
            </tr>
            <tr>
                <td colspan="3" style="text-align: center; background-color: #f9f9f9;">
                    Special Discount: Buy 2, Get 1 Free!
                </td>
            </tr>

            <tr>
                <td>Product B</td>
                <td align="center">
                    <img src="https://plus.unsplash.com/premium_photo-1681711647035-58a0cbb87296?q=80&w=1914&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D" alt="Product B" style="width: 200px; height: auto;">
                </td>
                <td>$15</td>
                <td>Eco-friendly and sustainable product.</td>
                <td>Out of Stock</td>
            </tr>

            <tr>
                <td rowspan="2">Product C</td>
                <td rowspan="2" align="center">
                    <img src="https://plus.unsplash.com/premium_photo-1681711647066-ef84575c0d05?q=80&w=1900&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D" alt="Product C" style="width: 200px; height: auto;">
                </td>
                <td>$20</td>
                <td>Stylish and trendy product for modern users.</td>
                <td>In Stock</td>
            </tr>
            <tr>
                <td colspan="3" style="text-align: center; background-color: #f9f9f9;">
                    Limited Offer: Free shipping on orders above $50!
                </td>
            </tr>
        </tbody>
    </table>
</body>
</html>
```

Product Catalog

Product Name	Product Image	Price	Description	Availability
Product A		$10	High-quality product with durable materials.	In Stock
		Special Discount: Buy 2, Get 1 Free!		
Product B		$15	Eco-friendly and sustainable product.	Out of Stock
Product C		$20	Stylish and trendy product for modern users.	In Stock
		Limited Offer: Free shipping on orders above $50!		