Module 2 – Mernstack – HTML:

➢ HTML Basics - Theory Assignment

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

Answer:

1. HTML stands for **HyperText Markup Language.**
2. It is the standard language used to create web pages.
3. HTML provides the basic structure of a webpage — like headings, paragraphs, images, links, buttons, forms, etc.
4. Browsers read HTML code and display the webpage accordingly.
5. Without HTML, a webpage cannot be created, as it acts as the foundation of every website.

Purpose in web development:

- It defines the layout and structure of content.
- It helps organize information in a readable format.
- It works together with CSS (for styling) and JavaScript (for interactivity).

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

Answer:

1. An HTML document follows a standard structure that every browser understands.
2. **<!DOCTYPE html>**
   - Declares that the document is using HTML5.
   - Helps browsers display the page correctly.
3. **<html> tag**
   - The root (main container) of the entire webpage.
   - Everything is written inside this tag.
4. **<head> section**
   - Contains information *about* the webpage, not the content visible to users.
   - Includes metadata, links to CSS files, scripts, SEO tags, etc.
5. **<title> tag**
   - Specifies the title shown on the browser tab.
   - Helps identify the page.
6. **Important tags inside <head>:**
   - <meta> → for character set, description, keywords
   - <link> → to attach CSS files

- <script> → to attach JavaScript files

7. **<body> section**
   - Contains everything that appears *on the screen*.
   - Includes text, images, links, buttons, forms, tables, videos, etc.
8. Browsers read this structure from top to bottom to display the page correctly.
9. Without this basic structure, the webpage will not load properly.

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

Answer:

1. Block-level Elements

- Always start on a **new line**.
- Take up the **full width** available (100% of the row).
- Used to create larger structures/sections on a webpage.
- Can contain other block or inline elements.

Examples:
, <p>, <h1> to <h6>, <section>, <header>, <footer>, <ul>,

2. Inline Elements

- Do **not** start on a new line.
- Take up **only the space required** by the content.
- Used for small pieces of text or formatting inside block elements.
- Cannot contain block-level elements (mostly text or other inline elements).

Examples:
, <a>, <img>, , , <label>,

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

Answer:

1. Role of Semantic HTML

- Semantic HTML means using meaningful tags that describe the purpose of the content.
- Instead of using only everywhere, we use proper tags like <header>, <nav>, <section>, etc.
- It makes the structure of the webpage clear and organized.

2. Importance for Accessibility

- Screen readers (used by visually impaired users) understand semantic tags better.
- They can easily identify sections like navigation, main content, footer, articles, etc.
- This helps improve the user experience for disabled users**.**

3. Importance for SEO (Search Engine Optimization)

- Search engines (Google) understand semantic tags more clearly.
- It helps Google identify headings, articles, navigation, and important content.
- This improves **search ranking**, crawling, and indexing of the webpage.

4. Examples of Semantic HTML Elements

- **Structural elements:** <header>, <footer>, <main>, <section>, <article>, <aside>
- **Navigation elements:** <nav>
- **Text meaning elements:** , , <mark>, <time>
- **Media-related:** <figure>, <figcaption>

➢ HTML Forms - Theory Assignment:

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

Answer:

1. What are HTML forms used for?

- HTML forms are used to **collect user information** and **send it to the server**.
- Examples: login forms, signup forms, search boxes, feedback forms, payment forms, etc.

2. Purpose of Form Elements

a) <input>

- Used for **single-line input**.
- Can take many types of data like text, email, password, number, date, checkbox, radio buttons, etc.
- Example: name box, email box, password field.

b) <textarea>

- Used for **multi-line text input**.
- Good for messages, comments, feedback, descriptions, etc.

- User can type long text here.

c) <select>

- Creates a **dropdown menu**.
- User can choose one option (or multiple if allowed) from a list.
- Example: selecting country, city, gender, category.

d) <button>

- Creates a clickable button.
- Used to **submit the form**, reset the form, or perform any action.
- Example: “Submit”, “Login”, “Search” buttons.

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

Answer:

1. GET Method

- Sends form data through the **URL** (visible in the address bar).
- Data is **not secure** because anyone can see it.
- Has a **data size limit** (URL length limit).
- Mostly used for **fetching data**, not sending sensitive info.
- GET requests **can be bookmarked** and shared.

Use GET When:

- The form is simple and not sensitive
- Search forms (e.g., Google search)
- Filters, sorting, or queries on a webpage

2. POST Method

- Sends form data **inside the request body**, not in the URL.
- Data is **more secure** (not visible in browser address bar).
- Can send **large amounts of data**.
- Cannot be bookmarked.

Use POST When:

- You are sending **sensitive data** like passwords, login info
- Form submissions like signup, login, payment

- Uploading files
- Sending long text (feedback forms, comments)

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

Answer:

1. Purpose of the <label> element

- The <label> element is used to **describe or identify** a form input.
- It tells the user what the input field is for, like Name, Email, Password, etc.
- When a label is linked to an input (using for and id), **clicking the label focuses the input field**.

Example:

```
<label for="name">Name:</label>
<input id="name" type="text">
```

2. How <label> improves accessibility

- Screen readers used by visually impaired users read the label aloud before the input. Example: It says → "Name: edit text"
- This helps blind users understand what information they must enter.
- Proper labels make forms easier to navigate using a keyboard.
- Without labels, screen readers may not know what each input field means.

➢ 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>, and <thead>

Answer:

1. <table>

- It is the **main container** for creating a table.
- All rows and columns are written inside this tag.

2. <tr> (Table Row)

- Used to create a **row** in the table.
- Each row contains table header cells (<th>) or table data cells (<td>).

3. <th> (Table Header)

- Creates a **header cell** (top row headings).
- The text inside <th> is usually **bold** and **centered** by default.
- Used for column titles like “Name”, “Age”, “Email”.

4. <td> (Table Data)

- Creates a **normal data cell** inside a row.
- It holds the actual data like “Kartik”, “21”, “Example@gmail.com”.

5. <thead>

- Used to group the **header section** of the table.
- Usually contains the <tr> with <th> elements.
- Helps in styling and improves structure/semantics.

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

Answer:

1. Colspan

- Colspan is used to **combine multiple columns** into one cell.
- The cell expands **horizontally (left to right)**.
- Meaning: One cell covers **two or more columns** in the same row.

2. Rowspan

- Rowspan is used to **combine multiple rows** into one cell.
- The cell expands **vertically (top to bottom)**.
- Meaning: One cell covers **two or more rows** in the same column.

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

Answer:

1. Why table should not be used for layout

- **Tables are meant for showing data**, not designing page structure.
- Using tables for layout makes the HTML **complex**, **hard to maintain**, and **difficult to style**.
- Tables reduce **flexibility** — they don't adjust well on mobile screens or responsive designs.
- Screen readers read tables as data, which **confuses visually impaired users**, making accessibility poor.
- Table-based layouts load slower and can cause layout issues.

2. Better alternative

- The modern and recommended way is to use **CSS layout techniques** such as:
    - **CSS Flexbox**
    - **CSS Grid**

Why these are better:

- Easier to design responsive websites.
- Cleaner, simpler HTML.
- Better for accessibility and SEO.
- Easier to control alignment, spacing, and layout.