Assignment.1

1. HTML Basics

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

Answer: HTML stands for Hyper Text Markup Language. It is the standard markup language used to create and structure content on the World Wide Web.

Purpose of HTML in web development:

1) It provides the **basic structure** of a webpage.
2) It defines **content and layout** that web browsers can display.
3) It allows integration of text, images, videos, and other media.
4) It works with CSS (for styling) and JavaScript (for interactivity) to build complete web applications.

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

Answer: **The basic structure of an HTML document defines how content is organized and interpreted by a web browser.**

Basic structure:

```
<!DOCTYPE html>
<html>
<head>
  <title>Page Title</title>
</head>
<body>
  <h1>Heading</h1>
```

```
  <p>This is a paragraph</p>
</body>
</html>
```

Mandatory tags and their purposes:

1) **<!DOCTYPE html>** – Declares the HTML version so the browser knows how to render the page.
2) **<html> ... </html>** – The root element that contains all HTML code.
3) **<head> ... </head>** – Contains metadata about the page (e.g., title, character encoding, styles, scripts).
4) **<title> ... </title>** – Specifies the title shown in the browser tab.
5) **<body> ... </body>** – Contains all the visible content of the webpage such as text, images, videos, and links.

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

Answer: In HTML, elements are categorized as **block-level** or **inline** based on how they are displayed in the browser.

1. Block-level element

Definition: Start on a new line and take up the full width available.
Purpose: They create a block or section in the page layout.

- Examples:
  - – Generic container for grouping content.
  - <p> – Paragraph.
  - <h1> to <h6> – Headings.
  - / – Lists.

2. Inline elements

Definition: Do **not** start on a new line; they only take up as much width as needed.
Purpose: They format content **within** a block-level element.

- Examples:
  - – Inline container for styling text.
  - <a> – Hyperlinks.
  - – Bold text (semantic).
  - – Italic text (semantic).

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

Role of Semantic HTML

Semantic HTML uses tags that clearly describe their meaning and purpose in a webpage's structure. These elements tell both the browser and developers what kind of content they contain, rather than just how they look.

Importance for Accessibility

- **Screen readers** can interpret semantic elements more effectively, enabling visually impaired users to navigate content.
- Improves **keyboard navigation** and logical reading order.
- Helps assistive technologies understand headings, lists, tables, and sections.

Importance for SEO

- Search engines use semantic HTML to better understand the **content hierarchy** and **context**.
- Well-structured semantic HTML improves indexing and seacrch ranking**s.**
- Elements like <article> and <header> can help search engines identify key content.

Examples of Semantic Elements

- **Structural**: <header>, <footer>, <main>
- **Content Sections**: <section>, <article>, <aside>
- **Text Meaning**: (important), (emphasis), <abbr> (abbreviation)
- **Media Descriptions**: <figure>, < figacaption>
-

Structal Elements exmples

1. <header>

i) Represents the introductory content of a page or a section, typically containing navigational links, logos, or headings.

Example:

```
<header>
  <h1>My Website</h1>
  <nav>
    <a href="#">Home</a>
```

```
  <a href="#">About</a> |
  <a href="#">Contact</a>
 </nav>
</header>
```

2.<footer>

I. Defines navigation links or menu. This helps both users and search engines understand the navigational structure of a site.

Example:

```
<footer>
 <p>&copy; 2025 My Website | All Rights Reserved</p>
</footer>
```

3. <main>

i Represents the dominant content of the document (the main content), excluding headers, footers, and sidebars.

Example:

```
<main>
 <h2>Welcome to My Website</h2>
 <p>This is the main content area.</p>
</main>
```

Content Sections example:

1: <section>

i. Represents a distinct section of content within a page, often with a heading. This can be used for grouping related content.

Example:

```
section>
 <h2>Our Services</h2>
 <p>We provide web design, development</p>
</section>
```

2. <aside>

i) Represents content that is tangentially related to the content around it, like sidebars or pull quotes

Example:

```
<aside>
 <h3>Related Links</h3>
 <ul>
  <li><a href="#">HTML Tutorial</a></li>
  <li><a href="#">CSS Basics</a></li>
 </ul>
</aside>
```

3. <article>

I. Represents independent content that could be distributed or syndicated (like blog posts, news articles, etc.).

Example:

```
<article>
 <h2>Blog Post Title</h2>
 <p>This is a blog post that can stand alone.</p>
</article>
```

2. HTML Forms

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

Answer: HTML forms are used to collect user and send it to a web server for processing. They provide interactive fields where users can enter information such as text, email, passwords, or select options. Forms are commonly used for **login pages, registrations, search boxes, feedback, surveys, and contact forms**

Purpose of key elements in forms:

1. <Input> elements
   - Used for different types of single-line user input (text, email, password, number, date, etc.

Example: <input type="text" placeholder="Enter your name">

2 < textarea>elements

- Used for **multi-line text input**, such as comments, messages, or feedback.

Example: <textarea name="message" rows="10" cols="30">

3 <select>elements

- Creates a **dropdown menu** for users to select one or more options

Example: <label for="cars">Choose a car:</label>
<select id="cars" name="cars">
<option value="volvo">Volvo</option>
<option value="saab">Saab</option>
<option value="fiat">Fiat</option>
<option value="audi">Audi</option>
</select>.

4 <button>elements - Used to **submit the form**.

Example:<input type="submit" value="submit">

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

Answer:

Difference between GET and POST methods in form submission:

When an HTML form is submitted, the data can be sent to the server using different **HTTP methods**, the most common being **GET** and **POST**.

1. **GET Method:**
   - This method appends the form-data to the URL in name/value pairs.
   - Form data is appended to the URL as query parameter.
   - Data is visible in the browser's address bar.

Example:

<form action="https://www.gujaratuniversity.ac.in/" method="get"
target="_blank">

When to use GET:

- Retrieving data or performing searches.
- When no sensitive information is involved.
- When you want the URL to be shareable/bookmarkable

2. **Post Method:**
   - This method sends the form-data as an HTTP post transaction.
   - Form data is sent in the HTTP request body.
   - Not visible in the URL.

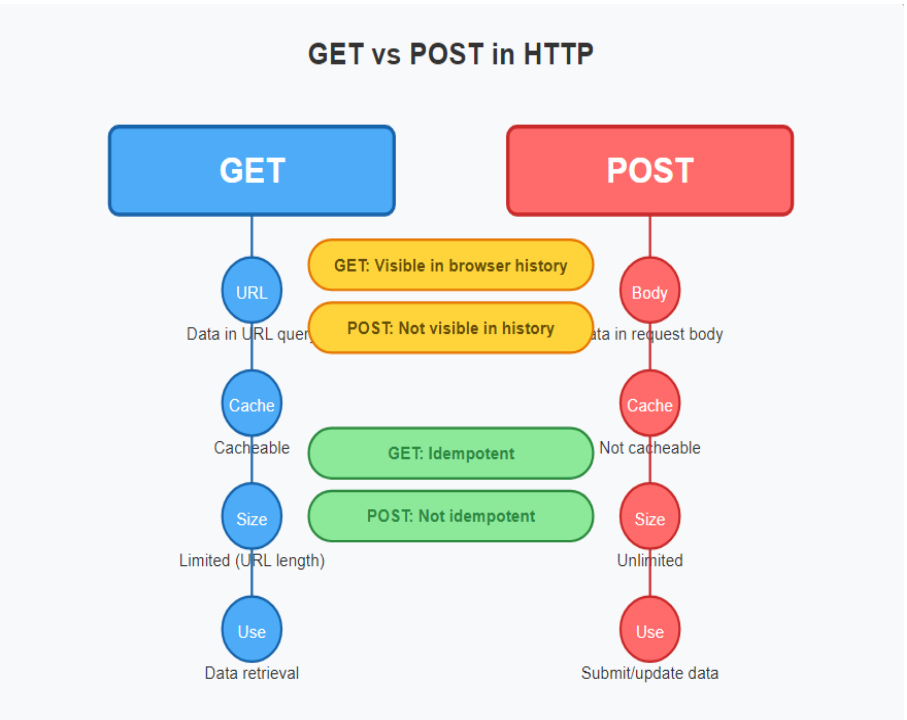
Example:

<form action="https://www.gujaratuniversity.ac.in/" method="post">

When to use POST:

- Submitting sensitive or large amounts of data
- Performing actions that change server state (e.g., creating an account, submitting a comment).

- When privacy is a concern (e.g., login forms)



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

Answer:

Purpose of the <label> element in a form:

The <label> element is used to **define a caption for form controls** such as <input>, <textarea>, <select>, etc. It helps users understand what information is expected in each field.

Purpose of the Element

- Associates a text description with a specific form input.
- Helps users understand what each input field is for.
- Improves form usability and user experience.

How it improves accessibility:

- Screen readers use labels to describe form fields, ensuring users know what information to enter.
- Increases the clickable area for small controls (like checkboxes and radio buttons), which is helpful for users with motor impairments.
- Ensures that forms are more intuitive and easier to navigate using keyboard or assistive devices.

Example:

```
<form>
 <label for="username">Username:</label>
 <input type="text" id="username" name="username">
```

</form>

3 HTML Tables

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:

Structure of an HTML Table:
An HTML table organizes data into **rows** and **columns**. It is mainly built using a combination of elements such as <table>, <tr>, <th>, <td>, and <thead>.

Structure of an HTML Table:

```
<table>
    <tr>
      <th>Roll nub</th>
      <th>Name</th>
      <th>Subjet</th>
    </tr>
    <tr>
      <td rowspan="3">1 2 3</td>
      <td>kirtan</td>
      <td>it</td>
    </tr>
</table
```

Elements and Their Purpose:

1. <table>

- The main container that defines the start and end of a table.
- All table-related elements must be placed inside it.

Example:

```
<table>
  <!-- Table rows and cells go here -->
</table>
```

2. <tr>
   - Defines a single **row** in the table.
   - Inside each `<tr>`, you place either `<th>` (header cell) or `<td>` (data cell).

Example:

```
<tr>
  <th>Roll nub</th>
  <th>Name</th>
  <th>Subjet</th>
</tr>
```

3. *<th>*

- *Represents a **header cell** (usually bold and centered by default).*
- *Used inside a <tr> to define column or row headings.*

Example:

```
<tr>
  <th>Roll nub</th>
  <th>Name</th>
  <th>Subjet</th>
</tr>
```

4. *<td>*

- *Represents a **data cell** in the table.*
- *Used inside a <tr> to hold actual content (text, images, links, etc.).*

Example:

```
<tr>
  <td rowspan="3">1 2 3</td>
  <td>kirtan</td>
  <td>it</td>
</tr>
```

5. <thead>
   - Groups the **header rows** of a table.
   - Helps separate header content from the body (`<tbody>`) and footer (`<tfoot>`).

Example:

```
<thead>
  <tr>
    <th>Name</th>
    <th>Age</th>
  </tr>
</thead>
```

- <table> → Defines the whole table.
- <tr> → Creates a row.
- <th> → Defines a header cell.
- <td> → Defines a data cell.
- <thead> → Groups table headers.

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

Answer:

The colspan and rowspan attributes in HTML tables are used to merge cells across multiple columns or rows, respectively. They are applied to or elements.

Colspan:

- *The colspan attribute merges a cell horizontally across multiple columns. The value assigned to colspan indicates the number of columns the cell should span.*

Example:

```
<table>
    <tr>
      <th>Roll nub</th>
      <th>Name</th>
      <th>Subjet</th>
    </tr>
    <tr>
      <td colspan="3">1 2 3</td>
```

```
        <td>kirtan</td>
        <td>it</td>
    </tr>
```

Rowspan:

The rowspan attribute merges a cell vertically across multiple rows. The value assigned to rowspan indicates the number of rows the cell should span.

Example:

```
 <table>
    <tr>
      <th>Roll nub</th>
      <th>Name</th>
      <th>Subjet</th>
    </tr>
    <tr>
      <td rowspan="3">1 2 3</td>
      <td>kirtan</td>
      <td>it</td>
    </tr>
</table>
```

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

Answer: Tables should be used sparingly for layout purposes because they were never designed for layout—they were designed to display tabular data. Using tables for layout introduces several problems and is considered outdated and inaccessible in modern web design.

1. Poor Accessibility:

i) Screen readers may misinterpret layout tables as data tables, confusing users with disabilities.
ii) Navigation becomes harder for assistive technologies.

2. Not Responsive:

i) Tables don’t adapt well to small screens (e.g., phones and tablets).
ii) They make mobile design complex and inflexible.

3. Hard to Maintain:
   i) Table-based layouts involve deeply nested code that is difficult to read and update.
   ii) ii) Changes require modifying large sections of HTML

4. Violates Semantic HTML :

i) HTML should separate content from presentation.
ii) Tables add presentational markup instead of using CSS, which breaks this principle.

5. Slower Page Rendering:

   i) Browsers must fully load and process table structures before rendering the layout.

Better Alternative: CSS Layout Techniques:

- Modern CSS provides powerful tools for creating flexible, accessible, and responsive layouts.

1. CSS Flexbox:

- Great for one-dimensional layouts (horizontal or vertical).
- Easy to align items and distribute space.

2. CSS Grid :

- Ideal for two-dimensional layouts (rows and columns).
- Powerful control over layout placement.

3. Media Queries :

- **Make layouts responsive based on screen size**