Module 2 – Frontend - HTML

HTML Basics

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

- HTML (HyperText Markup Language) is the standard markup language used to structure web pages.
- It is used to create various elements of a webpage/website such as nav-bar, paragraphs, images, video, Forms, and more, which are displayed in a web browser on how to display it. .
- HTML provides the basic framework for web pages, defining their layout and content structure

. **Purpose of HTML in Web Development:**

- **Structure and Organization:**HTML establishes the hierarchical structure of a web page, organizing content into sections, paragraphs, headings, and more.
- **Content Definition:**HTML defines the content of a web page, including text, images, videos, and other multimedia elements.
- **Link Creation:**HTML allows for the creation of hyperlinks, connecting different web pages and enabling navigation.
- **Basic Framework:**HTML forms the backbone of web pages, providing the necessary structure for subsequent styling (CSS) and interactivity (JavaScript).
- **Semantic Meaning:**HTML elements have semantic meaning, helping search engines understand the context and content of a page.

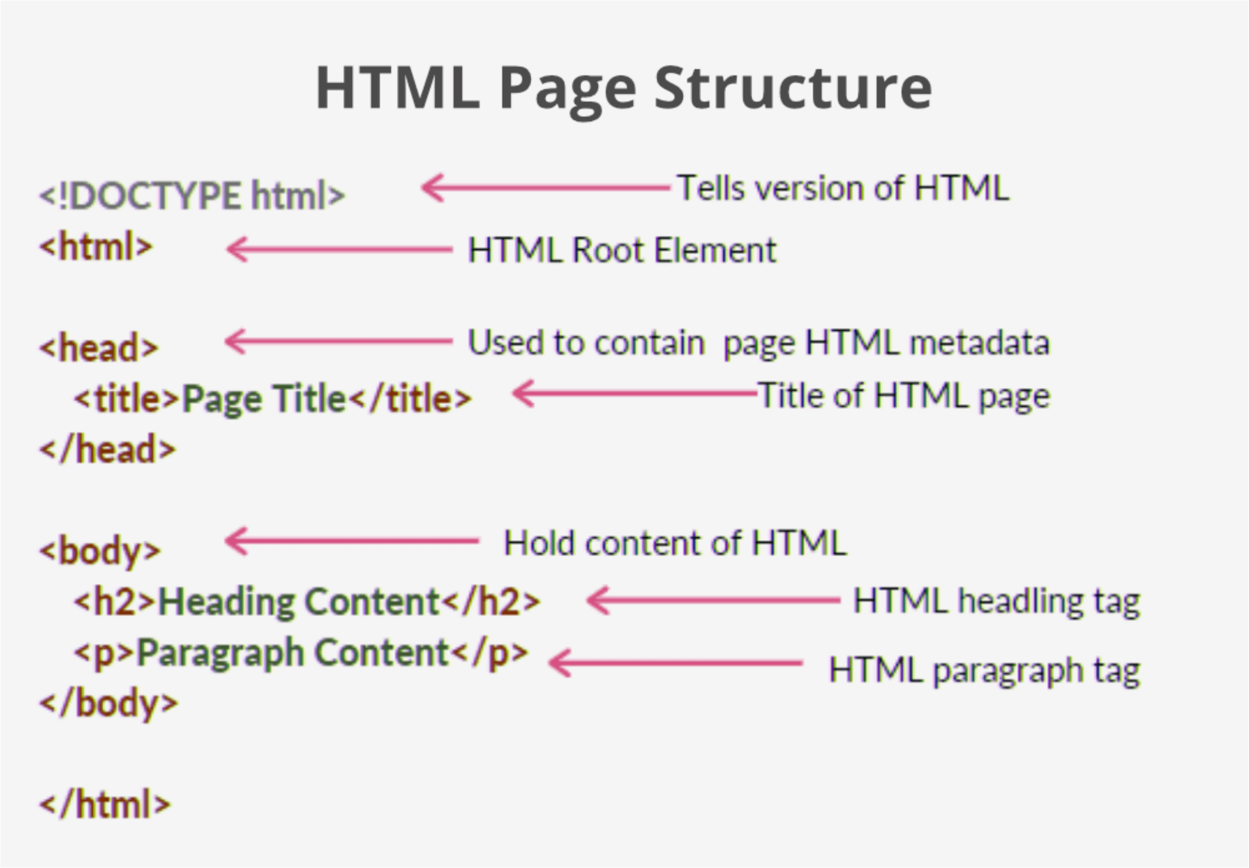
Question 2: Explain the basic structure of an HTML document. Identify the mandatory tags and their purposes.

- The basic structure of an HTML document involves a DOCTYPE declaration, an <html> tag, a <head> section, and a <body> section. The DOCTYPE informs the browser about the HTML version, and the <html> tag is the root element.
- The <head> contains metadata like the title and links to external resources, while the <body>holds the visible content of the webpage.

Detailed Explanation:

1. **DOCTYPE Declaration:**
    - This is not an HTML element but an instruction to the browser to interpret the document as HTML.
    - It typically looks like: <!DOCTYPE html>.
2. **<html> Tag:**
    - The <html> tag is the root element of the HTML document.
    - It encloses all other HTML elements.
    - It can optionally include a lang attribute to specify the language of the page.
3. **<head> Section:**
    - The <head> section contains metadata about the HTML document, which is not displayed directly on the page.
    - Common elements within the <head> include:
        - <title>: Defines the title of the page, which is displayed in the browser tab.
        - <meta>: Provides information like character set, viewport settings, and other meta data.
        - <link>: Links to external stylesheets or other resources.
        - <style>: Embeds CSS styles.
        - <script>: Embeds JavaScript code.
4. **<body> Section:**

- The <body> section contains the content that is displayed on the webpage.
- It includes all the visible elements of the page, such as text, images, links, and other HTML elements.



HTML Page Structure

```
<!DOCTYPE html>
<html>

<head>
  <title>Page Title</title>
</head>

<body>
  <h2>Heading Content</h2>
  <p>Paragraph Content</p>
</body>

</html>
```

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

Block-level elements take up the full available width on a page, starting a new line, while inline elements only take up the space needed for their content and stay within a line of text.

Block-level elements:

- Start on a new line.
- Take up the full width available (left to right).
- Are commonly used for structuring content like paragraphs, headings, lists, and sections.
- Examples: <p>, <h1>, <div>, <ul>, <ol>, <nav>, <footer>, <header>, <section>.

Inline elements:

- Do not start on a new line.
- Only take up as much space as necessary for their content.
- Are commonly used for formatting text within a line, like links, spans, or bold/italic text.
- Examples: <span>, <a>, <strong>, <em>, <br>, <input>, <label>.

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

- Semantic HTML plays a crucial role in web development, enhancing both accessibility and SEO.
- It uses specific HTML tags (like <header>, <nav>, <main>, <article>) to define the meaning and structure of content, making it easier for screen readers, search engines, and developers to understand and interpret the page.
- This improves user experience, especially for those with disabilities, and signals to search engines the context of the content, leading to better indexing and ranking.

Accessibility:

Screen Reader Compatibility: Semantic HTML elements are interpreted by screen readers, allowing users with visual impairments to navigate and understand the content more effectively.

Improved User Experience: By clearly defining sections and relationships within the page, semantic HTML makes it easier for users to locate specific information, navigate, and understand the overall structure.

Compliance with Accessibility Guidelines: Using semantic HTML helps websites meet the standards set by the Web Content Accessibility Guidelines (WCAG), ensuring a more inclusive and accessible online experience.

SEO:

Enhanced Search Engine Understanding: Search engines like Google can more accurately interpret the context and structure of content when semantic HTML is used.

Improved Indexing and Ranking: By providing clear signals about the content's meaning, semantic HTML helps search engines prioritize relevant keywords and improve a website's ranking for those keywords.

Facilitates Rich Results: Semantic HTML can work in conjunction with schema markup to enable rich results in search, providing additional information like ratings, prices, or event dates.

HTML Forms

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

- HTML forms are used to collect data from users and submit it to a server for processing.
- They are essential for various web applications, including user registration, login, search queries, and feedback submissions. Key elements within forms include:
- **Input:** Allows users to enter single-line text or other types of data (numbers, dates, etc.).
- **Textarea:** Provides a multi-line text box for longer inputs like comments or reviews.
- **Select:** Creates a dropdown list where users can choose from a predefined set of options.
- **Button:** Triggers actions, such as submitting the form or performing other tasks.

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

- In form submission, GET retrieves data from a server, while POST sends data to the server, often for creating or updating resources.

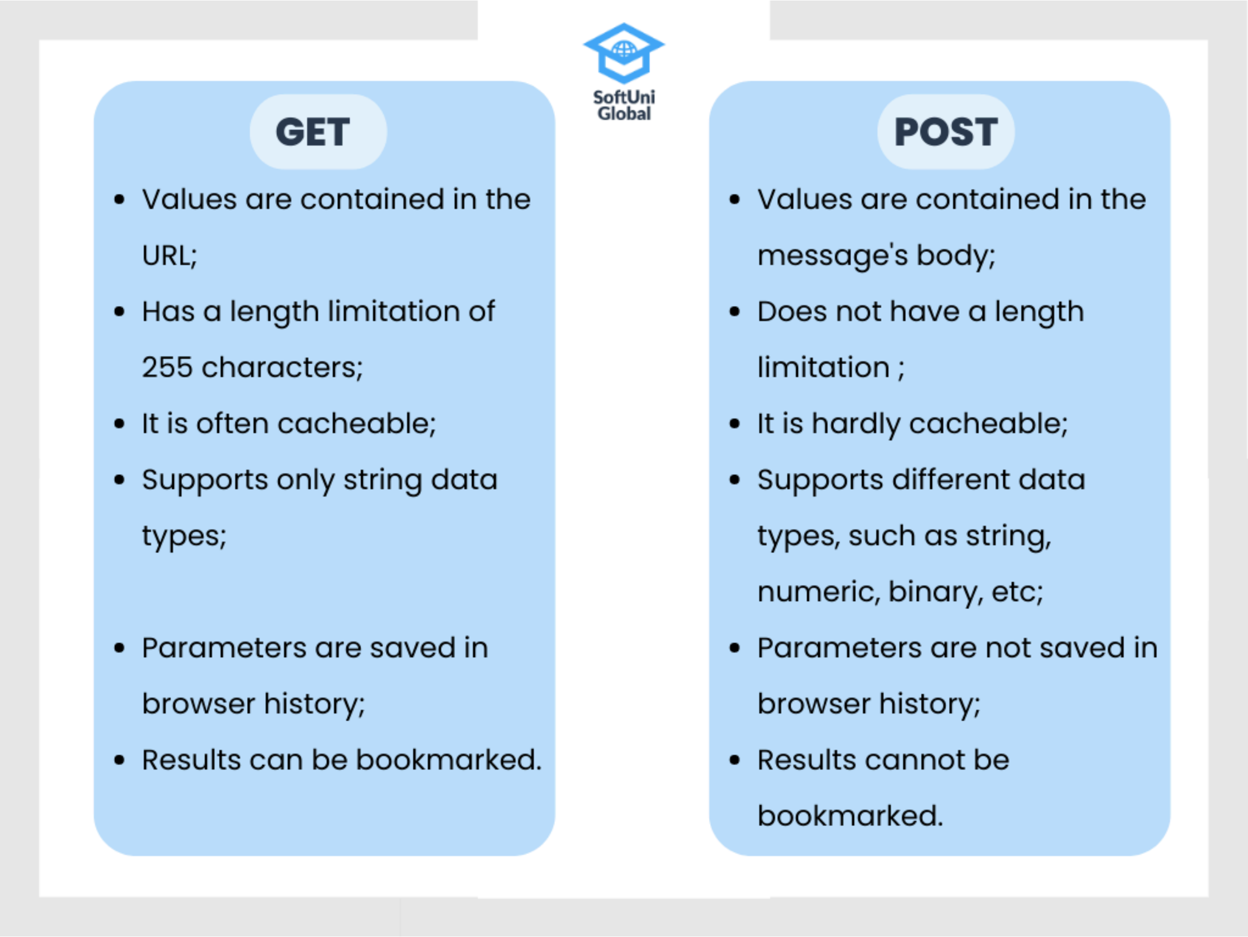
- GET passes data in the URL, while POST sends it in the request body. POST is generally preferred for sensitive data and larger amounts of information.

GET:

- **Use cases**: Retrieving data, searching, filtering, paging, and bookmarking.
- **Data transmission**: Data is appended to the URL.
- **Security**: Less secure due to data exposure in the URL.
- **Caching:** GET requests are often cached.
- **Size limitation**: Data length in the URL is limited.
- Example: A search bar where you enter a query.

POST:

- **Use cases:** Submitting forms, modifying data, creating new resources, and sending sensitive data.
- **Data transmission:** Data is included in the request body.
- **Security:** More secure because data is not visible in the URL.
- **Caching:** POST requests are rarely cached.
- **Size limitation:** No restrictions on data length
- **Example:** Submitting a login form.



GET

- Values are contained in the URL;
- Has a length limitation of 255 characters;
- It is often cacheable;
- Supports only string data types;
- Parameters are saved in browser history;
- Results can be bookmarked.

POST

- Values are contained in the message's body;
- Does not have a length limitation ;
- It is hardly cacheable;
- Supports different data types, such as string, numeric, binary, etc;
- Parameters are not saved in browser history;
- Results cannot be bookmarked.

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

- The label element in a form serves to visually and programmatically associate a label with a form input element, improving both accessibility and usability.
- By providing a clear description of the input field, labels allow screen readers to announce the purpose of each field, making it easier for users with visual impairments to understand and complete the form.

How it Improves Accessibility:

1. Screen Reader Support: Labels enable screen readers to identify the purpose of each form field, allowing users with visual impairments to understand what data is required in each input.

2. Enhanced Usability: Labels provide context for form controls, reducing user errors and making interactions more intuitive.

3. Larger Clickable Area: When a user clicks on the label, the browser automatically focuses on the associated input field, effectively increasing the clickable area for activation, especially helpful for those with motor impairments or using touch-screen devices.

4. Clarity and Understanding: Labels help users quickly understand what information is expected in each field, minimizing confusion and errors.

5. Semantic Structure: Labels help create a more semantic and understandable structure for the form, making it easier for developers and users alike to understand the purpose of each field.

HTML Tables

Question 1: Explain the structure of an HTML table and the purpose of each of the followingelements: <table>, <tr>, <th>, <td>, and <thead>.

- An HTML table organizes data into rows and columns using various elements.
- The <table> tag defines the entire table structure, while <tr>defines each row, <th> defines header cells (typically bold and centered), <td> defines data cells, and <thead> groups header content within a table.

Here's a more detailed breakdown:

<table>: The parent tag that encloses all other table elements, defining the table's boundaries.
<tr>: Represents a single row within the table. Each <tr>typically contains one or more <th> or <td>elements to define the cells in that row.
<th>: Defines a header cell. Header cells are generally displayed in a bold and centered font, indicating they contain column or row headings.
<td>:Defines a data cell. These cells hold the actual data or content within the table.
<thead>: Used to group header rows (containing <th>elements) into a separate section within the table. This helps in cases where the table may have a fixed header that remains visible even when scrolling the body of the table.

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

- In HTML table creation, colspan and rowspan are attributes used to merge cells, but in different directions. Colspan merges cells horizontally, spanning across columns, while rowspan merges cells vertically, spanning across rows.

  Here's a more detailed explanation:

- **colspan (column span):**

  This attribute is used to make a cell cover multiple columns in a table. When a cell has a colspan value (e.g., colspan="2"), it will span two columns, effectively merging the cell with the adjacent columns.
- **rowspan (row span):**

  This attribute allows a cell to span multiple rows in a table. When a cell has a rowspan value (e.g., rowspan="2"), it will span two rows, merging the cell vertically with the adjacent rows.

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

- In this article, we will learn why we should avoid using tables for layout in HTML.
- A website can be divided into various sections comprising of header, menus, content, and footer based on which there are many different layout designs available for developers.
- Different layouts can be created by using an HTML div tag and CSS property to style them.
- Tables are used to store or view data in a tabular form.

- When you look at this webpage then you will come to know that in this current webpage there is a number of different tabs or cards which are showing different information.
- When you look at the table then you can observe that there are cells of the same size.
- A webpage with the same size layout will not look good as compared to a grid layout.

The reasons to avoid using tables for layout in HTML are listed below:

- **Tables Are Not Accessible:** Most search engines read the webpage as they read HTML and it becomes difficult for the search engine to render the table layout. This is the main reason why we follow the HTML5 format.
- **Tables Are Tricky:** When you perform nesting in tables then it is difficult to maintain it. When you want to change something after some days then it will become complicated for the developer to debug the code.
- **Tables Are Inflexible:** When you want to create the table layout with specified widths then it will become a rigid layout or not flexible and then it will take some extra time to load your page properly. The flexible layout always looks good on any device.
- **Tables Hurt Search Engine Optimization:** Many developers create the navigation on the left-hand side and the rest of the content on the right side. If you use tables the search engine will load the content first then the navigation will start to load without navigation, the content will look not so good.
- **Tables Don't Always Print Well:** When you try to print the table layout the printer will change the interface because the table layout is too wide. The printer will then try to cut down some content or show extra content to the next page which will make it complicated.
- **Tables for Layout Are Invalid in HTML 4.01:** You can't create the table layout when you use HTML 4.01 because you can only be allowed to create a simple table. For example

spreadsheets or databases. Another reason is that other browsers find it tough to render through the table layout.