1. HTML Basics

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

➔ HTML stands for Hypertext Markup Language. HTML is used to create web pages and web applications. HTML is widely used language on the web.

Purpose :

1. **Structure Content:** HTML organizes and structures the content of web pages using tags and elements.
2. **Display Text and Media:** It allows the display of text, images, videos, and other media on websites.
3. **Hyperlinking:** HTML enables the creation of links to other web pages, allowing for navigation between different parts of a website or external resources.
4. **Embed Other Technologies:** HTML can embed other technologies like CSS (for styling) and JavaScript (for interactivity) into web pages.
5. **Accessibility and SEO:** Proper HTML structure enhances website accessibility and improves search engine optimization (SEO) by making content readable to screen readers and search engines.

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

➔Basic Structure of 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>

</body>
</html>
```

Mandatory tags and their purposes :

The <!DOCTYPE html> declaration defines that this document is an
HTML5 documentThe <html> element is the root element of an HTML
page
· The <head> element contains meta information about the HTML page
· The <title> element specifies a title for the HTML page (which is shown in the
browser's title bar or in the page's tab)
· The <body> element defines the document's body, and is a container for all the
visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists,etc.
· The <h1> element defines a large heading
· The <p> element defines a paragraph

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

→

Block-Level Elements :

Definition:

- Block-level elements start on a **new line** and take up the **full width** available by default (stretching out to the left and right as far as they can).
- They are used to structure the **main content** of a page.

Example :

```
<div>This is a div</div>

<p>This is a paragraph</p>

<h1>Heading 1</h1>

<ul>

  <li>List item</li>

</ul>

<section>Section of content</section>
```

Inline Elements :

Definition:

- Inline elements do **not** start on a new line. They only take up as much width as necessary.
- They're generally used for styling parts of text or embedding small pieces within block-level elements.

Example :

<span>This is a span</span>

<a href="#">This is a link</a>

<strong>Bold text</strong>

<em>Emphasized text</em>

<img src="image.jpg" alt="An image">

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

→

Semantic HTML refers to the use of HTML elements that **clearly describe their meaning and role** in the content structure — both to the browser and to developers (and assistive technologies like screen readers).

Why Semantic HTML Matters :

1. Accessibility

- Semantic elements provide **context and structure** to assistive technologies (e.g., screen readers).
- Helps users with disabilities understand the **purpose** of each section of the page (like navigation, main content, articles, etc.).

Example: A screen reader can announce "Navigation region" when it encounters a <nav> element, making the browsing experience smoother and more understandable.

---

2. SEO (Search Engine Optimization)

- Search engines use semantic elements to better **understand the content** of a webpage.
- Improves how content is **indexed and ranked** in search results.

Example: Google gives more relevance to content in <article> or <main> tags than in a generic <div>.

Examples of Semantic Elements

Element	**Purpose**
<header>	Defines a header section of a page or section
<nav>	Contains navigation links
<main>	Represents the main content
<section>	Groups related content into sections
<article>	Self-contained, reusable content (e.g., blog post)
<aside>	Secondary content, like a sidebar
<footer>	Defines the footer of a section or page
<figure> / <figcaption>	Contains images/media and captions

2. HTML Forms

• Q 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** and send it to a server for processing. Common use cases include:

- Logging in or registering
- Searching content
- Submitting feedback or surveys
- Uploading files
- Making online purchases

The purpose of the input,textarea,select,button element :

Element	Purpose
<input>	Captures single-line input (various types)
<textarea>	Captures multi-line text input
<select>	Dropdown menu for options
<button>	Triggers form submission or other actions

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

→

Difference between the GET and POST :

Feature	GET	POST
Data in URL	Yes (appended to URL as query string)	No (sent in HTTP request body)
Visibility	Visible in browser address bar	Hidden from the URL
Data Length	Limited (URL length restrictions)	Unlimited (within server limits)
Caching	Can be cached by browser	Not cached
Bookmarked	Can be bookmarked with parameters	Cannot be bookmarked with data
Use Case	Safe data retrieval (no side effects)	Data submission (e.g., login, forms)
Security	Less secure for sensitive data	Better for confidential data

Use GET **when:**

- You’re **retrieving data**, not modifying anything.
- The data is **not sensitive** (e.g., search queries).
- You want the form to be **bookmarkable or shareable**.

Use POST **when:**

- You’re **sending sensitive or private data** (e.g., passwords).
- You’re **changing data on the server** (e.g., submitting a form, saving to a database).
- The amount of data is large or includes **files**.

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

→

Purpose of the <label> **Element in a Form** :

1. **Improves form usability** (especially for all users)
2. **Enhances accessibility** for users relying on screen readers or assistive technologies
3. **Increases clickable area** – clicking the label focuses/selects the related input

How It Improves Accessibility :

- **Screen readers** can announce the label when focusing on the form control, giving users context (e.g., "Name: text input").
- Ensures that **visually impaired users** understand what each form field is for.
- Helps **keyboard navigation** users by linking labels to inputs explicitly.

3. HTML Tables

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

→

Basic Structure of an HTML Table :

An HTML table is used to **display tabular data** — data arranged in **rows and columns**.

<table>

```
  <thead>
    <tr>
      <th>Header 1</th>
      <th>Header 2</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Row 1, Col 1</td>
      <td>Row 1, Col 2</td>
    </tr>
    <tr>
      <td>Row 2, Col 1</td>
      <td>Row 2, Col 2</td>
    </tr>
  </tbody>
</table>
```

the purpose of each of the following elements: < table> ,<tr>,<th>,<td>, and <thead> .

Element	Purpose
<table>	Container for the entire table
<tr>	Defines a table row
<th>	Table header cell (bold + semantic)
<td>	Table data cell
<thead>	Groups header rows (usually <th>)

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

→ Difference Between colspan and rowspan :

Attribute	Stands For	Spans Across	Use Case
colspan	**Column Span**	**Multiple columns (horizontally)**	Merge cells across columns

Attribute	Stands For	Spans Across	Use Case
rowspan	**Row Span**	**Multiple rows (vertically)**	Merge cells across rows

1. colspan Example (Span Multiple Columns)

```
<table border="1">
  <tr>
    <th colspan="2">Name & Age</th>
  </tr>
  <tr>
    <td>John</td>
    <td>25</td>
  </tr>
</table>
```

2. rowspan Example (Span Multiple Rows)

```
<table border="1">
  <tr>
    <th rowspan="2">Name</th>
    <td>John</td>
  </tr>
  <tr>
    <td>Jane</td>
  </tr>
</table>
```

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

→

tables be used sparingly for layout purposes :

1.Tables Are Meant for Tabular Data

- HTML tables are **semantically designed** to present **data in rows and columns** (like spreadsheets).
- Misusing them for layout **confuses browsers, screen readers, and search engines**.

2. Poor Accessibility

- Screen readers expect tables to contain data.
- Using tables for layout **makes content harder to interpret** for users with disabilities, harming accessibility.

3. Hard to Maintain

- Table-based layouts are **rigid** and **hard to modify**.
- Any small design change often requires **restructuring the entire table**.

4. Not Responsive

- Tables do not adapt well to different screen sizes (e.g., mobile or tablets).
- They lack the flexibility needed for **modern, responsive design**.

Better Alternative is :

Reason Not to Use Tables for Layout	**Better Alternative**
Not semantic	Use <div>, <section>, etc.
Bad for accessibility	Use CSS with Flexbox/Grid
Difficult to maintain	CSS is modular and scalable
Not mobile-friendly	CSS supports responsiveness