Module 15) HTML in Full Stack 1

HTML Basics Theory Assignment

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

HTML (HyperText Markup Language) is the standard markup language used to create and design documents for the web. It provides the basic structure of web pages using a series of elements or "tags". These tags define different parts of the content such as headings, paragraphs, links, images, and more.

**Purpose in Web Development:**

* HTML forms the **foundation of all websites**.
* It helps structure and organize content so that web browsers can render it properly.
* It allows integration of multimedia (like images, videos, audio).
* It works together 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.

| **Tag** | **Purpose** |
| --- | --- |
| <!DOCTYPE html> | Declares the document type and HTML version |
| <html> | Root element of the HTML document |
| <head> | Contains metadata like title, charset, links to CSS, etc. |
| <title> | Sets the title of the web page (shown on browser tab) |
| <body> | Contains the visible content of the webpage like text, images, etc. |
|  |  |

<!DOCTYPE html>

<html>

<head>

<title>Page Title</title>

</head>

<body>

<h1>Hello, World!</h1>

<p>This is a paragraph.</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** occupy the full width available and start on a new line. They create larger sections or "blocks" in the layout.

**Examples:**

* <div>
* <p>
* <h1> to <h6>
* <ul>, <ol>, <li>
* <section>, <article>

🔹 **Inline elements** do not start on a new line and only take up as much width as necessary. They are used **inside** block elements, mainly for formatting parts of text.

**Examples:**

* <span>
* <a>
* <img>
* <strong>, <em>
* <br>

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

**Semantic HTML** uses meaningful tags that describe the purpose of the content inside them. These tags make the structure of a webpage more understandable to **browsers, developers, screen readers**, and **search engines**.

**Importance:**

* Improves **accessibility** for users with disabilities using assistive technologies.
* Enhances **SEO** (Search Engine Optimization) by helping search engines better understand the content.
* Makes the code more **readable, maintainable**, and **standardized**.

**Examples of Semantic Tags:**

* <header> – Defines the header section of a page or section.
* <nav> – Contains navigation links.
* <main> – Represents the main content area.
* <article> – Represents an independent piece of content.
* <section> – Groups related content together.
* <footer> – Defines the footer area.

Lab Assignment

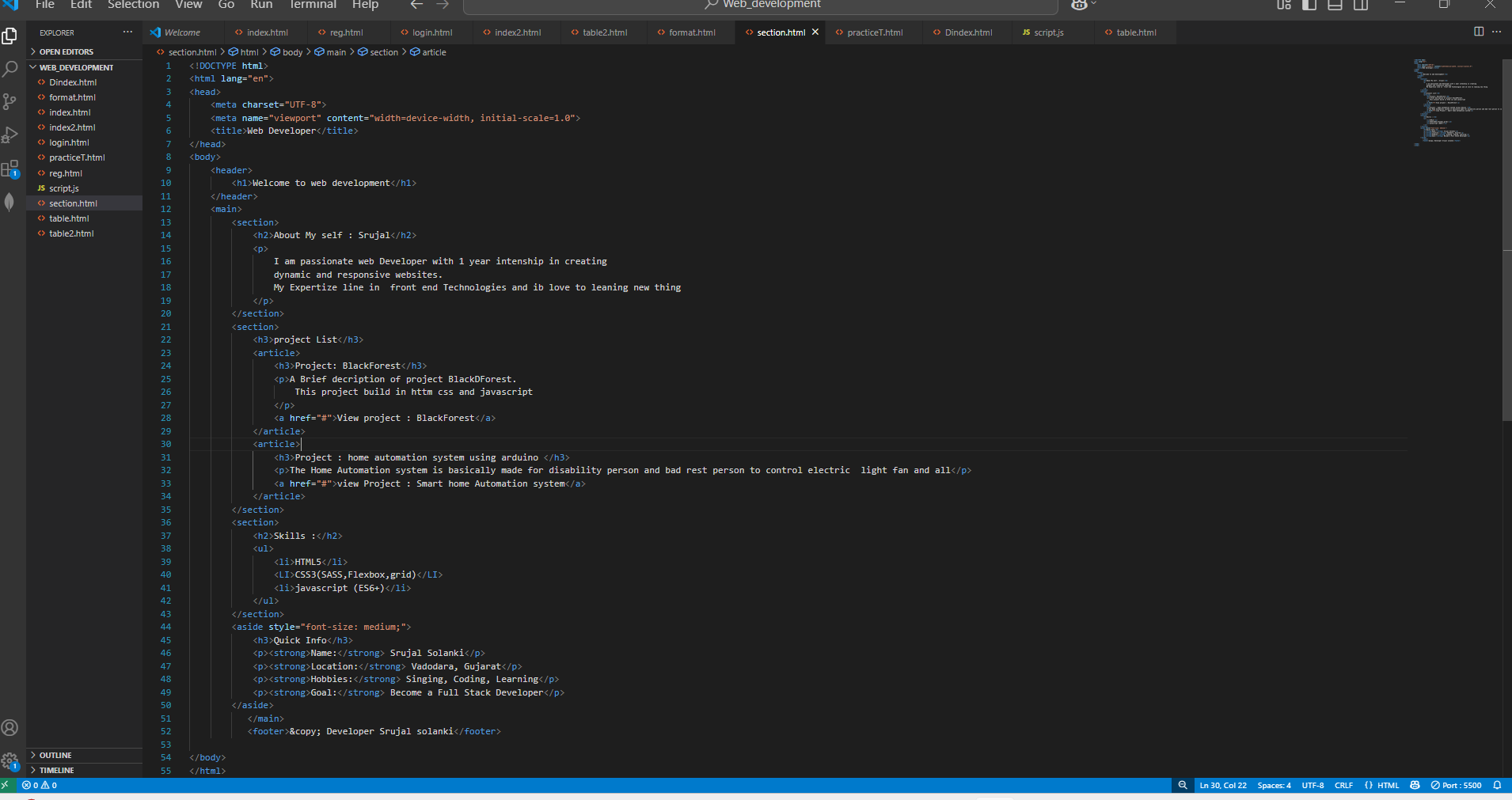
• Task: Create a simple HTML webpage that includes:

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

o A paragraph with some basic text.

o A list (both ordered and unordered).

o A link that opens in a new tab.



2. HTML Forms Theory Assignment

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

TML forms are used to **collect user input** and send it to a server for processing. They allow users to enter data such as names, emails, passwords, feedback, and much more.

**Key Form Elements:**

1. **<input>**  
   Used to accept single-line user input like text, email, password, checkbox, radio, etc.  
   Example:

<input type="text" name="username">

1. **<textarea>**  
   Used for multi-line text input, such as comments or messages.  
   Example:

<textarea name="message"></textarea>

1. **<select>**  
   Used to create a drop-down list of options for users to choose from.  
   Example:

<select name="country">

<option value="in">India</option>

<option value="us">USA</option>

</select>

1. **<button>**  
   Used to submit the form or trigger JavaScript actions.  
   Example:

<button type="submit">Submit</button>

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

|  |  |
| --- | --- |
| GET | POST |
| Sends data as URL parameters (visible in address bar). | Sends data in the request body (not visible in URL). |
| Limited data size (depends on URL length limit). | No size limit (can send large form data). |
| Less secure – not recommended for passwords. | More secure – suitable for sensitive information. |
| Can be bookmarked and cached. | Cannot be bookmarked or cached. |
| Used for search forms, filters, or non-sensitive queries. | Used for login, registration, and feedback forms. |

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

The <label> element defines a **text label** for form controls like <input>, <textarea>, or <select>.

It improves accessibility by:

* Helping **screen readers** read out what input is expected.
* Making it easier for users to **click on the label to focus** the related input.
* Clearly associating a form field with its description.

**Example:**

<label for="username">Username:</label>

<input type="text" id="username" name="username">

In this example, clicking on "Username" will focus the input field. Screen readers will also read "Username" before reading the input box, helping visually impaired users.

Lab Assignment

• Task: Create a contact form with the following fields:

o Full name (text input)

o Email (email input)

o Phone number (tel input)

o Subject (dropdown menu)

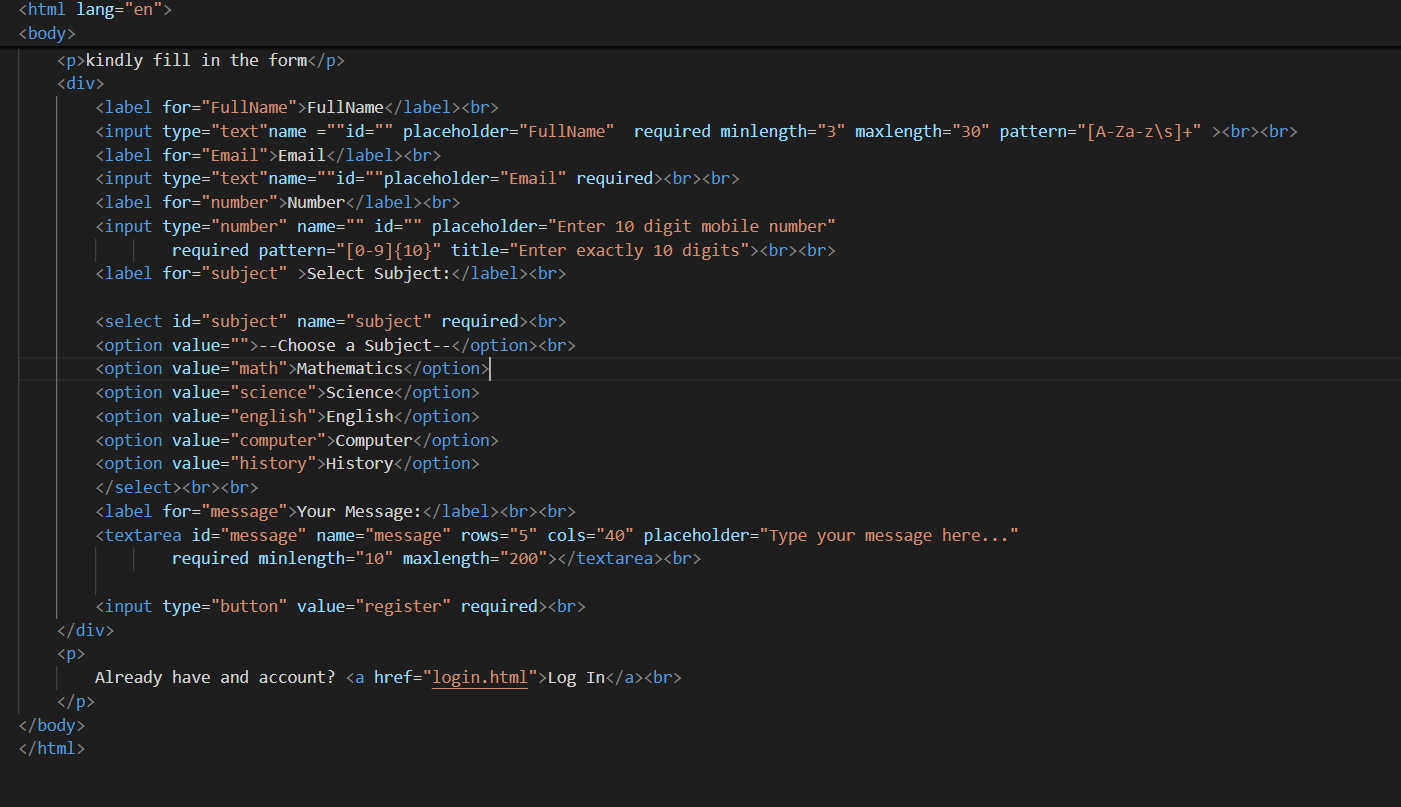
o Message (textarea)

o Submit button

Additional Requirements:

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

o Link form labels with their corresponding inputs using the for attribute.



3. HTML Tables

Theory Assignment

• Question 1: Explain the structure of an HTML table and the purpose of each of the following elements:

An HTML table is used to display data in a tabular format with rows and columns.

**Basic structure:**

<caption>Table title (optional)</caption>

<thead>

<tr>

<th>Header 1</th>

<th>Header 2</th>

</tr>

</thead>

<tbody>

<tr>

<td>Data 1</td>

<td>Data 2</td>

</tr>

</tbody>

<tfoot>

<tr>

<td>Footer 1</td>

<td>Footer 2</td>

</tr>

</tfoot>

</table>

**Explanation of elements:**

* <table>: The container element that defines the table.
* <caption>: Provides a title or summary for the table (optional but useful for accessibility).
* <thead>: Groups the header content of the table. Contains header rows.
* <tbody>: Groups the main body rows of the table.
* <tfoot>: Groups the footer rows of the table (e.g., summary or totals).
* <tr>: Defines a table row.
* <th>: Defines a header cell. Typically displayed bold and centered.
* <td>: Defines a standard data cell within a row.

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

* **colspan** attribute allows a cell to span across multiple columns.
* **rowspan** attribute allows a cell to span across multiple rows.

**Example of colspan:**

<table border="1">

<tr>

<th colspan="2">Name</th>

<th>Age</th>

</tr>

<tr>

<td>John</td>

<td>Doe</td>

<td>30</td>

</tr>

</table>

**Example of rowspan:**

<table border="1">

<tr>

<th rowspan="2">Name</th>

<th>Age</th>

</tr>

<tr>

<th>Location</th>

</tr>

<tr>

<td>John</td>

<td>30</td>

</tr>

<tr>

<td>Doe</td>

<td>USA</td>

</tr>

</table>

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

Lab Assignment

* **Reason to avoid tables for layout:**
  + Tables were originally designed to display tabular data, not for page layout.
  + Using tables for layout makes the HTML complex and less semantic.
  + It harms accessibility because screen readers interpret tables differently.
  + It reduces flexibility and responsiveness on different screen sizes.
  + Tables are harder to maintain and style compared to modern CSS.
* **Better alternative:**
  + Use **CSS layout techniques** such as **Flexbox** and **CSS Grid**.
  + These allow flexible, responsive, and semantic page structures.
  + They separate content (HTML) from presentation (CSS), improving maintainability.

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

o Product Name

o Product Image (use placeholder image URLs)

o Price

o Description

o Availability (in stock, out of stock) Additional Requirements:

o Use thead for the table header.

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

o Use colspan or rowspan to merge cells where applicab

