**Assignment** **- 2 (HTML Form)**

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

* HTML forms are used to collect user input and send it to a server for processing. They are essential in web applications for tasks like logging in, signing up, submitting feedback, placing orders, and more.

1. **<input> Element**

* **Purpose**: Collects single-line user input.
* **Types of input include:**
* text – for plain text input.
* password – for obscured text (e.g., passwords).
* email, number, checkbox, radio, file, date, etc.
* **Example**:
* <input type="text" name="username" placeholder="Enter your username">

1. **<textarea> Element**

* **Purpose:** Collects multi-line text input.
* **Use case:** Ideal for comments, messages, or descriptions.
* **Example:**

<textarea name="message" rows="5" cols="30">Your message here</textarea>

1. **<select> Element**

* **Purpose:** Creates a dropdown list of options.
* **Use case:** When users must choose from a predefined list.
* **Example:**

<select name="country">

<option value="us">United States</option>

<option value="ca">Canada</option>

<option value="uk">United Kingdom</option>

</select>

1. **<button> Element**

* **Purpose:** Triggers actions like submitting the form or running JavaScript.
* **Types:**
* submit – sends the form data to the server.
* reset – clears all fields in the form.
* button – used with JavaScript for custom 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?**

* The GET and POST methods are two HTTP methods used in HTML form submission, and they serve different purposes:

1. **GET Method**

**How it works:**

* Sends form data as URL parameters (appended to the URL as a query string).
* Example:

http://example.com/search?query=books&page=2

**Characteristics:**

* Data is visible in the URL.
* Has length limitations (depends on browser/server limits).
* Can be bookmarked or shared easily.
* Typically cached by browsers.

**When to use GET:**

* When retrieving or querying data without side effects.
* Example use cases:
  + Search forms
  + Filtered listings (e.g., product filters, blog categories)

1. **POST Method**

**How it works:**

* Sends form data in the HTTP request body (not visible in the URL).

**Characteristics:**

* Data is not shown in the URL, offering better privacy.
* No size limitations (more suitable for large data).
* Cannot be bookmarked.
* Not cached by browsers.

**When to use POST:**

* When submitting sensitive or large data, or making changes to server data.
* Example use cases:
  + Login and signup forms
  + Payment or order submissions
  + Feedback or contact forms

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

* The <label> element in an HTML form is used to define a caption or description for a form control (like an <input>, <textarea>, or <select>). It directly associates a text label with a specific form element.

**Purpose of the <label> Element**

1. Describes the input field  
   It tells users what information is expected in the corresponding input field.
2. Improves form usability  
   Clicking the label focuses or activates the associated input (e.g., clicking "Email" focuses the email field).
3. Enhances screen reader support  
   Screen readers read out the label text when the user focuses on the input, helping visually impaired users understand what each field is for.

**How to Use <label> Properly**

There are two main ways to associate a label with an input:

* 1. Using the for attribute (explicit association):

<label for="email">Email:</label>

<input type="email" id="email" name="email">

* 1. Wrapping the input inside the label (implicit association):

<label>

Email:

<input type="email" name="email">

</label>

Both methods are valid, but the first is preferred for better clarity and flexibility.

**Accessibility Benefits**

* Screen readers can announce both the label and the input together.
* Motor-impaired users benefit from larger click targets (the label).
* Ensures users understand form fields even without visible placeholders.
* Helps with form validation messages (e.g., “Please enter a valid email” makes sense when the field has a clear label).