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. They are a fundamental way to create interactive web pages that allow users to submit data, such as login information, search queries, or feedback.
* **<input>** : For single-line input like text, passwords, or files.
* **<textarea>** : For multi-line input like messages or comments.
* **<select>** : For dropdown menus to choose options.
* **<button>:** For actions like submitting or resetting the form.

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

* The **GET** and **POST** methods are used to send form data to a server, but they differ in how they handle and transmit data.
* **GET Method:**
* **Data Transmission:** Sends data appended to the URL as query parameters (e.g., example.com?name=John&age=30).
* **Visibility:** Data is visible in the URL, making it less secure.
* **Size Limit:** Limited amount of data can be sent (due to URL length restrictions).
* **Caching:** Requests can be cached and bookmarked.
* **Use Case:** Suitable for retrieving data or actions with no side effects, such as search queries.
* **POST Method:**
* **Data Transmission:** Sends data in the request body, not visible in the URL.
* **Visibility:** Data is hidden, making it more secure for sensitive information.
* **Size Limit:** Can handle large amounts of data (subject to server limits).
* **Caching:** Requests are not cached or bookmarkable.
* **Use Case:** Suitable for submitting sensitive or large data, such as login forms, file uploads, or any action that modifies server-side data.
* **When to Use:**
* **GET:** When the action is safe, data is non-sensitive, and needs to be cached or bookmarked (e.g., search forms).
* **POST:** When the data is sensitive, large, or changes server-side state (e.g., login, registration, or payments).

3. Why should tables be used sparingly for layout purposes? What is a better

alternative?

* Tables should be used sparingly for layout purposes because they can create a rigid and often inaccessible structure that might not adapt well to different screen sizes and devices. They can also make your code more complex and harder to maintain. Moreover, using tables for layout can disrupt the natural flow of the document, making it less accessible for screen readers and other assistive technologies.
* A better alternative is to use CSS (Cascading Style Sheets) for layout. CSS allows for more flexible and responsive designs, enabling your content to adjust seamlessly to different screen sizes and devices. Techniques like Flexbox and Grid Layout are particularly powerful for creating complex, yet responsive, designs without the drawbacks of table-based layouts.