**Lab Report: 17**

**4. Explain the distinct roles of HTTP methods, particularly GET and POST, in**

**handling form submissions within web applications. Discuss when and why you**

**would choose one method over the other, providing clear examples to illustrate**

**your points.**

* In web applications, “HTTP methods” define how the client (browser) communicates with the server. For handling form submissions, the most commonly used methods are “GET” and “POST”. Each serves distinct purposes:
* **GET Method:**
* **Purpose:** Retrieves data from the server without causing any changes on the server-side.
* **Characteristics:**
* Data is sent as part of the “URL query string” (e.g., `example.com/search?q=books`).
* Limited data size (URL length restrictions).
* Data is visible in the browser’s address bar, making it less secure.
* Bookmarkable and shareable.
* **Use Case:**
* **When to Use:** For non-sensitive operations like search queries or filtering data, where no state is modified on the server.
* **Example:** Submitting a search form on an e-commerce website:
* **HTML:**

<form method="GET" action="/search">

<input type="text" name="q" placeholder="Search">

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

</form>

* **URL after submission:** `/search?q=shoes`.
* **POST Method:**
* **Purpose:** Submits data to the server, typically causing changes (e.g., creating or updating records).
* **Characteristics:**
* Data is sent in the “request body”, not the URL.
* No data size limitations.
* Data is not visible in the address bar, offering better security for sensitive information.
* Cannot be bookmarked or cached.
* **Use Case:**
* **When to Use:** For sensitive or state-changing operations, like user authentication or creating a new record.
* **Example:** Submitting a login form:
* **HTML:**

<form method="POST" action="/login">

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

<input type="password" name="password" placeholder="Password">

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

</form>

* Data is sent securely in the request body.
* **Choosing GET vs. POST:**

|  |  |  |
| --- | --- | --- |
| **Criteria** | **GET** | **POST** |
| Purpose | Retrieve data | Submit or modify data |
| Data Sensitivity | Not secure (data in URL) | More secure (data in request body) |
| Bookmarkable | Yes | No |
| Caching | Cacheable | No Cacheable |
| Data Size | Limited by URL length | No size limitation |

* **Conclusion:**
* Use “GET” for safe, idempotent actions (e.g., search forms, filtering, pagination).
* Use “POST” for state-changing actions or sensitive data (e.g., login, registration, file uploads).