1. Giao thức HTTP

**HTTP (Hypertext Transfer Protocol)** là giao thức truyền tải siêu văn bản. Đây là giao thức tiêu chuẩn cho World Wide Web (www) để truyền tải dữ liệu dưới dạng văn bản, âm thanh, hình ảnh, video từ Web Server tới trình duyệt web của người dùng và ngược lại.

**HTTP** hoạt động theo mô hình **Client (máy khách) – Server (máy chủ).** Việc truy cập website được tiến hành dựa trên các giao tiếp giữa 2 đối tượng trên. Khi bạn truy cập một trang web qua giao thức **HTTP**, trình duyệt sẽ thực hiện các phiên kết nối đến server của trang web đó thông qua địa chỉ **IP** do hệ thống phân giải tên miền **DNS** cung cấp. Máy chủ sau khi nhận lệnh, sẽ trả về lệnh tương ứng giúp hiển thị website, bao gồm các nội dung như: văn bản, ảnh, video, âm thanh,…

2. Giao thức POST và GET

**GET** và **POST** là hai phương thức của giao thức **HTTP**, đều là gửi dữ liệu về server xử lí sau khi người dùng nhập thông tin vào form và thực hiện submit. Trước khi gửi thông tin, nó sẽ được mã hóa bằng cách sử dụng một giản đồ gọi là url encoding. Giản đồ này là các cặp name/value được kết hợp với các kí hiệu = và các kí hiệu khác nhau được ngăn cách bởi dấu &.

2.1 Phương thức **GET** gửi thông tin người dùng đã được mã hóa thêm vào trên yêu cầu trang: **http://www.example.com/index.htm?name=value1&name1=value1**

2.2 Phương thức **POST** truyền thông tin thông qua **HTTP** header, thông tin này được mã hóa như phương thức **GET**. Dữ liệu đươc gửi bởi phương thức POST rất bảo mật vì dữ liệu được gửi ngầm, không đưa lên **URL**, bằng việc sử dụng Secure **HTTP**, bạn có thể chắc chắn rằng thông tin của mình là an toàn. Parameters được truyền trong request body nên có thể truyền dữ liệu lớn, hạn chế tùy thuộc vào cấu hình của Server. Không cache và bookmark được cũng như không được lưu lại trong browser history. **POST** không có bất kì hạn chế nào về kích thước dữ liệu sẽ gửi, có thể gửi dữ liệu nhị phân, hình ảnh.Khi lấy dữ liệu nên dùng GET để truy xuất và xử lí nhanh hơn.

🡺 Lựa chọn: Khi tạo dữ liệu nên dùng POST để bảo mật dữ liệu hơn. Một ví dụ cụ thể như trong trường hợp cần tạo một form để người dùng upload avatar hay video thì chúng ta thường sẽ sử dụng phương thức POST.

Tuy nhiên khi tải về avatar hoặc video thì lại nên sử dụng phương thức GET. Ngoài ra khi cần xử lý các thông tin nhạy cảm ví dụ như email, password thì bạn cần sử dụng POST thay vì GET.

2 Xây dựng Web Form và Form hiển thị dữ liệu từ nhiều bảng

2.1 Xây dựng Web Form

Khái niệm: Form login là giao diện cho phép người dùng nhập thông tin tài khoản (username + password) để xác thực với hệ thống.

Quy trình:

1. Người dùng nhập username + password.

2. Form gửi dữ liệu bằng POST đến server.

3. Server kiểm tra trong database (hoặc mảng user giả lập).

4. Nếu hợp lệ → cấp session/cookie → chuyển hướng vào trang dashboard.

5. Nếu sai → trả về thông báo lỗi.

Chú ý:

• Luôn dùng POST thay vì GET để bảo mật mật khẩu.

• Mật khẩu trong DB phải băm (hash), không lưu plain text (password\_hash, password\_verify trong PHP).

• Có thể bổ sung CSRF Token trong form để chống giả mạo.

2.2 Form hiển thị dữ liệu từ nhiều bảng

Khái niệm: Một form hoặc bảng dữ liệu có thể cần kết hợp thông tin từ nhiều bảng trong CSDL.

Ví dụ: bảng users lưu thông tin học viên, bảng scores lưu điểm số.

• Kỹ thuật:

• Dùng JOIN trong SQL (INNER JOIN, LEFT JOIN, …).

• Lấy dữ liệu từ nhiều bảng → kết hợp → hiển thị dạng bảng HTML.

• Ứng dụng thực tế:

• Hiển thị danh sách đơn hàng (kết hợp bảng orders và customers).

• Hiển thị thông tin học viên kèm điểm số nhiều môn.

Chú ý:

• Tối ưu câu lệnh JOIN để tránh nặng nề.

• Có thể phân trang (pagination) khi dữ liệu quá nhiều.

3 Làm việc với Cookie và Session

3.1 Cookie

Định nghĩa: Cookie là dữ liệu nhỏ lưu ở trình duyệt client, do server gửi xuống.

Đặc điểm:

• Lưu trên máy người dùng.

• Có hạn sử dụng (expire time).

• Thích hợp cho “nhớ đăng nhập”, lưu theme, ngôn ngữ, giỏ hàng nhỏ.

3.2 Session

Định nghĩa: Session là cơ chế lưu thông tin trên server, gắn với một session\_id gửi cho client (thường qua cookie PHPSESSID).

Đặc điểm:

• Lưu trên server → an toàn hơn cookie.

• Dùng để lưu trạng thái đăng nhập, giỏ hàng, quyền hạn người dùng.

• Hết session khi đóng trình duyệt hoặc server xoá.