**Bài 9: Tổng quan ứng dụng web**

**1. HTTP là gì? HTTP và HTTPS khác nhau như thế nào?**

• **HTTP (HyperText Transfer Protocol)** là giao thức truyền tải siêu văn bản, giúp trao đổi dữ liệu giữa máy khách (client) và máy chủ (server).

• **HTTPS (HyperText Transfer Protocol Secure)** là phiên bản bảo mật của HTTP, có sử dụng **SSL/TLS** để mã hóa dữ liệu nhằm bảo vệ thông tin khỏi bị đánh cắp hoặc thay đổi.

**So sánh HTTP và HTTPS:**

|  |  |  |
| --- | --- | --- |
| **Tiêu chí** | **HTTP** | **HTTPS** |
| Bảo mật | Không có mã hóa | Có mã hóa bằng SSL/TLS |
| Tốc độ | Nhanh hơn do không mã hóa | Chậm hơn một chút do mã hóa dữ liệu |
| Cổng mặc định | 80 | 443 |
| Ứng dụng | Trang web công khai, không yêu cầu bảo mật | Trang web giao dịch tài chính, đăng nhập, thanh toán online |

**2. Dynamic Web và Static Web**

|  |  |  |
| --- | --- | --- |
| **Tiêu chí** | **Static Web** | **Dynamic Web** |
| Nội dung | Cố định, không thay đổi theo người dùng | Thay đổi theo dữ liệu đầu vào và logic xử lý |
| Công nghệ | HTML, CSS, JavaScript | PHP, JSP, ASP.NET, Node.js… |
| Dữ liệu | Không kết nối CSDL | Có kết nối CSDL |
| Ví dụ | Trang giới thiệu công ty, portfolio | Facebook, Gmail, Shopee |

**3. Phân biệt Request và Response trong mô hình Client - Server**

• **Request (Yêu cầu)**: Do **client (trình duyệt, ứng dụng)** gửi lên server để yêu cầu tài nguyên (HTML, JSON, hình ảnh…).

• **Response (Phản hồi)**: Do **server** gửi về client với nội dung được xử lý dựa trên request.

**Ví dụ quá trình hoạt động:**

1. Người dùng nhập www.google.com → Trình duyệt gửi một **HTTP Request** đến Google.

2. Server của Google xử lý và trả về một **HTTP Response** chứa mã HTML của trang tìm kiếm.

**4. Phân biệt GET và POST**

|  |  |  |
| --- | --- | --- |
| **Tiêu chí** | **GET** | **POST** |
| Truyền dữ liệu | Dữ liệu được gắn trên URL | Dữ liệu được gửi trong body của request |
| Bảo mật | Không an toàn (URL có thể bị lộ) | An toàn hơn (Dữ liệu không hiển thị trên URL) |
| Giới hạn dữ liệu | Bị giới hạn (khoảng 2048 ký tự) | Không giới hạn |
| Ứng dụng | Lấy dữ liệu (search, truy vấn) | Gửi dữ liệu (đăng nhập, đăng ký, upload file) |

**Cách gửi dữ liệu GET và POST:**

1. **GET**:

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

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

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

</form>

→ Dữ liệu sẽ được gửi dưới dạng: /search?query=abc

2. **POST**:

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

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

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

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

</form>

→ Dữ liệu được gửi đi trong phần **body**, không hiển thị trên URL.

**5. JSP và Servlet là gì? Phân biệt JSP và Servlet**

• **JSP (Java Server Pages)** và **Servlet** là công nghệ của Java dùng để phát triển **web động**.

• **Servlet**: Là Java class xử lý yêu cầu HTTP, được sử dụng để xử lý logic.

• **JSP**: Là trang web có thể nhúng Java code để hiển thị nội dung động.

**So sánh JSP và Servlet:**

|  |  |  |
| --- | --- | --- |
| **Tiêu chí** | **Servlet** | **JSP** |
| Viết code | Viết bằng Java thuần | Nhúng Java vào HTML |
| Dễ viết | Khó hơn do phải sử dụng nhiều Java code | Dễ hơn, có thể viết HTML trực tiếp |
| Tốc độ | Nhanh hơn | Chậm hơn một chút do cần biên dịch |
| Ứng dụng | Xử lý logic, điều hướng request | Giao diện người dùng (hiển thị nội dung) |