|  |  |
| --- | --- |
| Phân biệt HTTP/HTTPS | **HTTP (HyperText Transfer Protocol)**:  - Là giao thức truyền tải dữ liệu không được mã hóa.  - Dữ liệu dễ bị đánh cắp bởi các tấn công man-in-the-middle.  - Sử dụng cổng **80**.  - Không có cơ chế bảo mật tích hợp.  **HTTPS (HyperText Transfer Protocol Secure)**:  - Là phiên bản bảo mật của HTTP, sử dụng giao thức mã hóa SSL/TLS.  - Dữ liệu được mã hóa, đảm bảo tính toàn vẹn và an toàn.  - Sử dụng cổng **443**.  - Yêu cầu chứng chỉ SSL/TLS để xác minh tính hợp lệ của website. |
| Phân biệt dynamic web/static web | **Static Web (Trang web tĩnh)**:  - Nội dung không thay đổi hoặc thay đổi rất ít.  - Được xây dựng bằng các file HTML, CSS, JavaScript cơ bản.  - Không có tương tác hoặc xử lý phía server.  - Ví dụ: Trang giới thiệu công ty, hồ sơ cá nhân.  **Dynamic Web (Trang web động)**:  - Nội dung thay đổi dựa trên tương tác của người dùng hoặc dữ liệu từ server.  - Sử dụng các ngôn ngữ lập trình như PHP, Python, JSP, hoặc frameworks như React, Angular.  - Thường tích hợp cơ sở dữ liệu để lưu trữ và xử lý thông tin.  - Ví dụ: Mạng xã hội, trang thương mại điện tử. |
| Phân biệt phương thức get/post | **GET**:  - Dữ liệu được gửi kèm trên URL.  - Thích hợp cho các yêu cầu không cần bảo mật, ví dụ: truy vấn tìm kiếm.  - Giới hạn kích thước dữ liệu (URL bị giới hạn độ dài).  - Không bảo mật vì dữ liệu dễ bị nhìn thấy.  **POST**:  - Dữ liệu được gửi trong body của request.  - Thích hợp cho các yêu cầu cần bảo mật, ví dụ: gửi form đăng nhập.  - Không giới hạn kích thước dữ liệu (phụ thuộc vào server).  - Bảo mật hơn so với GET. |

|  |  |
| --- | --- |
| JSP, Servlet là gì? Phân biệt? | **JSP (Java Server Pages)**:  - Là công nghệ Java để tạo các trang web động.  - Cho phép nhúng mã Java vào HTML thông qua các thẻ JSP.  - Dễ dàng thiết kế giao diện hơn vì tập trung vào HTML.  **Servlet**:  - Là lớp Java chạy trên server, xử lý logic và giao tiếp với client.  - Chủ yếu dùng để xử lý dữ liệu và không phù hợp với việc hiển thị giao diện.  **Phân biệt**:  - JSP phù hợp cho việc xây dựng giao diện (front-end).  - Servlet phù hợp để xử lý logic (back-end). |
| Phân biệt request/response ở mô hình Client – Server? | **Request (Yêu cầu)**:  - Được gửi từ client (trình duyệt hoặc ứng dụng) đến server.  - Có thể chứa dữ liệu (parameters, headers, body).  - Ví dụ: Người dùng gửi yêu cầu tải một trang web hoặc gửi dữ liệu form.  **Response (Phản hồi)**:  - Được gửi từ server về client để trả lời request.  - Chứa thông tin kết quả như HTML, JSON, trạng thái HTTP (200, 404, 500).  - Ví dụ: Trả về trang web hoặc kết quả của một API. |
| Phân biệt forward/redirect? | **Forward**:  - Chuyển tiếp yêu cầu từ một Servlet hoặc JSP đến một tài nguyên khác trên cùng server.  - URL trên trình duyệt không thay đổi.  - Không tạo một yêu cầu HTTP mới.  - Thích hợp cho các tác vụ nội bộ.  **Redirect**:  - Yêu cầu trình duyệt gửi một request mới đến URL khác.  - URL trên trình duyệt thay đổi.  - Tạo ra một yêu cầu HTTP mới.  -Thích hợp cho việc điều hướng đến tài nguyên ngoài hoặc trang mới. |