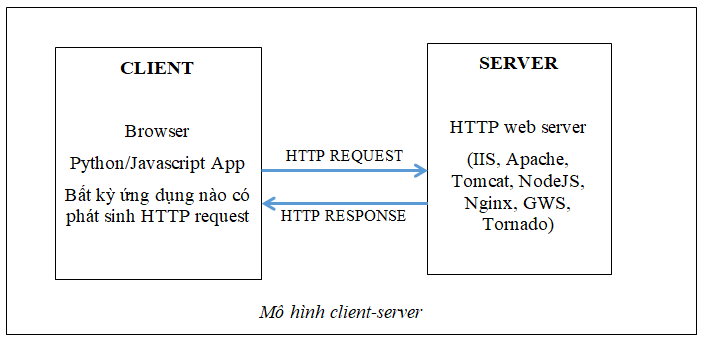
1.1 Hoạt động của một ứng dụng web

1.1.1 Mô hình client-server

Mọi trang web, hoặc ứng dụng web đều hoạt động dựa trên mô hình client-server. Vì vậy chúng ta sẽ cùng tìm hiểu về mô hình này.

Ý tưởng của mô hình này, đơn giản chỉ là: máy khách (client) gửi một yêu cầu (request) đến máy chủ (server), máy chủ sẽ xử lý và trả kết quả về cho máy khách.

Xem hình minh họa về mô hình client-server:



Mô hình client-server gồm một số thành phần:

– Client: khởi phát yêu cầu, gửi yêu cầu tới server, nhận kết quả từ server trả về. Client có thể là trình duyệt, ứng dụng viết bằng python/javascript hoặc bất kỳ ứng dụng nào mà có phát sinh ra HTTP request.

– Server: có vai trò cung cấp dịch vụ, xử lý và trả về kết quả cho máy client.

– Môi trường truyền thông tin: hạ tầng mạng (LAN, Internet), bao gồm cả phần cứng và phần mềm

– Giao thức truyền thông tin: các chuẩn công nghệ giúp giao tiếp và truyền thông tin giữa server-client. Ví dụ HTTP, HTTPS

Trong mô hình client-server, khi môi trường truyền đã được kết nối và sẵn sàng, thì client luôn là nơi khởi phát của ứng dụng. Client sẽ gửi một HTTP request tới server. Khi server nhận được request, nó sẽ xử lý và trả kết quả về cho client bằng một HTTP response.

Client là thành phần chủ động. Server là thành phần bị động.

Chúng ta sẽ thấy cách hoạt động của một ứng dụng web rất khác so với cách hoạt động của chương trình trên máy cục bộ (ví dụ Microsoft Word). Trong Microsoft Word, mọi thao tác đều được thực hiện ngay trên máy người dùng, từ việc ra lệnh, xử lý và trả về kết quả. Cũng là chương trình xử lý văn bản, nhưng Google Docs là một ứng dụng hoạt động theo mô hình client-server. Nếu không có kết nối mạng thì Google Docs sẽ không hoạt động được.

Để nắm rõ hơn về mô hình client-server, chúng ta sẽ cùng tìm hiểu về quá trình duyệt web diễn ra như thế nào ở phần tiếp theo.

1.1.2 Quá trình duyệt web diễn ra như thế nào

Do ứng dụng web hoạt động theo mô hình client-server, nên để có trang web trên trình duyệt, cần trải qua các bước sau:

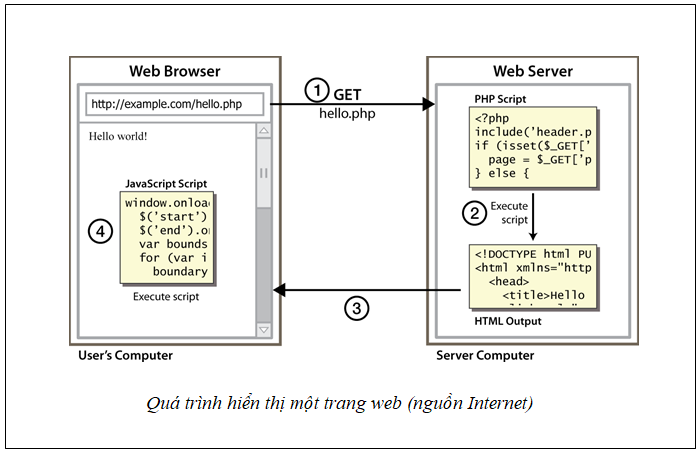
Bước 1: Người dùng nhập địa chỉ trang web (URL) vào thanh địa chỉ. Ví dụ: http://example.com/hello.php

Bước 2: Trình duyệt dựa vào URL trong thanh địa chỉ, kết nối tới máy web server, gửi yêu cầu tới web server (ví dụ yêu cầu: gửi cho nội dung trang web hello.php)

Bước 3: Web server xử lý yêu cầu, gửi trả kết quả về cho trình duyệt (ví dụ nội dung trang web dưới dạng mã nguồn HTML, CSS và JavaScript)

Bước 4: Trình duyệt thực thi mã HTML, CSS, JavaScript và hiển thị trang web ra màn hình.

Xem hình minh họa:



Chúng ta cùng thực hành, quan sát quá trình duyệt web trên một máy tính:

Bước 1. Mở trình duyệt web. Ví dụ Chrome

Bước 2. Nhập vào đường dẫn của trình duyệt địa chỉ một trang web, bấm Enter để trình duyệt lấy trang web từ máy server, và hiển thị nội dung ra màn hình. Ví dụ thanhnien.vn

Bước 3. Mở Developer tools của trình duyệt bằng một số cách sau:

– Bấm tổ hợp ba phím Ctrl+Shift+I,

– hoặc bấm phím F12,

– hoặc vào menu của trình duyệt tìm tới mục Developer tools,

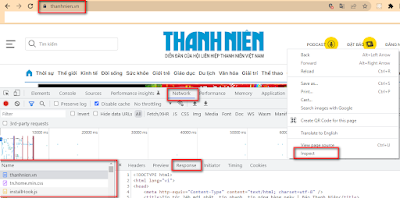
– hoặc bấm chuột phải vào trang web, chọn Inspect.

Bước 4. Trong cửa sổ của Developer tools, bấm chuột vào mục Network.

Bước 5. Quan sát sẽ thấy các tập tin HTML (thanhnien.vn), CSS (các tập tin có phần mở rộng là css), JavaScript (các tập tin có phần mở rộng là js) được server gửi về cho client.

Bước 6. Bấm chuột vào các tập tin do server gửi về và quan sát nội dung của nó ở cửa sổ bên phải. Nhớ chọn mục Response.

Hình minh họa:

[](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEhhYaoyD8trJPiTmXP1lJdQtPW86CYtPXcdCgZ1a-OftKkUbfnMtITdfNzo8PyVDmI32mwsynxbGME8Z3MYRSQ7oNXqUtGUKdQrM8OWuvc3KBv0PGkMsamtGQw9QLK5a-13ZaQO7AuyD4lQTeCuPMeWPN0qyfUBIKtdQ1MNEM1pjhV7OqKN_7q2g3eIWyY/s1118/Back%20end%20can%20ban.png)

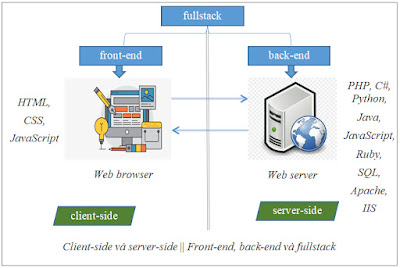
**1.1.3**       **Client side/server side/front end/back end/fullstack**

Ứng dụng web hoạt động dựa trên kiến trúc client-server, nghĩa là sẽ có các xử lý được thực hiện tại phía client (client-side), và có các xử lý được thực hiện ở phía server (server-side).

Các xử lý tại client-side bao gồm việc gửi yêu cầu tới server, nhận mã HTML, CSS và JavaScript từ server, thực thi mã và hiển thị kết quả cho người dùng.

Các xử lý tại server-side bao gồm việc nhận yêu cầu do client gửi tới, làm việc với cơ sở dữ liệu, xử lý yêu cầu, trả kết quả về cho client.

Trong lĩnh vực web, client-side bao gồm các hoạt động diễn ra tại trình duyệt, server-side bao gồm các hoạt động diễn ra tại web server.

[](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEicC0CKSrUmG7oIC2QM5KRCt9aM7NwT2E3metllF6SX-2pFnWpkW9KkhVixv9UaUzT9l_Yop_9iox-jcmJCfHjG7SQlrc4DQ46ps1DXT2tSylBY_7deEfKOx4fA17DRPAoyKVZbz3e2WW9nPZE6HwpZKJUf9WLSTxdwGG0SKodrN6ukm2wBXP6Q93Z1Tj4/s748/lap%20trinh%20front%20end%20back%20end.jpg)

Như vậy để tạo ra một ứng dụng web hoàn chỉnh, cần phải viết các đoạn mã xử lý ở cả bên server và bên client; đồng thời phải phối hợp các đoạn mã này một cách hợp lý để đảm bảo ứng dụng chạy đúng, an toàn và hiệu quả.

Việc viết mã xử lý tại phía client thường được gọi là lập trình phía client, hay lập trình front-end; các ngôn ngữ thường được sử dụng gồm HTML, CSS, và JavaScript. Front-end là toàn bộ giao diện và các trải nghiệm của người dùng trên trình duyệt.

Việc viết mã xử lý tại phía server thường được gọi là lập trình phía server, hay lập trình back-end; các ngôn ngữ thường được sử dụng gồm PHP, C#, Java, Python, Ruby, JavaScript. Back-end là các xử lý liên quan đến cơ sở dữ liệu, các xử lý nghiệp vụ chạy tại máy server.

Tùy theo khả năng, một lập trình viên có thể chọn làm ở mảng front-end, hoặc back-end hoặc cả hai (full-stack). Dù làm ở mảng front-end thì bạn cũng phải biết căn bản về mảng back-end và ngược lại.

**1.1.4**       **Xem và đọc thêm**

– [1] Từ khóa: kiến trúc client-server, quá trình hiển thị một trang web, How The Web Works

– [2] Kiến trúc client-server: https://vi.wikipedia.org/wiki/M%C3%B4\_h%C3%ACnh\_client-server

– [3] How The Web Works - The Big Picture (view): https://www.youtube.com/watch?v=hJHvdBlSxug

– [4] How The Web Works - The Big Picture (read): https://academind.com/tutorials/how-the-web-works/

– [5] CSR và SSR: https://www.freecodecamp.org/news/server-side-rendering-javascript/