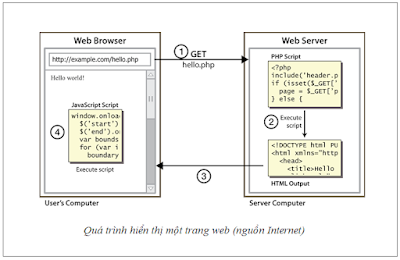
Câu 1. Quá trình duyệt web

Phân tích các bước trong hình ảnh sau



1. Phân tích hình sau

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.

1. Là một lập trình viên, bạn sẽ phải cài đặt hay cấu hình những nội dung gì để có một hệ thống chạy được như vậy?

Để có một hệ thống chạy được như trong hình ảnh, một lập trình viên cần cài đặt hoặc cấu hình các nội dung sau:

* **Máy chủ web:** Đây là máy tính lưu trữ các trang web. Máy chủ web có thể là một máy tính vật lý hoặc một máy ảo.
* **Hệ điều hành:** Máy chủ web cần chạy một hệ điều hành. Hệ điều hành phổ biến nhất cho máy chủ web là Linux và Windows.
* **Ngôn ngữ lập trình:** Máy chủ web cần được lập trình bằng một ngôn ngữ lập trình. Ngôn ngữ lập trình phổ biến nhất cho máy chủ web là PHP, Python và Node.js.
* **Framework web:** Framework web là một bộ công cụ và thư viện giúp lập trình viên phát triển trang web nhanh hơn và dễ dàng hơn. Một số framework web phổ biến là Laravel, Django và Angular.
* **Database:** Máy chủ web cần lưu trữ dữ liệu của trang web trong một cơ sở dữ liệu. Cơ sở dữ liệu phổ biến nhất cho trang web là MySQL, PostgreSQL và MongoDB.
* **Công cụ tự động hóa:** Lập trình viên có thể sử dụng các công cụ tự động hóa để giúp họ quản lý và bảo trì hệ thống. Một số công cụ tự động hóa phổ biến là Ansible, Chef và Puppet.

Ngoài ra, lập trình viên cũng cần cấu hình các cài đặt sau:

* **Cấu hình máy chủ web:** Lập trình viên cần cấu hình máy chủ web để chạy ứng dụng web.
* **Cấu hình cơ sở dữ liệu:** Lập trình viên cần cấu hình cơ sở dữ liệu để lưu trữ dữ liệu của ứng dụng web.
* **Cấu hình bảo mật:** Lập trình viên cần cấu hình hệ thống để đảm bảo an ninh.

1. Bạn biết gì về server-side rendering và client-side rendering, liên hệ thực tế?

Server-side rendering (SSR) và client-side rendering (CSR) là hai phương pháp hiển thị trang web.

**Server-side rendering** là phương pháp hiển thị trang web hoàn chỉnh trên máy chủ web trước khi gửi cho trình duyệt web. Khi trình duyệt web nhận được trang web, nó chỉ cần tải và hiển thị trang web.

**Client-side rendering** là phương pháp hiển thị trang web một cách dần dần trên trình duyệt web. Khi trình duyệt web nhận được trang web, nó sẽ tải các tệp HTML, CSS và JavaScript. Sau đó, trình duyệt web sẽ sử dụng các tệp này để hiển thị trang web.

Liên hệ thực tế:

Server-side rendering có một số ưu điểm so với client-side rendering, bao gồm:

Tốc độ tải trang nhanh hơn: Trang web được hiển thị hoàn chỉnh trên máy chủ web trước khi gửi cho trình duyệt web. Điều này giúp cải thiện tốc độ tải trang cho người dùng.

Tăng SEO: Trang web được hiển thị hoàn chỉnh trên máy chủ web, bao gồm cả nội dung tĩnh và nội dung động. Điều này giúp cải thiện thứ hạng trang web trong kết quả tìm kiếm của Google.

Tuy nhiên, server-side rendering cũng có một số nhược điểm, bao gồm:

Yêu cầu máy chủ web mạnh mẽ hơn: Máy chủ web cần đủ mạnh để tạo ra các trang web hoàn chỉnh cho mỗi yêu cầu.

Tăng độ phức tạp của phát triển: Phát triển trang web bằng server-side rendering có thể phức tạp hơn so với client-side rendering.

Client-side rendering có một số ưu điểm so với server-side rendering, bao gồm:

Tăng tương tác người dùng: Trang web được hiển thị dần dần trên trình duyệt web, cho phép người dùng tương tác với trang web ngay cả khi trang web vẫn đang tải.

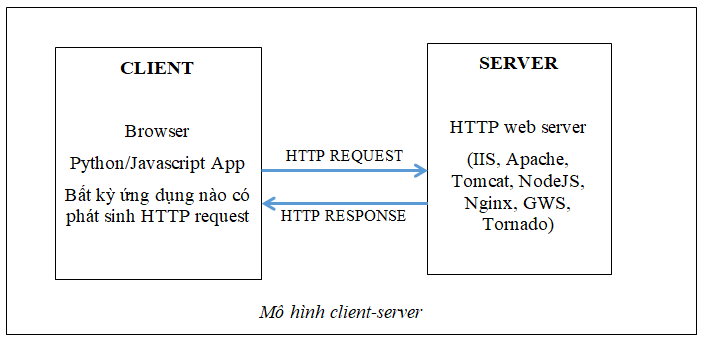
Giảm yêu cầu tài nguyên của máy chủ web: Máy chủ web chỉ cần gửi các tệp HTML, CSS và JavaScript cho trình duyệt web.

Tuy nhiên, client-side rendering cũng có một số nhược điểm, bao gồm:

Tốc độ tải trang chậm hơn: Trang web được hiển thị dần dần trên trình duyệt web. Điều này có thể ảnh hưởng đến tốc độ tải trang cho người dùng.

Giảm SEO: Trang web chỉ được hiển thị hoàn chỉnh trên trình duyệt web. Điều này có thể ảnh hưởng đến thứ hạng trang web trong kết quả tìm kiếm của Google.

Câu 2. 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.

Câu hỏi 1: Quan sát lộ trình học (roadmap) dưới đây. Bạn đã biết, hiểu và thực hành với mỗi chủ đề như thế nào?

Câu trả lời:

Theo lộ trình học (roadmap) được cung cấp, tôi đã biết, hiểu và thực hành với các chủ đề sau:

**Internet**: Tôi đã biết về cách thức hoạt động của Internet, bao gồm các khái niệm cơ bản như domain name, DNS, HTTP, và các giao thức mạng khác. Tôi cũng đã thực hành việc sử dụng các công cụ và dịch vụ Internet, chẳng hạn như trình duyệt web, email, và các dịch vụ đám mây.

**DNS**: Tôi đã biết về cách thức hoạt động của DNS, bao gồm cách thức chuyển đổi domain name thành địa chỉ IP. Tôi cũng đã thực hành việc sử dụng các công cụ và dịch vụ DNS, chẳng hạn như dig và nslookup.

**HTTP**: Tôi đã biết về cách thức hoạt động của HTTP, bao gồm cách thức truyền dữ liệu giữa các máy tính trên mạng. Tôi cũng đã thực hành việc sử dụng các công cụ và dịch vụ HTTP, chẳng hạn như curl và wget.

**Domain Name:** Tôi đã biết về cách thức hoạt động của domain name, bao gồm cách thức đăng ký và quản lý domain name. Tôi cũng đã thực hành việc sử dụng các công cụ và dịch vụ domain name, chẳng hạn như GoDaddy và Namecheap.

**Browsers and how they work:** Tôi đã biết về cách thức hoạt động của trình duyệt web, bao gồm cách thức tải trang web, giải mã HTML, CSS, và JavaScript. Tôi cũng đã thực hành việc sử dụng các công cụ và dịch vụ trình duyệt web, chẳng hạn như Chrome DevTools và Firefox Developer Tools.

**Basic Frontend Knowledge:** Tôi đã biết về các khái niệm cơ bản của Frontend, bao gồm HTML, CSS, và JavaScript. Tôi cũng đã thực hành việc sử dụng các công cụ và dịch vụ Frontend, chẳng hạn như Visual Studio Code và Sublime Text.

Cụ thể:

**Internet**: Tôi đã học về cách thức hoạt động của Internet từ các nguồn tài liệu trực tuyến và sách giáo khoa. Tôi cũng đã thực hành việc sử dụng các công cụ và dịch vụ Internet bằng cách truy cập các trang web, sử dụng email, và sử dụng các dịch vụ đám mây.

**DNS**: Tôi đã học về cách thức hoạt động của DNS từ các nguồn tài liệu trực tuyến và sách giáo khoa. Tôi cũng đã thực hành việc sử dụng các công cụ và dịch vụ DNS bằng cách sử dụng các lệnh dig và nslookup.

**HTTP**: Tôi đã học về cách thức hoạt động của HTTP từ các nguồn tài liệu trực tuyến và sách giáo khoa. Tôi cũng đã thực hành việc sử dụng các công cụ và dịch vụ HTTP bằng cách sử dụng các lệnh curl và wget.

**Domain Name:** Tôi đã học về cách thức hoạt động của domain name từ các nguồn tài liệu trực tuyến và sách giáo khoa. Tôi cũng đã thực hành việc sử dụng các công cụ và dịch vụ domain name bằng cách đăng ký và quản lý domain name trên các trang web như GoDaddy và Namecheap.

**Browsers and how they work:** Tôi đã học về cách thức hoạt động của trình duyệt web từ các nguồn tài liệu trực tuyến và sách giáo khoa. Tôi cũng đã thực hành việc sử dụng các công cụ và dịch vụ trình duyệt web bằng cách sử dụng các công cụ như Chrome DevTools và Firefox Developer Tools để kiểm tra cách thức trình duyệt web hoạt động.

**Basic Frontend Knowledge**: Tôi đã học về các khái niệm cơ bản của Frontend từ các khóa học trực tuyến và sách giáo khoa. Tôi cũng đã thực hành việc sử dụng các công cụ và dịch vụ Frontend bằng cách tạo các trang web đơn giản bằng HTML, CSS, và JavaScript.

Tôi vẫn đang tiếp tục học hỏi và thực hành các chủ đề này. Tôi hy vọng rằng với những kiến thức và kỹ năng mà tôi đã học được, tôi có thể phát triển các trang web và ứng dụng web chất lượng cao.

Câu hỏi 2: Bạn có nhận xét gì về lộ trình học này?

Câu trả lời:

Nhìn chung, tôi thấy lộ trình học này khá toàn diện và đầy đủ. Nó bao gồm các chủ đề cơ bản cần thiết để hiểu cách thức hoạt động của Internet và phát triển các trang web và ứng dụng web.

Tuy nhiên, tôi cũng có một số nhận xét về lộ trình học này, bao gồm:

\*\*Cần bổ sung thêm các chủ đề về Back-end, chẳng hạn như Python, Node.js, và MySQL. Các chủ đề này là cần thiết