

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ**

Giáo trình

XÂY DỰNG ỨNG DỤNG WEB VỚI PHP
(Building Web applications with PHP)

TS. NGUYỄN HỮU HÒA (Chủ biên)

Ths. VÕ HẢI ĐĂNG

Ths. HỒ VĂN TÚ

Ths. LÊ MINH LÝ

2021

LỜI GIỚI THIỆU

Nhà Xuất bản Đại học Cần Thơ chân thành cảm ơn các tác giả và sự đóng góp ý kiến của quý thầy cô trong Hội đồng thẩm định trường Đại học Cần Thơ để giáo trình “Công nghệ Web” được ra mắt bạn đọc.

Nhà Xuất bản Đại học Cần Thơ trân trọng giới thiệu đến sinh viên, giảng viên và bạn đọc giáo trình này.

NHÀ XUẤT BẢN ĐẠI HỌC CẦN THƠ

MỤC LỤC

Lời giới thiệu.....	i
Mục lục	iii
GIỚI THIỆU VỀ LẬP TRÌNH WEB.....	2
2.1 Lịch sử World Wide Web	2
2.2 Ứng dụng Web.....	6
2.2.1 <i>Giao thức HTTP</i>	6
2.2.2 <i>Giao thức HTTPS</i>	9
2.2.3 <i>Mô hình ứng dụng web</i>	10
2.3 Ngôn ngữ lập trình Web	14
2.3.2 <i>Ngôn ngữ lập trình Front-end</i>	15
2.3.3 <i>Ngôn ngữ lập trình Back-end</i>	17
Tài liệu tham khảo	25

CHƯƠNG 1

GIỚI THIỆU VỀ LẬP TRÌNH WEB

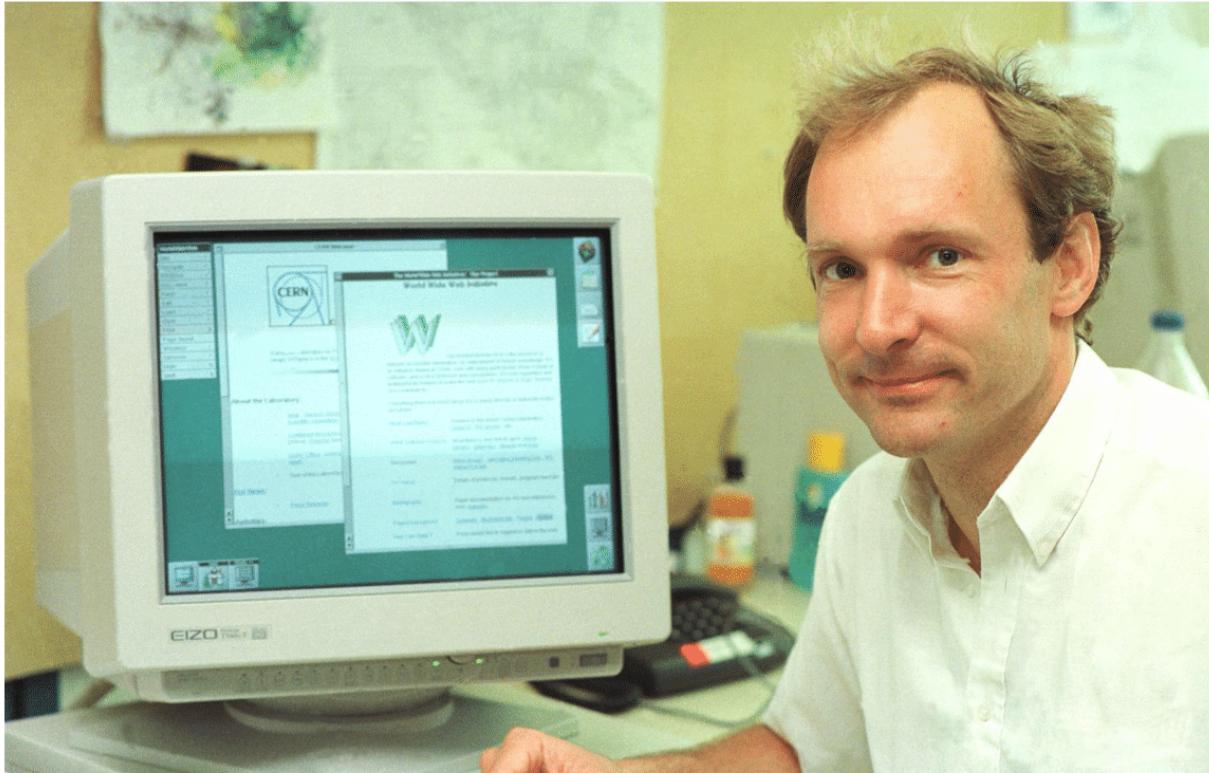
Chương này giới thiệu về lịch sử ra đời của Web, các giao thức HTTP, máy chủ web, các ngôn ngữ lập trình front-end, back-end và các framework đang được sử dụng phổ biến ngày nay.

2.1 LỊCH SỬ WORLD WIDE WEB

World Wide Web, thường được gọi tắt là Web, là một mạng thông tin không ngừng tiến hóa, vượt xa quan niệm ban đầu của nó vào đầu những năm 1990, khi mà nó được tạo ra để giải quyết một vấn đề cụ thể. Bắt nguồn từ phòng thí nghiệm CERN¹, nơi là đầu mối của hơn 17 nghìn nhà khoa học tại hơn 100 quốc gia trên toàn thế giới tham gia nghiên cứu, thực hiện các thí nghiệm mới. Mặc dù các nghiên cứu xuất phát từ phòng thí nghiệm CERN, nhưng các thí nghiệm được thực hiện bởi các nhà khoa học ở những trường đại học hay tại các quốc gia mà họ đang sinh sống. Các thí nghiệm đã tạo ra một lượng dữ liệu đáng kinh ngạc, nhiều đến mức dữ liệu được xem là không thể chia sẻ cho các nhà khoa học tham gia nghiên cứu trên toàn thế giới. Do vậy một công cụ đáng tin cậy để giao tiếp, trao đổi thông tin là rất cần thiết lúc bấy giờ.

Vào thời điểm này Internet đã xuất hiện, với hàng trăm ngàn máy tính đã được kết nối, vì vậy nhà khoa học người Anh, Tim Berners-Lee (một thành viên của CERN) đã nghĩ đến một phương pháp nhằm chia sẻ thông tin, dữ liệu một cách thuận tiện hơn. Ông đã phát minh ra phương pháp điều hướng dữ liệu bằng cách sử dụng siêu liên kết, thông qua giao thức HTTP (Hypertext Transfer Protocol). Ông cũng chính là người đã tạo ra ngôn ngữ HTML để biểu diễn thông tin. Những thông tin được định vị bởi bộ định vị tài nguyên URI (Uniform Resource Identifier). Bằng cách kết hợp URI, HTTP và HTML ông đã tạo nên Web.

¹ CERN là tên viết tắt của Conseil Européen pour la Recherche Nucléaire theo tiếng Pháp, là Tổ chức Nghiên cứu Hạt nhân châu Âu, tiếng Anh có tên là European Organisation for Nuclear Research. Đây là phòng thí nghiệm vật lý hạt lớn nhất thế giới, nằm phía Tây Bắc ngoại ô Geveva, Thụy Sỹ.



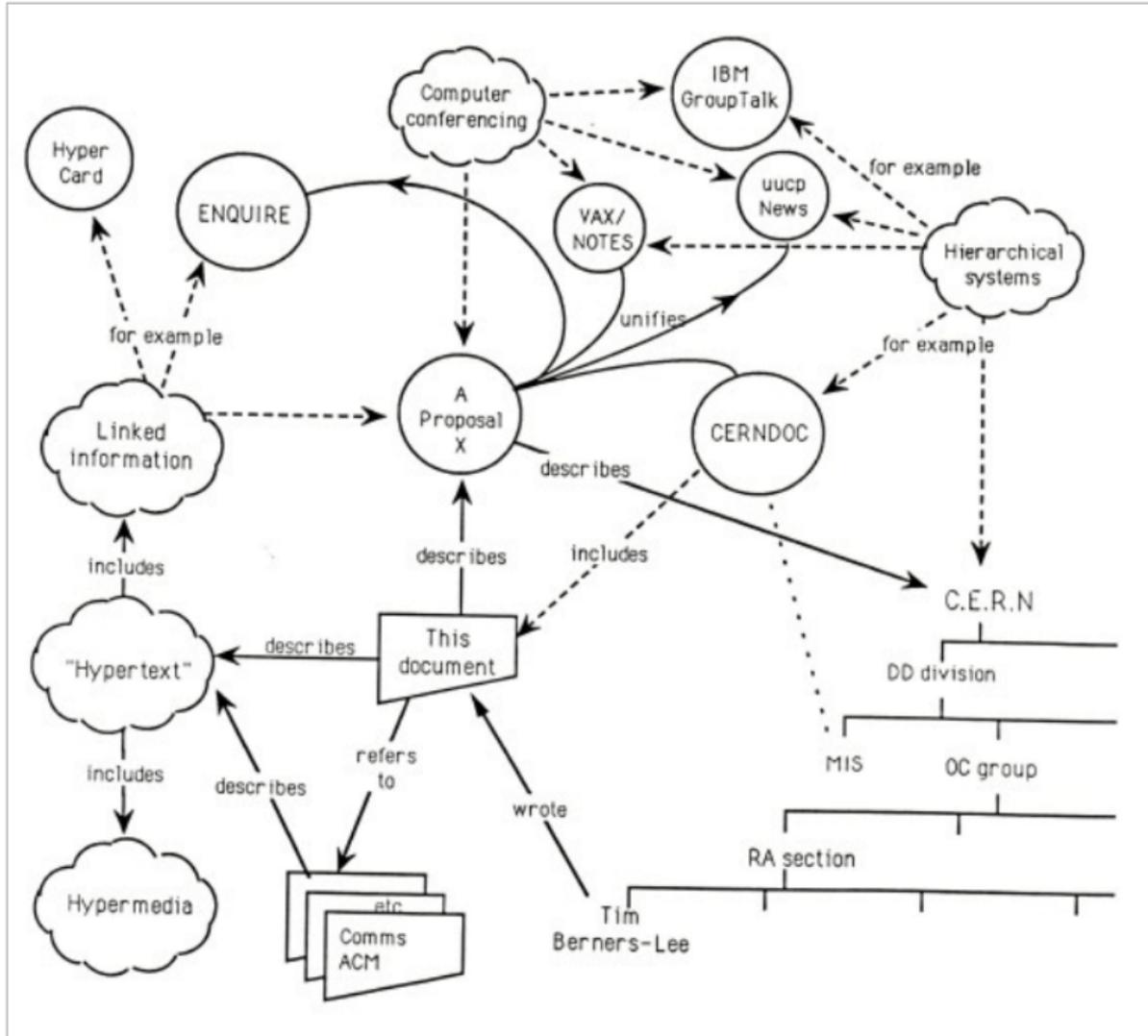
Hình 0.1 Tim Berners-Lee khi còn làm việc tại CERN

Vào những năm đầu 1990, Web được xem là một khái niệm mang tính cách mạng, bởi tại thời điểm đó những người sử dụng máy tính - phần lớn là những nhà khoa học chỉ có thể liên lạc với nhau bằng cách kết nối đến BBS² thông qua terminal để gửi tin nhắn, hoặc truyền dữ liệu thông qua chế độ dòng lệnh. Để kết nối, người dùng phải là thành viên của các BBS đó. Điều này gây cản trở rất lớn trong quá trình chia sẻ thông tin cho cộng đồng.

Ý tưởng cơ bản của Web là hợp nhất các công nghệ đang phát triển của máy tính, các mạng dữ liệu và siêu văn bản thành một hệ thống thông tin toàn cầu mạnh mẽ và dễ sử dụng.

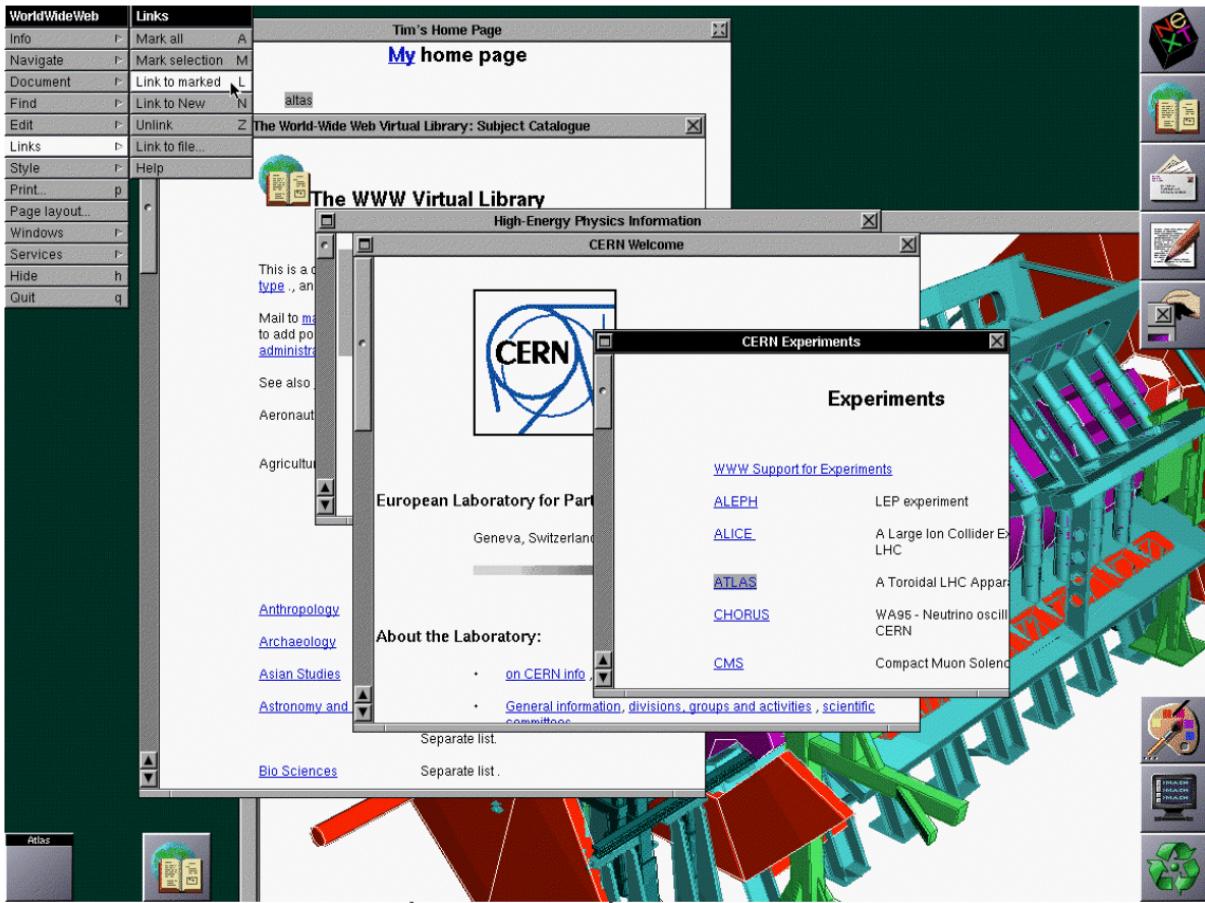
Tim Berners-Lee đã viết đề xuất cho Web vào tháng 3 năm 1989 và đề xuất thứ hai của ông vào tháng 5 năm 1990[1]. Cùng với kỹ sư hệ thống người Bỉ Robert Cailliau, ông đã hoàn chỉnh bản đề xuất và nó được chính thức hóa (formalized) vào năm 1990 với tên WorldWideWeb: Proposal for a HyperText Project[2].

² BBS – Bulletin Board Service là một dịch vụ tương tự như Web Server ngày nay nhưng chỉ cho phép kết nối thông qua terminal.



Hình 0.2 Ý tưởng của Tim Berners-Lee về Web trong bản đê xuất

Cuối năm 1990, Berners-Lee đã cho ra đời máy chủ web (Web server) và trình duyệt web (Web Browser) đầu tiên trên thế giới có tên WorldWideWeb[3] (sau này được đổi thành Nexus) để chứng minh ý tưởng của mình. Hệ thống được vận hành tại CERN. Ông đã viết mã lệnh cho máy chủ web trên máy tính NeXT. Website đầu tiên trên thế giới có địa chỉ <http://info.cern.ch> và trang web đầu tiên là <http://info.cern.ch/hypertext/WWW/TheProject.html>.



Hình 0.3 Trình duyệt web đầu tiên trên thế giới

Đến năm 1993 trình duyệt thứ hai xuất hiện có tên Mosaic, sau này đổi tên thành Netscape Navigator. Từ khi Nexus và Netscape xuất hiện, nó đã làm thay đổi rất nhiều cách tiếp cận thông tin của người dùng khi hỗ trợ đồ họa, hình ảnh trực quan và đa phương tiện. Các web browser này đã thống trị Internet trong những năm 1990. Đến năm 1995, Microsoft phát hành phiên bản đầu tiên của trình duyệt web Internet Explorer (IE) nhằm cạnh tranh trực tiếp với Netscape Navigator, vốn đang thống lĩnh thị phần trình duyệt web. Trong vòng bốn năm sau khi phát hành, IE đã chiếm được 75% thị phần, và đến năm 1999, nó đã đạt 99%[5].

Các trình duyệt web thế hệ tiếp theo lần lượt ra đời bởi các hãng công nghệ lớn nhằm cạnh tranh với IE vốn đã thống lĩnh thị phần trình duyệt trong một thời gian dài, có thể kể đến như Safari, FireFox, Chrome và Opera. Sau khi bị chiếm thị phần bởi các trình duyệt đối thủ, IE đã được Microsoft cải tiến và phát hành phiên bản mới với tên Microsoft Edge.

Năm 1994, Berners-Lee đã chuyển đến Viện công nghệ Massachusetts (Massachusetts Institute of Technology - MIT) và thành lập W3C (World Wide

Web Consortium), là một tổ chức quốc tế chịu trách nhiệm phát triển các tiêu chuẩn cho Web. W3C hiện có hơn 400 thành viên, bao gồm các trường đại học, các công ty, chính phủ, các tổ chức phi lợi nhuận và các cá nhân. Tổ chức này duy trì đội ngũ thành viên tích cực làm việc cùng nhau trong việc phát triển các tiêu chuẩn cho World Wide Web.

2.2 ỨNG DỤNG WEB

Kể từ khi W3C ra đời, nó đã đặt ra nhiều tiêu chuẩn cho ứng dụng Internet, đặc biệt là ứng dụng Web. Để hiểu rõ ứng dụng web trước khi bắt tay vào xây dựng nó, chúng ta cần phải hiểu rõ cách thức mà ứng dụng Web hoạt động, tìm hiểu về các giai đoạn và kiến trúc của nó.

2.2.1 Giao thức HTTP

Ứng dụng Web hoạt động trên nền tảng giao thức mạng TCP/IP. Nó sử dụng giao thức HTTP (Hypertext Transfer Protocol), là một giao thức cho phép tìm và truy xuất các tài nguyên, chẳng hạn như tài liệu HTML. Nó là nền tảng của bất kỳ trao đổi dữ liệu nào trên Web, và nó là một giao thức hoạt động theo kiến trúc client-server. Điều đó có nghĩa là các yêu cầu truy xuất được khởi tạo bởi người dùng (client), thường là trình duyệt (browser) và gửi đến máy chủ web (Web server). Khi máy chủ tiếp nhận yêu cầu, các nguồn tài nguyên như văn bản, hình ảnh, video hay các kịch bản (script) sẽ được tổng hợp và trả về kết quả cho client. Khi thông điệp yêu cầu được gửi từ client, nó được gọi là HTTP Request. Khi server nhận được yêu cầu và gửi trả kết quả, nó được gọi là HTTP Response.

2.2.1.1 Các phiên bản

Kể từ khi ra đời đến thời điểm hiện tại, HTTP đã trải qua nhiều phiên bản khác nhau.

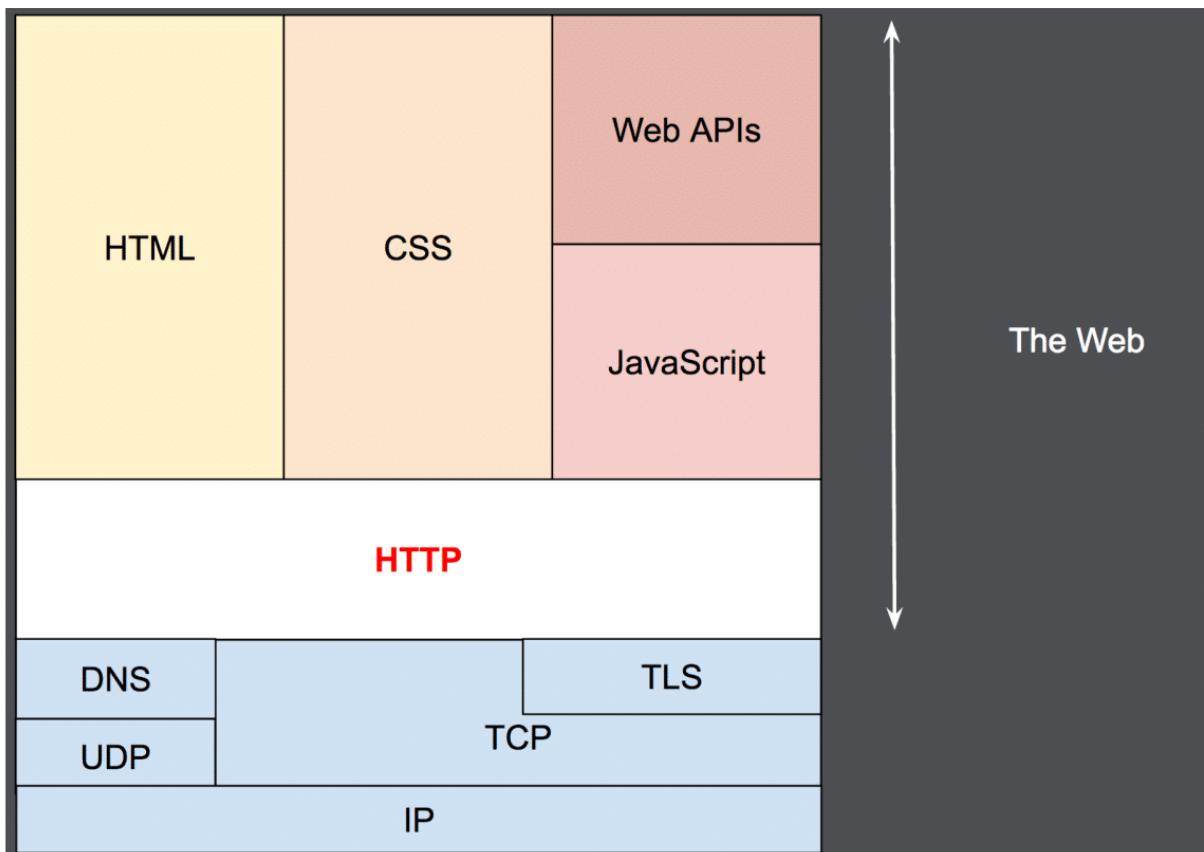
Bản đặc tả đầu tiên của HTTP được xuất bản năm 1999 với số hiệu phiên bản 0.9, ký hiệu HTTP/0.9.

Phiên bản tài liệu đặc tả thứ hai được của HTTP được xuất bản năm 1996 có số hiệu phiên bản 1.0, ký hiệu HTTP/1.

Phiên bản thứ ba có số hiệu 1.1 được xuất bản một năm sau đó, năm 1997 với ký hiệu HTTP/1.1.

Phiên bản thứ ba, HTTP/2 được phát triển bởi Google, tài liệu đặc tả được xuất bản năm 2015 với nhiều cải tiến so với phiên bản HTTP/1.1, được sử dụng rất phổ biến trên các website ngày nay.

Bản thảo của phiên bản ba, HTTP/3 được công bố đầu năm 2021. Mặc dù nó chưa được chính thức xuất bản nhưng nó đã được sử dụng khá nhiều trong các ứng dụng web. Ước tính có khoảng 20% website đã hỗ trợ HTTP/3. Các trình duyệt Google Chrome và FireFox là những trình duyệt tiên phong trong hỗ trợ phiên bản mới này.



Hình 0.4 Mô hình hoạt động của Web.

2.2.1.2 *Thông điệp yêu cầu và đáp ứng*

Giao thức HTTP giúp client và server giao tiếp với nhau thông qua HTTP Request và HTTP Response. Đây là các thông điệp được gửi qua lại giữa client và server trong quá trình kết nối.

Thông điệp HTTP Request giúp client chuyển tiếp yêu cầu đến server có dạng như sau:

```
GET /en-US/docs/Glossary/index.html HTTP/1.1  
Host: developer.mozilla.org
```

```
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9; rv:50.0)  
Gecko/20100101 Firefox/50.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate, br  
Referer: https://developer.mozilla.org/en-US/docs/Glossary/index.html
```

(parameters)

Trong thông điệp này gồm có 2 phần, phần header giúp client mô tả phương thức yêu cầu (GET, HEAD, POST, PUT, DELETE, CONNECT, OPTIONS, TRACE, PATCH), nơi thông tin cần truy cập (URI), phiên bản HTTP sử dụng, khai báo thông tin của client (User-agent), loại kết quả được chấp nhận trả về và ngôn ngữ, dữ liệu mã hóa chấp nhận và một số các thông tin liên quan khác. Phần body giúp client gửi kèm thêm các tham số (parameters), ví dụ thông tin người dùng nhập vào trang web như tên đăng nhập và mật khẩu hay là yêu cầu tìm kiếm nội dung có chứa từ khóa cần tìm.

Khi web server nhận yêu cầu, nó sẽ xử lý và trả về kết quả bằng một thông điệp đáp ứng HTTP Response, có dạng như sau:

```
HTTP/1.1 200 OK  
Connection: Keep-Alive  
Content-Encoding: gzip  
Content-Type: text/html; charset=utf-8  
Date: Wed, 20 Jul 2016 10:55:30 GMT  
Etag: "547fa7e369ef56031dd3bff2ace9fc0832eb251a"  
Keep-Alive: timeout=5, max=1000  
Last-Modified: Tue, 19 Jul 2016 00:59:33 GMT  
Server: Apache  
Transfer-Encoding: chunked  
Vary: Cookie, Accept-Encoding
```

(content)

Tương tự, HTTP Response cũng gồm hai phần, phần header chứa phiên bản HTTP, mã trạng thái (200 là OK)³, các thông tin kết nối, loại tài liệu trả về và một số thông tin khác. Phần body chứa kết quả trả về cho client, thông thường là tài liệu HTML, XML, JSON hoặc văn bản thuần (plain text).

2.2.1.3 *Phiên làm việc*

Phiên làm việc (session) là một lần giao dịch giữa client-server. Điều này có nghĩa là khi client khởi tạo yêu cầu bằng cách thiết lập kết nối qua giao thức TCP tới cổng cụ thể trên máy chủ, thường là cổng 80, lúc này session được bắt đầu. Máy chủ HTTP sẽ lắng nghe trên cổng dịch vụ HTTP (mặc định là 80) để chờ nhận yêu cầu từ client. Khi nhận được yêu cầu, server sẽ trả về thông điệp, lúc này session sẽ được kết thúc.

Mỗi lần giao tiếp là mỗi lần session được tạo ra và kết thúc ngay sau đó, do vậy server sẽ không ghi nhớ trạng thái giao dịch giữa client và server ở những lần trước đó. Chính vì vậy mà giao thức HTTP được gọi là giao thức phi trạng thái.

Các ứng dụng web ngày nay là các ứng dụng ghi nhớ trạng thái. Ví dụ, khi người dùng truy cập vào một website thương mại điện tử và thực hiện thao tác đăng nhập, tên đăng nhập và mật khẩu sẽ được gửi đến máy chủ web để thực hiện thao tác xác thực người dùng. Khi người dùng cung cấp thông tin hợp lệ, trạng thái đã đăng nhập sẽ được lưu trữ cho người dùng đó. Khi đó, người dùng chuyển qua trang có bảo mật (không cho truy cập khi chưa đăng nhập), ví dụ đơn hàng, web server sẽ không yêu cầu người dùng đăng nhập nữa. Vì giao thức HTTP không hỗ trợ ghi nhớ các lần giao dịch, do đó các ứng dụng web phải tự cài đặt các cơ chế ghi nhớ trạng thái, ví dụ sử dụng cookies phía máy chủ web server hoặc các biến ẩn bên trong các form.

2.2.2 Giao thức HTTPS

HTTPS (Hypertext Transfer Protocol Secure) là giao thức mở rộng của giao thức HTTP. Nó được sử dụng trong các giao dịch cần được bảo mật, ví dụ như

³ Mã trạng thái mô tả kết quả giao dịch giữa client và server. Ví dụ 200 ~ OK, 400 ~ Bad request.

Xem thêm các mã trạng thái của HTTP tại <https://developer.mozilla.org/en-US/docs/Web/HTTP/Status>

internet banking. Các giao dịch sẽ được mã hóa sử dụng TLS (Transport Layer Security) hoặc giao thức cũ hơn SSL (Secure Sockets Layer).

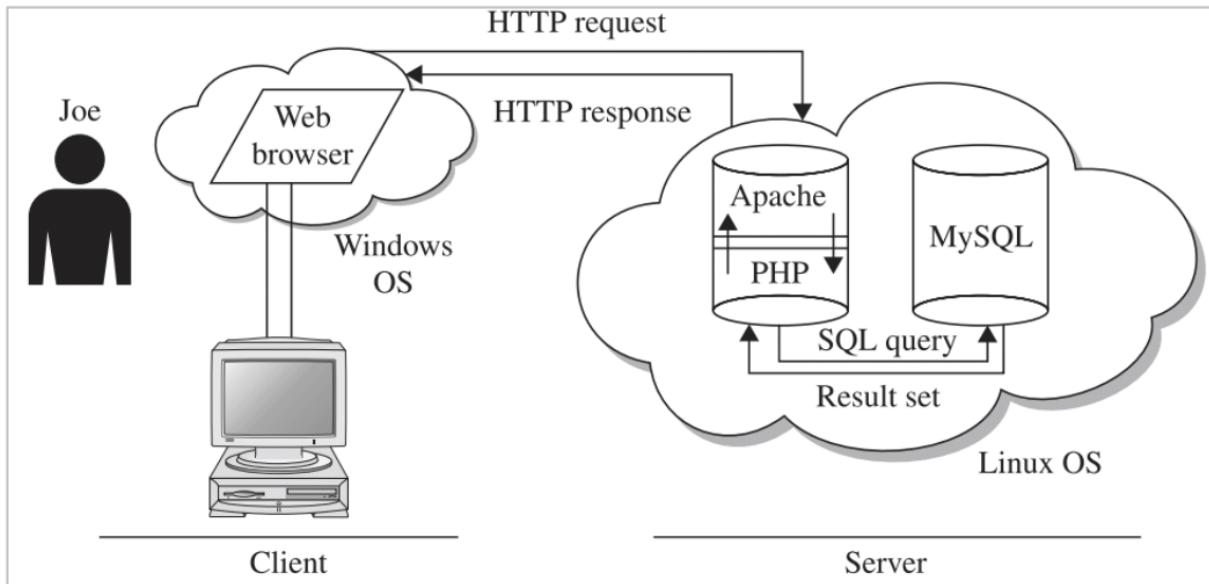
Các website sử dụng giao thức HTTPS có địa chỉ bắt đầu bằng https:// và sử dụng cổng mặc định 443, trong khi đó địa chỉ giao thức HTTP bắt đầu bằng http:// và sử dụng cổng mặc định là 80.

Giao thức HTTPS sử dụng cơ chế chứng thực ba bên, điều đó có nghĩa là giao dịch sẽ được chứng thực bởi bên thứ ba, được gọi là CA (Certificate Authorities). Do vậy, các website sử dụng giao thức này cần phải mua chứng chỉ số để chứng thực cho website của mình. Cơ chế này nằm giúp xác thực chủ sở hữu của website, tránh việc website bị giả mạo để đánh cắp thông tin người dùng.

2.2.3 Mô hình ứng dụng web

Như đã đề cập, ứng dụng web hoạt động dựa trên mô hình client-server. Client thông thường là các trình duyệt web hoặc các chương trình ứng dụng, ví dụ BOT được tạo ra để lấy tin tức từ nhiều nguồn khác nhau. Client sẽ gửi yêu cầu và nhận kết quả trả về, hiển thị kết quả trên màn hình.

Server là một máy chủ, được cài đặt các chương trình phục vụ web, được gọi là web server. Nó có nhiệm vụ lắng nghe các yêu cầu từ client, xử lý các yêu cầu và trả về kết quả. Như vậy, nhiệm vụ web server là lắng nghe yêu cầu từ client, phân tích yêu cầu, chuyển yêu cầu đến ứng dụng web, nhận kết quả trả về từ ứng dụng web và trả kết quả về cho client thông qua thông điệp đáp ứng. Hình 5 mô tả mô hình hoạt động của ứng dụng web.



Hình 0.5 Mô hình hoạt động của ứng dụng web

2.2.3.1 *Web server*

Để các ứng dụng web hoạt động được, nó phải được đặt trên máy chủ được cài đặt phần mềm máy chủ web nhằm chuyển tiếp yêu cầu từ client. Phần mềm này có khả năng phục vụ hàng triệu kết nối cùng một lúc. Hiện nay có rất nhiều phần mềm web server tồn tại ở cả hình thức miễn phí và phần mềm thương mại, hỗ trợ các nền tảng hệ điều hành và các ngôn ngữ lập trình khác nhau.

Hàng triệu website trên toàn thế giới đang được host bởi nhiều web server khác nhau. Một số web server được sử dụng phổ biến có thể kể đến như Apache, IIS, Nginx, GWS. Mỗi phần mềm có cách cài đặt và cấu hình riêng. Để sử dụng được các phần mềm này, chúng ta cần phải đọc các tài liệu hướng dẫn đi kèm để thiết lập các thông số phù hợp với mục đích.

① Apache HTTP server

Là phần mềm máy chủ web miễn phí, thường được gọi là Apache, được phát triển bởi Apache Software Foundation. Phiên bản đầu tiên được phát hành năm 1995. Apache được sử dụng rất phổ biến để vận hành các website, ước tính có hơn 30% website trên thế giới đang được vận hành bởi phần mềm này. Nó có thể hoạt động trên nhiều hệ điều hành bao gồm Unix, Linux, MacOS, Windows và OpenVMS. Apache được viết bằng ngôn ngữ lập trình C, có thể mở rộng các tính năng thông qua việc cài đặt thêm các modules. Hiện nay nó hỗ trợ các ngôn ngữ lập trình web đang được sử dụng phổ biến như Perl, Python, Tcl và PHP.

② Apache Tomcat

Là web server cũng do Apache phát triển để vận hành các website được xây dựng bằng ngôn ngữ lập trình Java web như Servlet và JavaServer Page (JSP). Nó được viết bằng ngôn ngữ lập trình Java, có thể hoạt động trên mọi nền tảng tương tự như Apache.

③ Microsoft Internet Information Services (IIS)

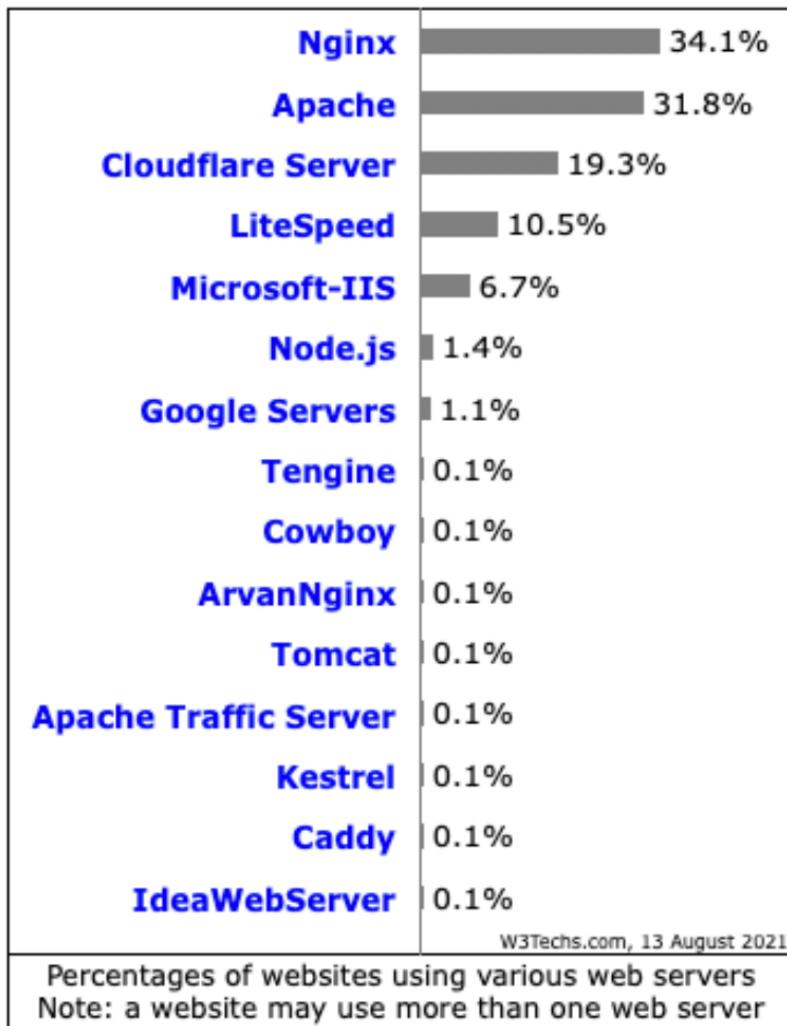
Là một web server được Microsoft phát triển để hoạt động trên hệ điều hành Windows. IIS hỗ trợ giao thức HTTP, HTTP/2, HTTPS, FTP, FTPS, SMTP và NNTP. Nó được tích hợp vào hệ điều hành từ Windows NT 4.0 cách nay hơn 20 năm. Phiên bản hiện tại IIS 10 được tích hợp vào Windows Server 2019. IIS hỗ trợ vận hành các website viết bằng ngôn ngữ .NET như ASP.NET và ASP.NET Core.

Các phiên bản trước đây như IIS 5.0 trở về trước chỉ hỗ trợ vận hành các website ASP (Active Server Page), là ngôn ngữ kịch bản phía server (server-side scripting language) được phát triển bởi Microsoft. Từ năm 2002, ASP.NET ra đời thay thế ASP, cho phép lập trình viên có thể viết mã lệnh cho ứng dụng web bằng các ngôn ngữ C#, F# và Visual Basic. Phiên bản mới phát hành cho phép IIS có thể host các website được viết bằng các ngôn ngữ khác như Java, PHP, Python.

④ Nginx

Nginx (phát âm engine-x) là phần mềm mã nguồn mở được phát triển bởi kỹ sư phần mềm người Nga Igor Sysoev năm 2004. Thời gian gần đây Nginx đã vượt qua Apache dành vị trí dẫn đầu các web server đang host website trên thế giới. Được biết đến như web server có hiệu năng cao (high-performance), Nginx có thể phục vụ 500 ngàn yêu cầu trong một giây với mức sử dụng CPU thấp. Nó sử dụng cơ chế cân bằng tải (load balancer) để phân phối công việc và lưu lượng mạng giữa các máy chủ để tăng hiệu năng đáp ứng.

Hiện nay các website lớn đang được vận hành bởi Nginx, có thể kể đến như LinkedIn, Adobe, Xerox, Facebook và Twitter.



Thống kê web server của w3techs.com

⑤ lighttpd

Đúng như tên gọi của nó lighttpd (phát âm lighty) là một web server rất nhỏ gọn, chiếm rất ít tài nguyên hệ thống khi vận hành. Mục tiêu nó được thiết kế nhằm vận hành các website vừa và nhỏ. Theo thông tin được công bố trên trang chủ (<https://www.lighttpd.net/story>), nó có thể xử lý 10 nghìn kết nối cùng một lúc (parallel connection). Lighttpd cũng hỗ trợ cân bằng tải, FastCGI, TLS/SSL, giao thức HTTP/1 và HTTP/2.

Lighttpd hoàn toàn miễn phí, nó được sử dụng chủ yếu trên Linux/Unix, Raspberry Pi và MacOS. Các ngôn ngữ lập trình web có thể được host bởi lighttpd như PHP, Python, Perl và Ruby.

⑥ NodeJS

NodeJS có thể được sử dụng như là một ngôn ngữ lập trình phía máy chủ, đồng thời cũng đóng vai trò như một web server để host các website được phát triển bởi ngôn ngữ lập trình JavaScript. NodeJS được xây dựng trên nền tảng V8 JavaScript Engine của Google. Nó là một phần mềm mã nguồn mở, hoạt động đa nền tảng. Nó được xem như là một cuộc cách mạng lập trình web khi mà ngôn ngữ JavaScript có thể lập trình phía máy chủ.

Trong những năm gần đây, việc phát triển thần tốc của NodeJS đã kéo theo sự ra đời nhiều công nghệ web, có thể kể đến như sự ra đời của các JavaScript backend framework (AngularJS, VueJS, React, EmberJS, ExpressJS, NextJS, NuxJS) và lập trình đa nền tảng (hybrid-app) như React-Native.

⑦ Google Web Server (GWS)

Là phần mềm máy chủ web độc quyền mà Google sử dụng cho cơ sở hạ tầng web của mình. GWS được sử dụng độc quyền trong hệ sinh thái của Google để host các website. Do đây là phần mềm độc quyền nên người sử dụng không thể tải về và cài đặt giống như các phần mềm máy chủ web khác.

2.3 NGÔN NGỮ LẬP TRÌNH WEB

Hiện nay có rất nhiều ngôn ngữ lập trình web khác nhau được sử dụng bởi các nhà phát triển phần mềm. Mỗi ngôn ngữ có thể mạnh riêng của nó. Do vậy, trước khi bắt đầu xây dựng phần mềm, các nhà phát triển phải tìm hiểu rất kỹ yêu cầu của phần mềm, từ đó xem xét lựa chọn ngôn ngữ lập trình cho phù hợp với mục đích cụ thể. Trong một dự án phần mềm, đôi khi các nhà phát triển cần phải sử dụng nhiều ngôn ngữ lập trình khác nhau cho từng module hoặc phân hệ nhằm tận dụng tối đa công nghệ mà nó hỗ trợ, từ đó nâng cao hiệu quả của phần mềm. Vậy, để đưa ra được giải pháp phù hợp chúng ta cần phải biết thế mạnh của từng ngôn ngữ lập trình, các thư viện, các công nghệ mà nó hỗ trợ để đưa ra quyết định đúng đắn.

Ngày nay, ngôn ngữ lập trình được phân chia làm hai loại: Front-end và Back-end.

2.3.2 Ngôn ngữ lập trình Front-end

Ngôn ngữ lập trình front-end là ngôn ngữ lập trình phía máy khách. Theo cơ chế hoạt động của web, tất cả các xử lý đều thực hiện ở phía máy chủ. Lấy ví dụ, khi người sử dụng tạo tài khoản đăng nhập, người dùng cần phải điền các thông tin cá nhân và địa chỉ email, sau đó gửi (submit) đến server và chờ đợi kết quả trả về. Giả sử, người sử dụng nhập vào email đã được đăng ký (đã tồn tại) trên hệ thống, khi đó hệ thống sẽ trả về thông báo lỗi, người dùng sẽ phí khoảng thời gian chờ đợi. Đó là cách lập trình trên các website trước đây khi công nghệ còn sơ khai. Với cách lập trình kiểu mới, người dùng không cần phải đợi kết quả trả về từ server và nhận được thông báo lỗi. Khi người sử dụng nhập vào email và di chuyển sang ô khác, lập tức email sẽ được âm thầm gửi lên server để kiểm tra đã được sử dụng hay chưa.

Hiện nay ngôn ngữ lập trình front-end chủ yếu là JavaScript (JS). Mặc dù có nhiều ngôn ngữ lập trình khác như CoffeScript, LiveScript, TypeScript, Dart, Kotlin nhưng tất cả chúng đều phải biên dịch sang mã JavaScript để có thể thực thi trên trình duyệt. Ngoài ra, một số các ngôn ngữ lập trình có thể được thực thi trên trình duyệt nhưng ngày nay không còn được sử dụng phổ biến như Flash, VBScript, Silverlight và Java Applets.

Để lập trình front-end, các lập trình viên thường sử dụng các thư viện (library) hoặc các khung (framework) để tiết kiệm thời gian và công sức xây dựng nền tảng cho ứng dụng. Các framework đang rất thịnh hành trong lập trình front-end ngày nay như Angular, VueJS, React.

2.3.2.1 *Angular*

Là front-end framework được phát triển bởi Google từ năm 2016. Angular giúp lập trình viên xây dựng giao diện người dùng (UI – User Interface) có khả năng tương tác cao với người sử dụng. Nó là một framework mã nguồn mở, giúp cho các lập trình viên có thể phát triển ứng dụng SPAs (single-page web application) một cách hoàn hảo. Với SPA, người dùng sẽ có cảm giác như đang sử dụng phần mềm trên nền tảng desktop hơn là ứng dụng web, bởi nó đã loại bỏ cơ chế gửi-chờ-nhận kết quả từ server như web truyền thống.

Phiên bản đầu tiên được đặt tên AngularJS, kể từ phiên bản thứ 2, kiến trúc của nó đã được thay đổi, để phân biệt phiên bản 1 và các phiên bản tiếp theo, tên của nó đã được bỏ đi "JS".

Angular có thể sử dụng ngôn ngữ JS hoặc TypeScript (TS) để lập trình. Ngày nay các lập trình viên có xu hướng chuyển sang TS do JS không hỗ trợ kiểu lập trình hướng đối tượng. TS ra đời nhằm khắc phục các khuyết điểm của JS. Khi lập trình bằng ngôn ngữ TS, nó sẽ tự động biên dịch sang mã JS khi ứng dụng được build.

2.3.2.2 *React*

React là một thư lập trình UI tương tự Angular. Nó được Facebook phát triển và cho ra đời phiên bản đầu tiên vào năm 2013 dưới dạng phần mềm mã nguồn mở.

React cho phép các lập trình viên tạo ra ứng dụng web qui mô lớn, có thể nạp hoặc thay đổi dữ liệu mà không cần phải nạp lại (reload) trang. Mục đích chính của React là thực thi nhanh, có thể mở rộng, và đơn giản. React xem các thành phần trên giao diện đều là component. Để xây dựng UI, các lập trình viên sẽ tập trung tạo ra các components và có thể sử dụng lại chúng, nhờ vậy mà tiết kiệm được thời gian và công sức.

Cùng với sự đóng góp bởi cộng đồng lập trình viên trên toàn thế giới, React trở thành một thư viện mạnh mẽ trong lập trình UI. Chính vì vậy nó đã được sử dụng bởi rất nhiều các nhà phát triển và các công ty lớn như Netflix, Airbnb, Instagram và New York Times.

Tương tự Angular, React có thể lập trình bằng ngôn ngữ JS hoặc TS.

2.3.2.3 *VueJS*

Tương tự React hay Angular, VueJS là một front-end framework hỗ trợ xây dựng UI và SPA một cách dễ dàng. Nó được phát triển bởi kỹ sư Evan You sau khi đã làm việc với Angular trong một vài dự án tại Google. Anh đã nhìn thấy được những ưu điểm tuyệt vời của Angular nhưng lại chỉ muốn sử dụng lại một phần. Từ đó anh ta đã tạo ra VueJS với tiêu chí gọn nhẹ.

Phiên bản đầu tiên được ra đời năm 2014, được phát hành dưới dạng phần mềm mã nguồn mở sử dụng giấy phép MIT. Từ khi ra đời đến nay nó không ngừng

được cộng đồng phát triển và đã trở thành một trong số những framework được sử dụng phổ biến nhất hiện nay.

Kể từ phiên bản thứ 2, VueJS đã được viết lại với ngôn ngữ TS thay cho JS.

2.3.2.4 *jQuery*

jQuery là một thư viện JS được ra đời từ rất sớm, phiên bản đầu tiên xuất hiện năm 2006. Nó được tạo ra nhằm mục đích đơn giản hóa thao tác trên DOM như duyệt phần tử, thêm, sửa, xóa các phần tử DOM trên trang HTML cũng như xử lý các sự kiện, hiệu ứng và ajax.

Với jQuery, lập trình viên không cần viết các đoạn JS phức tạp để lập trình phía máy khách. Thay vào đó, jQuery cung cấp nhiều phương thức, APIs để có thể sử dụng một cách đơn giản, hiệu quả. jQuery cho phép mở rộng các chức năng bằng cách viết thêm các plugin để gắn vào.

jQuery đã tạo ra phong cách lập trình mới, là nguồn cảm hứng để cho ra đời các framework như YUI v3, Dojo và sau này là tiêu chuẩn Selector API đã được W3C đưa vào tài liệu chuẩn hóa. Hơn thế nữa, nó còn được các nhà khoa học ở Đại học Standford cải tiến với sự kết hợp thuật toán sâu (deep algorithm) với hợp nhất dữ liệu (data fusion) để tạo nên framework D3.js, là framework rất mạnh trong trực quan hóa dữ liệu (data-visualization) được rất nhiều nhà khoa học dữ liệu sử dụng.

Ngày nay nó là một thư viện được sử dụng phổ biến nhất với gần 80% website trên thế giới sử dụng, theo thống kê từ W3Techs năm 2019⁴.

Ngoài các thư viện đã được liệt kê còn rất nhiều các thư viện và framework khác đang hình thành và phát triển và càng trở nên phổ biến, có thể kể đến như Ember.js, Backbone.js, MeteorJS, Preact, Aurelia.

2.3.3 Ngôn ngữ lập trình Back-end

Ngôn ngữ lập trình back-end là ngôn ngữ lập trình phía máy chủ, thực hiện các thao tác như truy vấn dữ liệu, cập nhật dữ liệu hay xử lý thao tác tính toán. Mã lệnh chỉ được thực thi trên server mà không thể thực thi trên trình duyệt như JavaScript.

⁴ https://w3techs.com/technologies/overview/javascript_library

Trước khi có các ngôn ngữ lập trình cho web, người ta đã viết các đoạn mã bằng ngôn ngữ lập trình C, được gọi là CGI (Common Gateway Interface) script. Đây thực chất là các đoạn mã lệnh được hệ điều hành gọi thực thi thông qua web server. Sau này, CGI đã được thay thế bằng ngôn ngữ Perl. Đây có thể được xem là ngôn ngữ lập trình web đầu tiên trên thế giới.

Nhiệm vụ của các ngôn ngữ lập trình là thực hiện các thao tác tính toán và sinh ra kết quả là tài liệu HTML để có thể hiển thị trên trình duyệt.

Do sự phát triển thần tốc của công nghệ web mà ngày nay có rất nhiều ngôn ngữ lập trình phía máy chủ ra đời. Mỗi ngôn ngữ sử dụng kỹ thuật và công nghệ khác nhau, một số ngôn ngữ tiêu biểu như PHP, Java, Ruby, Python, C# và JavaScript.

2.3.3.1 **PHP**

PHP được ra đời năm 1994 và nó đã trở thành ngôn ngữ lập trình web phổ biến nhất thế giới trong một khoảng thời gian dài. Có rất nhiều công ty lớn với hàng trăm triệu khách hàng đã sử dụng PHP để phát triển các hệ thống như Facebook, Wikipedia, Slack, Tumblr, WordPress.

PHP được xem là ngôn ngữ có nhiều framework theo chuẩn MVC (Model-View-Controller) nhất hiện nay như Laravel, Symfony, CodeIgniter, Zend, CakePHP, Yii, Phalcon, FuelPHP, Slim, PHPixie.



Hình 0.6 Các PHP MVC Frameworks

Ngoài ra, còn có rất nhiều các hệ thống mã nguồn mở được phát triển bằng PHP như CMS (Content Management System) platforms: WordPress, Drupal,

Joomla!; các hệ thống thương mại điện tử: Magento, OpenCart, OsCommerce, ZenCart, WooCommerce; hệ thống học trực tuyến (e-learning) Moodle; công cụ quản trị cơ sở dữ liệu phpMyAdmin, Adminer; các diễn đàn (forum) phpBB, vBulletin (đã thương mại hóa), MyBB, Simple Machine Forum.

2.3.3.2 **Java**

Java được biết đến như là một ngôn ngữ lập trình đa nền tảng, được tạo ra từ đầu những năm 1990 để xây dựng các phần mềm hoạt động trên những thiết bị có bộ nhớ giới hạn. Ngày nay ứng dụng Java có mặt ở khắp mọi nơi, bao gồm cả các ứng dụng web.

Các công nghệ Java dành cho web bao gồm Java Servlet API, JavaServer Page (JSP), JavaServer Faces (JSF) và Java Message Service API. Trong đó JSP là công nghệ được sử dụng phổ biến để tạo ra các website, nó tương tự ngôn ngữ lập trình PHP hay ASP.

Khi mô hình MVC ra đời và nó dần trở thành mô hình chuẩn cho việc phát triển ứng dụng web, các framework ra đời theo chuẩn MVC ngày càng nhiều. Tương tự PHP, Java cũng có nhiều MVC framework đang được sử dụng rất nhiều bao gồm Spring MVC, Spring Boot (APIs), Apache Struts, Hibernate, JSF (JavaServer Faces), ActFramework và một số framework không theo mô hình MVC như Apache Wicket, GWT (Google Web Toolkit), Vaadin Vert.X, Play!, Apache Spark và Grails.



Hình 0.7 Các Java Frameworks

2.3.3.3 **ASP.NET**

Là ngôn ngữ lập trình web được tạo ra bởi Microsoft từ cuối những năm 1990. Ban đầu nó có tên là ASP (Active Server Page), đến năm 2002 nó được thay thế

bởi ASP.NET khi Microsoft cho ra đời .NET Framework hỗ trợ nền tảng lập trình tốt hơn.

ASP.NET hỗ trợ công nghệ, các framework trong lập trình web như Entity Framework, ASP.NET Web Forms, ASP.NET MVC, ASP.NET Web API, ASP.NET WebHooks và ASP.NET Core MVC.

ASP.NET Core là phiên bản kế nhiệm ASP.NET. Không giống như các phiên bản trước chỉ có thể chạy trên Windows, nó là web framework được phát hành dưới dạng phần mềm mã nguồn mở hoàn toàn miễn phí, có thể hoạt động trên Windows, Mac và Linux. Framework này đã được viết lại hoàn toàn mới với mục tiêu cốt lõi là hỗ trợ đa nền tảng (cross-platform). Những nền tảng trước đây bao gồm ASP.NET MVC và ASP.NET Web API được tích hợp vào ASP.NET Core.



ASP.NET Core

Hình 0.8 Logo ASP.NET Core

Các nhà phát triển phần mềm có thể sử dụng ngôn ngữ lập trình C# hoặc VB.NET để xây dựng ứng dụng ASP.NET.

2.3.3.4 *Python*

Python được biết đến như là ngôn ngữ lập trình thông dịch (interpreter) cấp cao đa mục đích. Nó được tạo ra bởi lập trình viên người Hà Lan Guido van Rossum năm 1991, người đã từng làm việc tại các công ty công nghệ lớn như Google, Dropbox và Microsoft.

Python hỗ trợ nhiều kiểu lập trình khác nhau: Lập trình hướng đối tượng (OOP - Object-oriented programming), lập trình cấu trúc (Structured programming), lập trình chức năng (Functional programming) và lập trình hướng khía cạnh (AOP – Aspect-oriented programming).

Python được đánh giá là ngôn ngữ rất mạnh, cấu trúc gần với ngôn ngữ tự nhiên, ngắn gọn và dễ học. Ban đầu nó được thiết kế để chạy trên hệ thống Unix, sau này nó đã được sử dụng nhiều bởi cộng đồng các nhà phát triển do những ưu điểm vượt bậc so với các ngôn ngữ lập trình khác, do vậy tác giả đã mở rộng sang các dòng hệ điều hành như Windows, Linux và MacOS.

Ngày nay Python được sử dụng rất nhiều bởi các nhà khoa học để lập trình các thuật toán trong nhiều lĩnh vực như trí tuệ nhân tạo (AI – Artificial Intelligence), máy học (Machine learning), khoa học dữ liệu (Data Science). Tuy nhiên, vì nó là ngôn ngữ đa mục đích, do vậy nó cũng được sử dụng để phát triển các ứng dụng trong các lĩnh vực khác như game, ứng dụng desktop, ứng dụng thương mại (ERP - Enterpise resource planning, CRM – Customer Relationship Management) và ứng dụng web.

Django và Flash là hai framework lập trình web trong Python phổ biến nhất hiện nay. Nó hỗ trợ đầy đủ các tính năng bao gồm MVC (Model-View-Controller), MTV (Model-Template-View), ORM (Object-Relational Mapping), RESTful APIs, GraphQL API, các hệ quản trị cơ sở dữ liệu quan hệ (MySQL, PostgreSQL, Oracle, MS SQL Server) và cơ sở dữ liệu phi quan hệ NoSQL (MongoDB).



Hình 0.9 Django và Flash Frameworks

Ngoài Flash và Django còn có các framework khác như Pyramid, TurboGears, Web2py, Bottle, Tornado, CherryPy, Falcon và FastAPI.

2.3.3.5 *JavaScript / TypeScript*

Mục đích ban đầu của Brendan Eich tạo ra JavaScript là để thực thi một số tác vụ giới hạn trên trình duyệt Netscape Navigator, nhằm phá vỡ giới hạn của web lúc bấy giờ chỉ có nội dung tĩnh.

Năm 2009, Ryan Dahl, một kỹ sư người Mỹ đã tạo ra Node.js, là một môi trường thực thi JavaScript phía server sử dụng JavaScript Engine V8 của Google. Node.js cho phép các nhà lập trình viết mã lệnh JavaScript và thực thi nó bên ngoài trình duyệt. Nhờ đó nó có thể được dùng để tạo các CLI (Command line interface) và được sử dụng như ngôn ngữ lập trình máy chủ.

Năm 2010, một công cụ quản lý các gói (package) có tên npm được giới thiệu giúp quản lý các gói phụ thuộc (package dependencies) trong dự án Node.js. Nó tương tự như các công cụ composer (PHP), NuGet (.NET), pip (python) hay maven (Java).

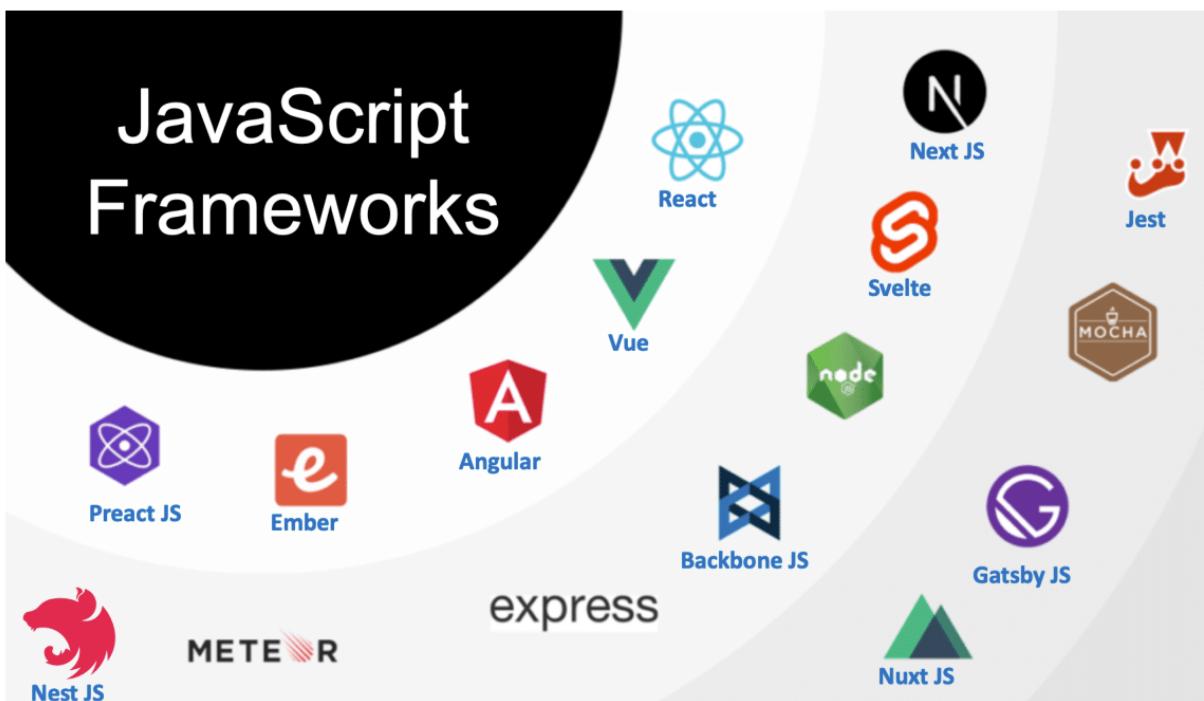
Công cụ npm đã giúp cho các nhà phát triển dễ dàng xuất bản và chia sẻ các mã nguồn của Node.js cũng như làm đơn giản hóa việc cài đặt, nâng cấp và gỡ bỏ các gói trong dự án. Chính vì vậy nó đã thúc đẩy quá trình phát triển của Node.js diễn ra rất nhanh, rất nhiều các phần mềm, các framework ra đời dựa trên nền tảng Node.js.

Node.js hỗ trợ ngôn ngữ JavaScript, một số ngôn ngữ khác như CoffeScript, Dart, TypeScript, ClojureScript cũng được hỗ trợ do nó có thể complie sang JavaScript.

Ngày nay JavaScript dần được thay thế bởi TypeScript do nó không hỗ trợ định kiểu mạnh, và không hỗ trợ lập trình hướng đối tượng. Do đó các framework về sau được viết bằng ngôn ngữ TypeScript thay vì JavaScript.

Node.js là phần mềm nguồn mở, hỗ trợ đa nền tảng. Nó đang được phát triển bởi hơn 3000 nhà lập trình trên khắp thế giới. Các nhà phát triển đã xây dựng các framework nhằm đẩy nhanh sự phát triển của các ứng dụng. Có thể kể đến Express.js, Next.js, Gatsby.js, Nest.js, Nuxt.js, Socket.IO, Koa.js, Hapi.js, Sails.js và Meteor.

JavaScript Frameworks



Hình 0.10 Các Front-end và Back-end Java Frameworks

2.3.3.6 *Các ngôn ngữ khác*

Ngoài các ngôn ngữ lập trình back-end đã liệt kê, còn một số các ngôn ngữ khác cũng được sử dụng khá phổ biến để lập trình phía server như:

Golang

Là ngôn ngữ được thiết kế bởi các kỹ sư của Google từ năm 2009. Golang⁵ đôi khi còn được gọi ngắn gọn là Go, là một ngôn ngữ lập trình biên dịch (compiled language) có cú pháp tương tự như ngôn ngữ lập trình C. Nó được sử dụng để phát triển các phần mềm như Docker, Dropbox hay Couchbase và các dịch vụ như Netflix, Uber, Youtube.

Rust

Rust được phát triển bởi kỹ sư làm việc tại Mozilla năm 2010. Nó được thiết kế nhằm mục đích tối ưu hóa hiệu năng, có cú pháp tương tự C++. Năm 2016 nó được bình chọn là ngôn ngữ được yêu thích nhiều nhất trên trang Stack Overflow.

⁵ <https://golang.org>

Đầu năm 2021, tổ chức Rust Foundation⁶ được thành lập bởi các công ty Huawei, AWS (Amazon Web Services), Facebook, Google, Microsoft và Mozilla nhằm thúc đẩy sự phát triển của Rust.

⁶ <https://foundation.rust-lang.org>

TÀI LIỆU THAM KHẢO

- [1] Tim Berners-Lee(1990). *Information Management: A proposal*. CERN. Retrieved from <http://cds.cern.ch/record/369245/files/dd-89-001.pdf>
- [2] Tim Berners-Lee, Robert Cailliau. WorldWideWeb: Proposal for a HyperText Project. CERN. Retrieved from http://cds.cern.ch/record/2639699/files/Proposal_Nov-1990.pdf
- [3] Tim Berners-Lee: *The WorldWideWeb browser*. Retrieved from <https://www.w3.org/People/Berners-Lee/WorldWideWeb.html>
- [4] J. Gettys, J. Mogul, H. Frystyk, T. Berners-Lee (1997). *Hypertext Transfer Protocol – HTTP/1.1*. Retrieved from <https://datatracker.ietf.org/doc/html/rfc2068>
- [5] Pavlik & McIntosh, John & Shawn (2015). *Converging Media Fourth Edition*. New York, New York: Oxford University Press. p. 168. ISBN 978-0-19-934230-3.
- [6]

