**TÌM HIỂU VỀ LỊCH SỬ ỨNG DỤNG WEB**

**Nhóm 1**

*Các thành viên:*

*Đào Tuấn Anh*

*Đặng Văn Đô*

*Hoàng Xuân Phương*

*Nguyễn Thanh Việt*

*Bế Văn Khanh*

*Lê Hồng Thăng*

MỤC LỤC

[1 GIỚI THIỆU 3](#_Toc441613531)

[2 QUÁ TRÌNH HÌNH THÀNH VÀ PHÁT TRIỂN 4](#_Toc441613532)

[2.1 Cơ sở hình thành 4](#_Toc441613533)

[2.2 Thế hệ đầu tiên của ứng dụng Web 4](#_Toc441613534)

[2.3 AJAX 5](#_Toc441613535)

[2.4 Thời đại không nhất quán về công nghệ (1999 - 2005) 7](#_Toc441613537)

[2.4.1 Server-side được mở ra 7](#_Toc441613538)

[2.4.2 Lựa chọn thay thế Front-end 8](#_Toc441613539)

[2.4.3 RIAs 10](#_Toc441613540)

[2.5 Kỉ nguyên của Html5 & JQuery (2005 - 2009) 11](#_Toc441613541)

[2.5.1 HTML5 11](#_Toc441613542)

[2.5.2 JQuery 12](#_Toc441613543)

[2.6 Kỉ nguyên của các JS Framework (2010 đến hiện tại) 12](#_Toc441613544)

[2.7 Tương lai của ứng dụng Web 13](#_Toc441613545)

[2.7.1 Trình duyệt sẽ như hệ điều hành 13](#_Toc441613546)

[2.7.2 Các ứng dụng di dộng 14](#_Toc441613547)

[2.7.3 Web trên các thiết bị mở rộng 16](#_Toc441613548)

[2.7.4 Cá nhân hóa 17](#_Toc441613549)

[3 TÀI LIỆU THAM KHẢO 18](#_Toc441613550)

# **GIỚI THIỆU**



Một ứng dụng Web hay Web Application là một trình ứng dụng phần mềm có thể tiếp cận qua Web thông qua mạng Internet. Thường thường những phần mềm này được lập trình để thực thi một số lượng lớn các nhiệm vụ như các trình xử lí văn bản hay các phần mềm kế toán. Trong thế giới mạng ngày nay, nơi mà các trình duyệt web và điện thoại thông minh phát triển mạnh mẽ, những ứng dụng này thường được lập trình nhỏ gọn hơn, tập trung vào những nhiệm vụ riêng lẻ. Những ứng dụng web nói riêng, chúng thực thi nhiệm vụ mà mình được lập trình sẵn ngay trên các trình duyệt web và thường có giao diện phong phú, giúp người dùng dễ dàng tương tác và sử dụng.

Google Maps là ví dụ của một ứng dụng web. Nó tập trung thực thi một nhiệm vụ duy nhất: cung cấp những thông tin cụ thể về đường đi cho người dùng ngay trong trình duyệt web. Bạn có thể định vị và thay đổi kích cỡ của một bản đồ nhất định, tìm kiếm một siêu thị hay quán cà phê, hoặc có thể tìm đường để lái xe đến một địa điểm nào đó. Tất cả thông tin mà bạn cần được lập trình và lưu trữ trên ứng dụng web một cách linh động, để phục vụ nhu cầu của bạn mỗi khi bạn cần tới.

# **QUÁ TRÌNH HÌNH THÀNH VÀ PHÁT TRIỂN**

## Cơ sở hình thành

Ban đầu, các trang Web là tĩnh; người dùng gửi yêu cầu một tài nguyên nào đó, và server sẽ trả về tài nguyên đố. Các trang Web không có gì hơn là một văn bản được định dạng và phân tán. Đối với các trình duyệt thì các trang Web tĩnh không phải là vấn đề khó khan, và trang Web lúc đầu chỉ để thông tin về các sự kiện, địa chỉ, lịch làm việc qua Internet mà thôi, chưa có sự tương tác qua các trang Web. Năm 1990, Tim Berners-Lee, tại CERN, đã sáng chế ra HTML (HyperText Markup Language), ngôn ngữ đánh dấu siêu văn bản. HTML rất đơn giản và dễ dùng, nó đã trở thành một ngôn ngữ phổ biến và rất cơ bản. Tuy nhiên không lâu sau đó, nhu cầu về các trang Web động, có sự tương tác ngày một tăng , chính vì thế sự ra đời của các ứng dụng Web là một điều tất yếu.

## Thế hệ đầu tiên của ứng dụng Web

Thế hệ đầu tiên của các ứng dụng được phát triển cho Web rất rất đơn giản. Những ứng dụng này thông thường gồm có một form nhỏ để người dùng có thể điền vào và submit đến server. Các dữ liệu đi đến server, nó được xử lý và một page mới của web sẽ ngày lập tức được phản hồi về phía người dùng. Tất cả xử lý trên server hoàn thành, và các trình duyệt chỉ có thể hiển thị tài liệu. Giải pháp này có tên là Common Gateway Interface (CGI). Tất nhiên, nó rất mất thời gian, nhất là kể từ khi không có một thứ nào giống như broadband

Nếu bạn nhập sai dữ liệu, nó vẫn sẽ gửi đến server, server kiểm tra, tìm các lỗi và trả về một tin nhắn không hoàn thành. Thường thì các form sẽ trả về trạng thái ban đầu – tất cả các vùng đều bị bỏ trống – và bạn sẽ phải nhập tất cả mọi thứ lại từ đầu. Không phải tất cả ứng dụng đều kiểm tra cho lỗi, thế cho nên thường thì các thông tin sai sẽ được lưu trữ và được xử lý muộn hơn.

CGI script có thể được viết bằng một số ngôn ngữ từ Perl cho đến Visual Basic. Tuy nhiên CGI không phải là các an toàn cho các ứng dụng Web. Với CGI, người khác có thể chạy chương trình trên hệ thống. Vì thế, có thể chạy các chương trình không mong muốn gây tổn hại hệ thống. Nhưng dù vậy cho đến hôm nay thì CGI vẫn được sử dụng.

## AJAX

### Trước khi tìm hiểu tại sao Ajax lại được xem là "cứu tinh" của các ứng dụng Web, hãy thử phân tích những giới hạn của các ứng dụng web hiện tại khiến nó chưa thể thay thế cho các phần mềm phía client truyền thống.

Quay lại thời điểm, khi mà các dịch vụ web bùng nổ, người ta đã nghĩ đến một lúc nào đó tất cả các ứng dụng mà ta sử dụng sẽ là các ứng dụng Web thay vì các phần mềm chạy độc lập trên các máy tính đơn lẻ. Quả thật, với sự phát triển chóng mặt của mạng Internet cùng với những ưu điểm của các ứng dụng Web (truy cập tại mọi nơi, không cần nâng cấp,…), tương lai của các phần mềm chắc chắn sẽ gắn chặt với các ứng dụng Web, nếu không muốn nói là có thể sẽ bị thay thế. Tuy nhiên, cho đến giờ, giấc mơ đó vẫn chưa thành sự thật và người ta bắt đầu nghĩ rằng, có lẽ nó sẽ không bao giờ trở thành sự thật.

Tại sao vậy? Bởi vì một trong những giới hạn quan trọng của các ứng dụng Web hiện tại là cách thức nó tương tác với người dùng. Khác với các phần mềm chạy độc lập ở máy khách có những khả năng dường như vô tận trong cách thức tương tác với người dùng, các ứng dụng Web bị giới hạn bởi chính nguyên lý hoạt động của nó: tất cả các giao dịch phải thực hiện thông qua phương thức giao dịch HTTP (HyperText Transport Protocol - Giao thức truyền tải qua các siêu liên kết).

Để hiểu tại sao tính chất này lại trở thành một rào cản của các ứng dụng web, hãy phân tích cách thức hoạt động của các dịch vụ web hiện tại xử lý một tác vụ đơn giản như xóa email trong YahooMail. Ta đang duyệt qua hòm thư “Inbox” của Yahoo!Mail. Khi chọn một số email và nhấn nút Delete để xóa chúng (chuyển vào thùng rác). Yahoo!Mail trước hết sẽ lấy danh sách các email được chọn (quá trình này chạy trên máy local), sau đó gửi danh sách này cùng với mã lệnh qua một siêu liên kết đến server của Yahoo, yêu cầu server thực hiện tác vụ xóa đối với các email đó và gửi lại trang web Yahoo!Mail với nội dung mới, rồi cập nhật để trình duyệt hiển thị. Việc gửi nhận yêu cầu này mất một khoảng thời gian trễ, nếu ta sử dụng ADSL thì thời gian này cũng không quá lâu, còn nếu dùng dịch vụ dial-up thì thời gian chờ đợi là rất lớn. Ta cũng sẽ phải trải qua một quá trình tương tự đối với các tác vụ khác, ví dụ như chuyển từ thư mục “Inbox”(hòm thư đến) sang “Sent” (hòm thư đi).

Ta sẽ không bao giờ phải trải qua việc chờ đợi trên khi sử dụng các phần mềm chạy trên máy tính đơn lẻ: không bao giờ thấy phần mềm một khi đã được mở ra lại phải “vô hiệu” trong vài giây để cập nhật dù chỉ là một tác vụ đơn giản nhất, và ngay cả khi phần mềm cần thời gian xử lý một tác vụ nào đó thì ta vẫn thấy nó vẫn tương tác với người dùng. Nếu xét về khía cạnh khả năng ứng dụng trong các tác vụ hàng ngày thì hạn chế trên của các ứng dụng web là không thể chấp nhận được.

Tất nhiên, bên cạnh rào cản về cách thức tương tác, các ứng dụng Web còn vấp phải nhiều giới hạn khác (ví dụ như bản thân việc phải hoạt động dựa trên các trình duyệt đã là một rào cản quan trọng) nhưng một khi chưa giải quyết được vấn đề trên thì các ứng dụng web sẽ không bao giờ có thể thay thể cho các phần mềm độc lập.

Năm 2005, Google đã đưa ra 2 dich vụ được làm dựa trên viêc sử dụng các yêu cầu không đồng bộ để gửi XML fragments giữa trình duyệt và server. Đó chính là khởi nguồn cho cái tên Ajax. Ajax – Asynchronous JavaScript and XML programming ra đời là một giải pháp cho các ứng dụng Web hiện nay, và như ta nói, nó là một trong số các công nghệ Web thế hệ thứ hai. Hai dịch vụ đã được nhắc đến đó là Google Maps và Gmail

* **Google Maps**

Trước đây, sơ đồ thường là tĩnh: người xem bấm vào mũi tên trái, đợi vài giây để trang web tải nội dung và dịch chuyển hình ảnh sang trái. Quá trình cứ lặp đi lặp lại như thế một cách chậm chạp và gây khó chịu cho nhiều người. Với Google Maps, các sơ đồ được chia nhỏ thành các mảnh, các mảnh này được request và receive động bởi page nằm bên trong trình duyệt. Các mảnh này load nhanh hơn rất nhiều so với việc load toàn bộ một bức ảnh lớn và còn cho phép người dùng có thể dễ dàng phóng to và thu nhỏ vị trí đang xem. So với những gì trước đó, đây quả là một bước tiến vượt bậc của các ứng dụng Web.

* **Gmail**

Gmail, một dịch vụ email miễn phí, đơn giản để sử dụng, và là một trong những ứng dụng Web tiêu biểu trong sử dụng Ajax. Gmail được thực thi giống như một single webpage và thay đổi khi nhận yêu cầu của người dùng. Các Emails xuất hiện thanh một danh sách và khi ta ấn vào một email, page sẽ đổi sang hiện ra toàn bộ nội dùng của tin nhắn. Các lần đổi trang yêu cầu lượng data nhỏ hơn so với việc tải lại toàn bộ trang. Không chỉ có vậy, nó còn tránh được “blinking”, xảy ra khi thay thế page này bằng page kia trong khi cả hai có cùng nội dung.

## Thời đại không nhất quán về công nghệ (1999 - 2005)

Sau những ấn tượng của Google maps và Gmail để lại, các developer bắt đầu tiếp cận Ajax và đưa nó vào sử dụng ở rất nhiều các website khác và các ứng dụng online. Chỉ với một lượng nhỏ code Ajax được sử dụng một cách khéo léo, có thể cải thiện rất nhiều cho các ứng dụng đã tồn tại.

Ajax ngày nay càng được sử dụng rộng rãi hơn, nó được tích hợp ở rất nhiều hệ thống lớn như Twitter, Facebook và LinkedIn. Kèm theo đó là ngày càng nhiều hơn các công cụ chỉnh sửa HTML đưa ra các đoạn code Ajax có sẵn nhằm giảm thiểu công việc của người lập trình.

Google đã làm Google Documents dựa trên công nghệ Ajax, giống với Zoho, một công ty nhỏ hơn với một portfolio ấn tượng của ứng dụng Web dựa trên Ajax. Những ví dụ này đã cho ta thấy sự phát triển của ứng dụng Web mới kèm theo đó là những yêu cầu ngày càng cao và khó khan từ phía người dùng

### Server-side được mở ra

Trong khi tất cả những thay đổi đều được hiện rõ qua các hiệu ứng trên trình duyệt của các ứng dụng web, phát triển trên server-side cũng không chịu dậm chân tại chỗ. Nó thực sự rất cần thiết để cung cấp thông tin tới trình duyệt của ứng dụng, và không đáp ứng toàn bộ các HTML page, giúp cho việc thay đổi Web server sang Application Server. Để hỗ trợ cho vấn đề này, các tiêu chuẩn mới cho các dịch vụ đã được thiết lập ở trong form của Web Services.

Với các Web Service được sử dụng rộng rãi và hữu dụng, người ta đặt ra câu hỏi là làm thế nào để có thể mô hình hóa và quản lý những môi trường mới này và từ đó Service Oriented Architecture (SOA) ra đời. Đây là thứ được sử dụng rộng rãi để mô tả các hệ thống nên được xây dựng và sử dụng Web ( và có thể là non-Web) Services như thế nào.

### Lựa chọn thay thế Front-end

Số lượng của người sử dụng Internet ngày một tăng lên, và sớm có những cách tiếp cận khác nhau để cung cấp trải nghiệm nhiều hơn và phong phú hơn, cái mà người dùng muốn. Ba trong số các cách tiếp cận đó đã tự tách mình thông qua việc được sử dụng rộng rãi.

#### Adobe Flex and Flash

Thực tế là người dùng muốn các ứng dụng đáp ứng nhiều hơn đã không được chú ý. Adobe, những người đã có trong danh mục đầu tư của mình đó là Flash plug-in cho các trình duyệt, quyết định rằng nó có thể làm tăng phản ứng bằng cách thay thế công cụ rendering trong trình duyệt riêng của mình, giảm vai trò của trình duyệt để nó chỉ đơn giản là bệ phóng của các ứng dụng. Trong nhiều năm, các Flash plug-in đã được sử dụng rộng rãi để tạo ra các hình ảnh động và biểu đồ. Nó cũng có thể được sử dụng để tạo ra các game nhỏ được chơi trên Web mà không cần phải cài đẳ bất cứ thứ gì trên máy tính cá nhân.

Adobe làm cho Flash trở thành một nền tảng tốt hơn cho việc tạo ra các ứng dụng Internet tương tác và đáp ứng. Không chỉ có vậy, họ còn tạo ra một tập hợp toàn bộ các phần mở rộng Flash và cuối cùng làm mới toàn bộ engine phía sau của plug-in, mở đầu cho Flex – một nền tảng công nghệ, công cụ phát triển phần mềm RIA (Rich Internet application).

Flash và Flex cho phép tạo ra các ứng dụng rất hấp dẫn nhưng có thể bị coi là “foreign” khi trải nghiệm Web trừ khi chúng thực hiện một vai trò đặc biệt ở một HTML page bình thường, các video trên Youtube là một ví dụ, chúng được trình bày với Flash, nhưng những trang đó vẫn được tạo ra bằng cách sử dụng các tiêu chuẩn Web cơ bản như HTML và CSS. Nếu toàn trang Youtube được tạo ra bằng Flash, nó có thể sẽ đẹp hơn nhưng sẽ theo cách tinh tế “khác biệt” cho người dùng cuối cùng. Ngoài ra, tiêu đề phim, tác giả, comments và các thông tin khác được hiển thị trên trang sẽ không thể truy cập bởi các công cụ tìm kiếm.

Flash plug-in cũng được tự động bị chặn bởi tường lửa ở nhiều doanh nghiệp và trường học bởi vì sự phong phú của các Flash online game mà hiện nay đang có sẵn.

Plug-in cũng là một vấn đề đối với các trang Web có khả năng được truy cập thông qua thiết bị di động. Iphone của Apple là một ví dụ, đó là một chiếc điện thoại được người ta sử dụng để truy cập vào Web trên di động nhiều nhất nhưng lại không hỗ trợ các plug-in.

Gần đây Adobe đã tạo ra công nghệ mới có tên là Air, một runtime hỗ trợ việc thực hiện của Flash và các ứng dụng dựa trên Flex bên ngoài của các trình duyệt.

Mặc dù có những tiến bộ, nhưng dường như Flash không được áp dụng rộng rãi ngoài những vai trò truyền thống của nó như cung cấp đồ họa và các hình ảnh động trong một trang Web. Trong thực tế, tuy rằng số lượng người sử dụng các thiết bị di động ngày càng tăng, nhưng Flash vẫn được các công ty khuyến khích áp dụng hay thậm chí là duy trì nó.

#### Microsoft Silverlight

Nếu như Flash của Adobe tập trung vào đồ họa và hình ảnh động để trở thành một nền tảng cho các ứng dụng, Microft đã phản ứng để đối phó ngăn chặn việc thống trị tổng thể trong lĩnh vực này của Adobe. Đó chính là lý do vì sao họ cho ra đời Silverlight, thứ mà tạo ra nội dung tương tác giống như Flash và Flex, nhung nó cho phép các ứng dụng hoạt động bên ngoài các trình duyệt giống như Air. Rất đáng tiếc là Silverlight lại không đạt được thành công như mong đợi của Microsoft. Silverlight được sử dụng trên một số trang web cho video streaming, nhưng việc sử dụng này lại tương tự với các công nghệ của Adobe, ngoài ra nó còn bắt người dùng phải cài đặt rồi mới có thể sử dụng.

#### Ajax và các chuẩn về Web

Ajax và các công nghệ đại diện cho các tiêu chuẩn Web đang dùng như các công nghệ được lựa chọn để xây dựng các trang Web có tính tương tác và đáp ứng cao hơn. Những công nghệ này được thực hiện trực tiếp bởi các trình duyệt mà không cần phải cài đặt bất cứ một thứ gì. Điều này tạo ra sự khác biệt rất lớn trong khả năng tiếp cận một trang Web. Bất cứ ai có quyền truy cập Web có thể tương tác với một ứng dụng Web được xây dựng bằng cách sử dụng các kỹ thuật Ajax.

### RIAs

Các ứng dụng sử dụng Internet như một phương tiện để giao tiếp và phân phối được gọi là Rich Internet Applications hay RIA. Thuật ngữ này bao gồm các ứng dụng được xây dựng với công nghệ không dựa trên tiêu chuẩn như Flash, Flex, Air hay Silverlight và các ví dụ khác như Skype hay phần mềm Messaging.

Các ứng dụng này rất quan trọng trong cuộc sống hằng ngày của chúng ta và chắc chắn xứng đáng được công nhận, đánh dấu một kỉ nguyên mới trong việc sử dụng máy tính. Chúng ngay lập tức được trang bị sẵn ở bất kì trình duyệt hiện đại nào và rất dễ sử dụng.

Các công nghệ mà ta chọn cuối cùng phụ thuộc vào mục đích của ta. Ta muốn trang web có thể được truy cập rộng rãi, được hiển thị ở kết quả của các trang Web tìm kiếm thì có một số điều cần phải xem xét.

#### Độc lập về thời gian chạy

Điểm yếu lớn nhất của công nghệ plug-in đó là dựa trên một thời gian chạy cụ thể và không có đảm bảo rằng điều này sẽ có trong máy tính của người dùng cuối. Nhiều doanh nghiệp đã áp dụng kiểm soát rất chặt chẽ về việc cài đặt các plug-in và môi trường runtime khác do lo ngại rằng chúng có thể được gắn một số mã độc để vượt qua các biện pháp bảo mật. Các tổ chúc giáo dục như trường học và đại học thường chặn truy câp vào các plug-in, vì vậy tốt nhất là chỉ nên sử dụng khả năng mặc định của các trình duyệt. Điều này sẽ đảm bảo rằng, ứng dung của bạn sẽ chạy đúng và máy tính cá nhân sẽ được bảo vệ.

#### Môi trường kiểm soát chặt chẽ

Có một số lượng ít các tổ chức đang rất quan tâm đến vấn đề bảo vệ sở hữu trí tuệ điều này đã gây nên việc họ vô hiệu hóa JavaScript ở các trang. Điều này sẽ làm Ajax không thể hoạt động. Nhưng hiện nay Ajax rất phổ biến thế cho nên mức hạn chế này là rất nhỏ.

#### Các nên tảng di động

Hiện nay, hai nền tảng di động Android OS và iOS đang có số lượng người dùng rất lớn, các Smartphone chạy trên hai nền tảng này đều trang bị sẵn 3G, Wifi và mới nhất là 4G. Chính vì thế, lượng người truy cập từ trình duyệt của các nền tảng này cũng không hề nhỏ. Trình duyệt di động trở nên quan trọng hơn trong tương lai, và trong ki Adobe và Microsoft đang cố gắng để nền tảng RIAs họ đang phát triển hỗ trợ cho các thiết bị này.

## Kỉ nguyên của Html5 & JQuery (2005 - 2009)

### HTML5

Năm 2008, phiên bản nháp đầu tiên của HTML 5 được W3C công bố. Tuy rằng chưa được hoàn thiện, nhưng phiên bản thứ 5 này đã chứng tỏ được tầm quan trọng của mình bằng các nhóm thẻ, thuộc tính, api mới,…

Với sự ra đời của HTML 5 đã phần nào cải thiện được trải nghiệm của người dùng khi sử dụng các website hoặc ứng dụng dùng HTML 5. Ngoài ra, nó cũng hỗ trợ cho lập trình viên và nhà phát triển đỡ phải nhức đầu khi phải bị lệ thuộc quá nhiều vào phía thứ 3 hoặc phải mất nhiều thời gian, công sức để tích hợp website, ứng dụng của mình cho nhiều thiết bị khác nhau như Desktop, Smartphone, Tablet, Tivi, Máy giặt…

HTML5 đã mở ra một kỷ nguyên mới, nó đã đưa đến cho người sử dụng:

* Đem lại sự tiện ích
* Khả năng tương thích
* Tính tương tác cao

### JQuery

Thư viện jQuery được John Resig giới thiệu lần đầu tiên vào năm 2006 trong sự kiện BarCamp tại thành phố New York. Tại thời điểm jQuery được giới thiệu lần đầu tiên thì việc lập trình javascript là ác mộng đối với các nhà lập trình. Do vấn đề tương thích javascript đối với các trình duyệt như Internet Explorer 6, Mozilla Firefox là khác nhau và nó luôn ra sự bối rối cho các lập trình viên.

Kể từ đó, thư viện javascript jQuery được sử dụng rộng rãi và theo thống kê có trên 77.8% các website sử dụng thư viện này và jQuery gần như là lựa chọn mặc định của các nhà phát triển website trong việc lập trình javascript.

jQuery có thể:

* Hướng tới các thành phần trong tài liệu HTML
* Thay đổi giao diện của một trang Web
* Tương tác với người dùng
* Tạo hiệu ứng động cho thay đổi của tài liệu
* Lấy thông tin từ server mà không cần tải lại trang web

## Kỉ nguyên của các JS Framework (2010 đến hiện tại)

JQuery luôn là thư viện JavaScript quan trọng mà bất kể lập trình viên web nào cũng phải học để sử dụng để có thể tạo ra những trang web mạnh mẽ đáp ứng trào lưu web 2.0. Nhưng với sự xuất hiện của trào lưu Single Page Web Application (gọi tắt là SPA), jQuery dường như không đủ để giúp lập trình viên xây dựng các ứng dụng SPA một cách dễ dàng. Các framework JavaScript hiện tại giúp lập trình viên có thể xây dựng các ứng dụng dựa trên mô hình MVC (hoặc MVVM, hay MV\*), nhưng đây là ứng dụng được xây dựng trên nền JavaScript và HTML và ở phía trình duyệt thay vì ở phía server như các server-side framework như ASP.NET MVC, RoR hay Spring MVC.   
Dưới đây là danh sách Framework được cho là tiêu biểu nhất:

* AngularJS
* ReactJS
* EmberJS
* MeteorJS
* BackboneJS
* KnockOutJS

Cũng ở giai đoạn này, các trình duyệt đã được cải tiến và được nâng cấp nhằm tránh các rủi ro về bảo mật cho người dùng và để phù hợp với các công nghệ mới.

## Tương lai của ứng dụng Web

### **Trình duyệt sẽ như hệ điều hành**

Nhờ sự phát triển của các [ứng dụng web](http://wampvn.com/), những công cụ đầy sáng tạo trực tuyến, những lựa chọn phong phù về giải trí trên internet, mà giờ đây chúng ta dành nhiều thời gian cho trình duyệt hơn bất kể phần mềm ứng dụng nào. Việc này dẫn đến người dùng mong đợi nhiều hơn  ở trình duyệt, cần thêm độ tin cậy, tốc độ, sự tinh tế cho việc lướt web.

Sự kỳ vọng này cũng đã được đáp lại, đại loại như là các sản phẩm ứng dụng trực tuyến của Goolge: Gmail, Google talk, Google Calendar, Google Docs. Các công cụ chỉnh sửa ảnh trực tuyến như **Picnik, Adobe Photoshop Online**.

Và những kỳ vọng từ phía người dùng sẽ được đáp ứng bằng những sản phẩm online trong tương lai gần. Chẳng hạn như **hệ điều hành Chrome**. Ý tưởng trình duyệt như là 1 hệ điều hành từng là 1 giấc mơ, nhưng giờ đây đã là sự thật.

### Các ứng dụng di dộng

Giờ đây việc lướt web bằng i**phone**, PDA không còn xa lạ gì với người dùng nữa.

Tuy nhiên cũng có rất nhiều website chưa có kế hoạch nào cho việc tạo ra 1 phiên bản rút gọn để chạy trên màn hình của iphone hay di động nhỏ bé thay vì màn hình Desktop.

Khối lượng dữ liệu lớn làm sao có thể đưa lên iphone và chạy như 1 chiếc máy tính để bàn. Điều này làm cho người dùng cảm thấy bất tiện. Nhưng **Facebook, Twitter** và rất nhiều các ứng dụng khác đã làm được điều này, trình bày đẹp mắt tiện dụng, khi dùng ngón tay trên iphone.

Điều này cho thấy tương lai của website là phát triển như các ứng dụng trên PDA nhỏ gọn tương tác cao, tiện dụng.

Vì vậy các nhà thiết kế cần phải nắm rõ kỹ thuật để có thể phát triển 1 phiên bản cho PDA  về mẫu thiết kế lẫn việc xử lý dữ liệu.

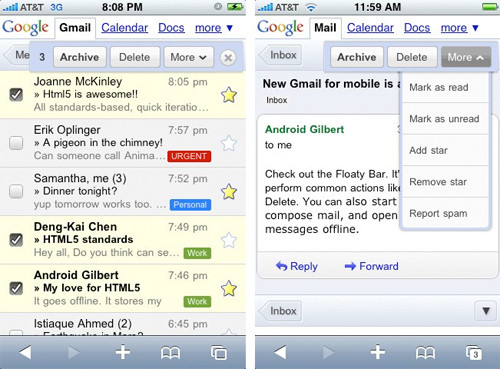
dich vu sms, dich vu SEO, tin nhan sms, top 10 google

1 ví dụ thành công trong việc chuyển đổi website thương mại thành 1 phiên bản nhỏ hơn ( lite Version ) đó là**Amazon**, chỉ giữ lại những tính năng cần thiết, tìm kiếm, giỏ hàng, catalog sản phẩm. Chạy rất tốt trên màn hình iphone. Như vậy người dùng có thể mua cái họ cần ở bất kỳ đâu.

**Facebook**cũng là 1 ví dụ điển hình cho xu hướng tiện dụng này. Và họ không ngừng cải thiện phiên bản thu gọn này. Phiên bản di động mới nhất của họ: gồm có 1 màn hình chủ, chứa các tính năng mà người dùng hay sử dụng nhất. và một bản tổng hợp thông tin. Các dữ liệu được phân đoạn hợp lý, khiến việc truy cập trở nên nhanh chóng. Người dùng có thể cảm nhận được sự quen thuộc như trên bản đầy đủ.



Và cuối cùng là Gmail trên iphone. Có danh sách thư, nội dung mô tả của thư, và các icon nổi bật hiện thị các nhóm thư. Khiến người dùng tiện lợi không cần phải mở từng hộp thư mà cũng có thể biết thư có nội dung gì.



### Web trên các thiết bị mở rộng

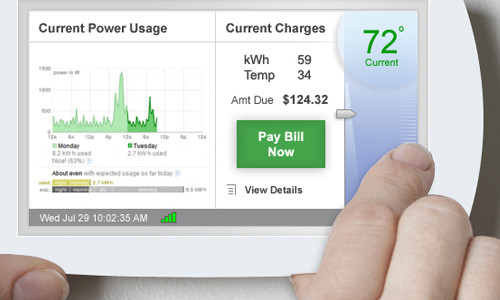
* **Thẻ VIP mua hàng**

Các thẻ mua hàng sẽ được kết nối cơ sở dữ liệu hàng hóa của các cửa hàng chi nhánh, để có thể báo cho người dùng biết khi nào họ sử dụng chúng, và biết được thông tin khuyến mãi, sản phẩm mới.



* **Thiết bị hỗ trợ thông tin.**

1 trong những ví dụ đó là các màn hình hiển thị thông tin năng lượng điện họ sử dụng và còn có thể thanh toán thông qua màn hình này. Bằng việc sử dụng ứng dụng Google PowerMeter.



* **Thẻ tài khoản ngân hàng cá nhân**



Đây là thiết bị hết sức hữu dụng, người dùng có thể xác thực thông tin trên thẻ và xem số dư, số nợ trong tài khoản của mình. Vấn đế lớn ở đây là phải bảo mật an toàn, chứng thực chính xác. Thiết bị này sẽ có mặt trên thị trường trong thời gian không xa.

### Cá nhân hóa

Nhờ sự phát triển nhanh chóng của mạng xã hội, mà người dùng giờ đã rất quen với việc tùy biến các thông tin các nhân của họ, những công cụ khiến cho việc duyệt web của người dùng trở nên phù hợp với thói quen của từng người hơn.

* **Hồ sơ cá nhân tập trung.**

Nếu bạn là 1 người thường xuyên duyệt web, có nhiều tài khoản trên các mạng xã hội, trang e-store ( cửa hàng trực tuyến ) tài khoản mail, thì bạn đã có trung bình từ 2 – 6 bộ hồ sơ thông tin các nhân trên các mạng xã hội website. Có thể bạn dùng 1 tài khoản cho tất cả. Nhưng tại sao phải như vậy, dữ liệu chồng chéo như vậy.

Ngày nay người dùng đã bắt đầu quen với việc chỉ sử dụng 1 tài khoản cho tất cả các ứng dụng trên mạng internet, và họ có thể kiểm soát được thông tin của họ cho mọi người cũng như chỉ cần cập nhật 1 lần cho sự thay đổi của bạn.

* **Quyền sở hữu dữ liệu**

Điều này ngày càng trở nên quan trọng khi mạng xã hội phát triển.  Đặc biệt là những dữ liệu riêng tư, hình ảnh, video, tin nhắn, hồ sơ nhạy cảm. .. Theo thống kê của Royal Pingdom, trong năm 2009 đã có 2500000000 bức ảnh được tải lên facebook. Con số này cho thấy người dùng giờ đây quan tâm việc bảo mật, kiểm soát thông tin của họ hơn bao giờ hết.

Một xu hướng về chia sẻ dữ liệu cá nhân thông qua các mạng xã hội: thay vì ta phải tải lên Flickr bản sao hình ảnh trên máy bạn. thì chỉ cần cho Flickr truy cập đến file này trên máy. Tư duy này có vẻ là 1 bước lùi của công nghệ, nhưng thật chất việc xử lý không hề đơn giản. Nhưng tác dụng nó đem lại thì lại rất lớn. vô cùng tiện dụng và không phải tốn tài nguyên lưu trữ của nhà cung cấp.

# **TÀI LIỆU THAM KHẢO**

* <https://evolution-of-apps.firebaseapp.com/#/>
* <http://wiki.morfik.com/wiki3/Web_Application_Development_-_A_General_Overview>
* <https://www.w3.org/TR/html5/>
* <https://en.wikipedia.org/wiki/Ajax_(programming)>