

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG**

-----o0o-----



IS # 9001:2000

ĐỒ ÁN TỐT NGHIỆP

NGÀNH CÔNG NGHỆ THÔNG TIN

HẢI PHÒNG 2013

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG**

-----o0o-----

XÂY DỰNG ỨNG DỤNG WEB VỚI HTML 5.0

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

NGÀNH: CÔNG NGHỆ THÔNG TIN

HẢI PHÒNG 2013

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG**

-----o0o-----

XÂY DỰNG ỨNG DỤNG WEB VỚI HTML 5.0

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

NGÀNH: CÔNG NGHỆ THÔNG TIN

Sinh viên thực hiện: Đặng Đức Tuyển

Giáo viên hướng dẫn: Th.s Nguyễn Trịnh Đông

Mã sinh viên: 1351010015

HẢI PHÒNG 2013

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

-----oOo-----

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập – Tự do – Hạnh phúc

-----oOo-----

NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP

Sinh viên: Đặng Đức Tuyên

Mã số: 1351010015

Lớp: CT1301

Ngành: Công nghệ thông tin

Tên đề tài:

XÂY DỰNG ỨNG DỤNG WEB VỚI HTML 5.0

NHIỆM VỤ ĐỀ TÀI

1. Nội dung và các yêu cầu cần giải quyết trong nhiệm vụ đề tài tốt nghiệp

a. Nội dung:

- Tìm hiểu công nghệ Web-based, Web 2.0
- Tìm hiểu các công nghệ trong HTML 5.0
- Xây dựng ứng dụng với HTML 5.0

b. Các yêu cầu cần giải quyết

.....

.....

.....

.....

.....

2. Các số liệu cần thiết để thiết kế, tính toán

.....

.....

.....

.....

3. Địa điểm thực tập

.....

.....

.....

.....

.....

.....

CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

Người hướng dẫn thứ nhất:

Họ và tên: Nguyễn Trịnh Đông

Học hàm, học vị: Thạc sĩ

Cơ quan công tác: Khoa Công nghệ Thông tin – Trường Đại Học Dân Lập Hải Phòng

Nội dung hướng dẫn:

.....
.....
.....
.....

Người hướng dẫn thứ 2:

Họ và tên:

Học hàm, học vị:

Cơ quan công tác:

Nội dung hướng dẫn:

.....
.....
.....
.....

Đề tài tốt nghiệp được giao ngày ... tháng ... năm 20...

Yêu cầu phải hoàn thành trước ngày ... tháng ... năm 20...

Đã nhận nhiệm vụ: Đ.T.T.N

Sinh viên

Đã nhận nhiệm vụ: Đ.T.T.N

Cán bộ hướng dẫn Đ.T.T.N

Hải Phòng, ngày tháng năm 20.....

HIỆU TRƯỞNG

GS.TS. NGUYỄN TRẦN HỮU NGHỊ

PHÂN NHẬN XÉT TÓM TẮT CỦA CÁN BỘ HƯỚNG DẪN

1. Tinh thần thái độ của sinh viên trong quá trình làm đề tài tốt nghiệp:

.....
.....
.....
.....
.....
.....
.....
.....
.....

2. Đánh giá chất lượng của đề tài tốt nghiệp (so với nội dung yêu cầu đã đề ra trong nhiệm vụ đề tài tốt nghiệp)

.....
.....
.....
.....
.....

Cho điểm của cán bộ hướng dẫn: (Điểm ghi bằng số và chữ)

.....
.....
.....
.....

Ngày ... tháng năm 20...

Cán bộ hướng dẫn chính

(Ký, ghi rõ họ tên)

**PHẦN NHẬN XÉT ĐÁNH GIÁ CỦA CÁN BỘ CHĂM PHẢN BIỆN
ĐỀ TÀI TỐT NGHIỆP**

1. Đánh giá chất lượng đề tài tốt nghiệp (về các mặt như cơ sở lý luận, thuyết minh chương trình, giá trị thực tế, ...)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

2. Cho điểm của cán bộ phản biện:(Điểm ghi bằng số và chữ)

.....

.....

.....

.....

Ngày ... tháng năm 20...

Cán bộ chăm phản biện

(Ký, ghi rõ họ tên)

LỜI CẢM ƠN

Trước hết em xin bày tỏ tình cảm và lòng biết ơn đối với thầy Nguyễn Trịnh Đông – Khoa Công nghệ Thông tin – Trường Đại học Dân Lập Hải Phòng, người đã dành cho em rất nhiều thời gian quý báu, trực tiếp hướng dẫn tận tình giúp đỡ, chỉ bảo em trong suốt quá trình làm đồ án tốt nghiệp.

Em xin chân thành cảm ơn tất cả các thầy cô giáo trong khoa Công nghệ Thông tin - Trường ĐHDL Hải Phòng, chân thành cảm ơn các thầy giáo, cô giáo tham gia giảng dạy và truyền đạt những kiến thức quý báu trong suốt thời gian em học tập tại trường, đã đọc và phản biện đồ án của em giúp em hiểu rõ hơn các vấn đề mình nghiên cứu, để em có thể hoàn thành đồ án này.

Em xin cảm ơn GS.TS.NGƯT Trần Hữu Nghị Hiệu trưởng Trường Đại học Dân lập Hải Phòng, Ban giám hiệu nhà trường, Bộ môn tin học, các Phòng ban nhà trường đã tạo điều kiện tốt nhất trong suốt thời gian học tập và làm tốt nghiệp.

Tuy có nhiều cố gắng trong quá trình học tập, trong thời gian thực tập cũng như trong quá trình làm đồ án nhưng không thể tránh khỏi những thiếu sót, em rất mong được sự góp ý quý báu của tất cả các thầy giáo, cô giáo cũng như tất cả các bạn để kết quả của em được hoàn thiện hơn.

Em xin chân thành cảm ơn!

Hải Phòng, ngày 24 tháng 06 năm 2013

Sinh viên

Đặng Đức Tuyên

Mở Đầu

Với sự tăng trưởng ngày càng mạnh mẽ của Internet, các thuật ngữ như Web 2.0 và RIA (Rich Internet Application) hầu như hiện diện ở khắp nơi. Người sử dụng máy tính, được trang bị phần cứng nhanh hơn và băng thông mạng tốt hơn, cũng thường xuyên đặt ra những yêu cầu phức tạp. Web hiện đại không phải chỉ là để sử dụng được, mà còn phải bắt mắt và giàu khả năng tương tác.

Trước thực tế đó, rõ ràng HTML cũng cần phải thay đổi, và sự thay đổi đó là cả một quá trình dài! Hơn 13 năm kể từ khi HTML4 xuất hiện, HTML5 mới đang tập tễnh những bước đi đầu tiên! Mặc dù vậy, HTML5 mang trong mình đủ sức hấp dẫn để gây nên nhiều sự chú ý.

HTML5 làm cho việc thiết kế và phát triển web dễ dàng hơn bằng cách tạo một giao diện ngôn ngữ đánh dấu chuẩn hóa và trực quan. HTML5 cung cấp các công cụ quản lý dữ liệu, đồ họa (2D và 3D), phim, và âm thanh có hiệu quả mà không cần phải cài đặt thêm một phần phần của hãng thứ 3 nào. Nó tạo điều kiện cho sự phát triển của các ứng dụng giữa các trình duyệt với nhau cho trang web cũng như cho các thiết bị di động. HTML5 là một trong những công nghệ thúc đẩy những cải tiến trong các dịch vụ điện toán đám mây di động, vì nó tính đến tính linh hoạt rộng hơn, cho phép phát triển các trang web thú vị và có khả năng tương tác. Nó cũng đưa vào thẻ và các cải tiến mới, bao gồm cấu trúc thu nhỏ, các nút điều khiển của biểu mẫu, các API, đa phương tiện, hỗ trợ cơ sở dữ liệu, và tốc độ xử lý nhanh hơn đáng kể. Do vậy nên HTML5 có khả năng xuất bản tất cả mọi thứ trên thế giới từ nội dung văn bản đơn giản đến đa phương tiện phong phú, tương tác cho các nhà thiết kế và các nhà phát triển ở mọi trình độ.

Các thẻ mới, các phương thức mới, và một framework phát triển chung dựa trên sự tác động lẫn nhau của HTML5 và hai đối tác của nó, CSS3 và JavaScript. Đây là cốt lõi của hiện tượng xử lý ứng dụng lấy máy khách làm trung tâm. Ngoài các việc triển khai các kỹ thuật và các phương thức của công nghệ HTML5 cho máy tính để bàn, có thể triển khai thực hiện HTML5 trong nhiều trình duyệt điện thoại di động web có tính năng phong phú - một thị trường đang phát triển, khi đã chứng kiến sự phổ biến của các hệ điều hành web Apple iOS, Google Android, và các điện thoại chạy Palm.

Chính vì những lý do trên, em đã chọn đề tài: “**Xây dựng ứng dụng Web với HTML 5.0**”. Đồ án bao gồm 3 chương:

- **Chương 1:** Tìm hiểu về công nghệ Web-based và Web 2.0:

Trình bày về sơ lược của công nghệ Web-based và những khái niệm và kiến trúc cơ bản của Web 2.0.

- **Chương 2:** Tìm hiểu các công nghệ mới trong HTML 5.0:

Trình bày về sự ra đời và những giai đoạn phát triển của HTML. Tiếp đó là những điểm mới của HTML5 so với các phiên bản trước đây. Và phần quan trọng nhất là những công nghệ tuyệt vời mà HTML5 đang có được.

- **Chương 3:** Xây dựng ứng dụng

Trình bày về sản phẩm sử dụng các công nghệ của HTML5 đã xây dựng được trong thời gian nghiên cứu đề tài.

Cuối cùng là phần kết luận.

MỤC LỤC

Lời cảm ơn	1
Mở đầu	2
Mục lục.....	4
Danh mục các chữ viết tắt và giải nghĩa	6
Danh mục hình	7
Danh mục bảng	9
Chương 1: Tìm hiểu công nghệ Web-based, Web 2.0.....	10
1.1 Giới thiệu Web-based	10
1.2 Giới thiệu Web 2.0	10
1.2.1 Khái niệm.....	10
1.2.2 Web 2.0	11
1.2.3 Kiến trúc cơ bản của Web.....	12
Chương 2: Tìm hiểu các công nghệ trong HTML 5.0	14
2.1 Sự phát triển của HTML	14
2.2 Giới thiệu HTML 5.0	15
2.2.1 HTML5 là gì?.....	15
2.2.2 Những điểm mới trong HTML5	17
2.3 Những công nghệ mới trong HTML5	20
2.3.1 Canvas API.....	20
2.3.2 Scalable Vector Graphics.....	25
2.3.3 WebGL – 3D trên Web	27
2.3.4 Audio và Video	30
2.3.5 Geolocation API.....	35
2.3.6 Communication APIs.....	38
2.3.7 WebSocket API.....	41
2.3.8 Forms API	43
2.3.9 Drag-and-Drop	50

2.3.10 Web Workers API.....	53
2.3.11 Storage APIs.....	55
2.3.12 Tạo Offline Web Applications	57
Chương 3: Xây dựng ứng dụng.....	60
3.1 Giới thiệu ứng dụng	60
3.2 Yêu cầu phần cứng.....	60
3.3 Yêu cầu phần mềm.....	60
3.4 Một số giao diện chính.....	60
3.4.1 Website chia sẻ video và Xác định vị trí.....	61
3.4.2 Một số ứng dụng khác.....	63
Kết luận	64
Tài liệu tham khảo.....	65

DANH MỤC CÁC CHỮ VIẾT TẮT VÀ GIẢI NGHĨA

Từ viết tắt	Giải nghĩa
2D	Two dimension – Hai chiều
3D	Three dimension – Ba chiều
AJAX	Asynchronous JavaScript and XML - JavaScript và XML không đồng bộ
API	Application Programming Interface – Giao diện lập trình ứng dụng
CSS	Cascading Style Sheets - Miêu tả cách trình bày các tài liệu viết bằng ngôn ngữ HTML và XHTML
DOM	Document Object Model - Mô hình Đối tượng Tài liệu, là một giao diện lập trình ứng dụng (API)
ECMAScript	Là phiên bản chuẩn hóa của JavaScript.
HTTP	HyperText Transfer Protocol - Giao thức truyền tải siêu văn bản
HTTPS	Secure HTTP - Một sự kết hợp giữa giao thức HTTP và giao thức bảo mật SSL hay TLS cho phép trao đổi thông tin một cách bảo mật trên Internet.
IETF	The Internet Engineering Task Force - Lực lượng đặc nhiệm kỹ thuật Internet. Tổ chức IETF phát triển và xúc tiến các tiêu chuẩn Internet, có quan hệ hợp tác gần gũi với các tổ chức tiêu chuẩn W3C và ISO/IEC
JS	JavaScript - một ngôn ngữ lập trình kịch bản dựa trên đối tượng được phát triển từ các ý niệm nguyên mẫu.
JSON	JavaScript Object Notation - Là một kiểu dữ liệu trong JavaScript
RDF	Resource Description Framework - Framework Mô Tả Tài Nguyên
REST	Representational State Transfer - Dạng yêu cầu dịch vụ web mà máy khách truyền đi trạng thái của tất cả giao dịch
SGML	Standard Generalized Markup Language – Một chuẩn ISO, là một hệ thống tổ chức và gắn thẻ yếu tố của một tài liệu
SOAP	Simple Object Access Protocol -Giao thức sử dụng XML để định nghĩa dữ liệu dạng thuần văn bản (plain text) thông qua HTTP
XML	eXtensible Markup Language - Ngôn ngữ Đánh dấu Mở rộng

DANH MỤC HÌNH

Hình 1.1. So sánh Web 1.0 và Web 2.0 11

Hình 2.1. Mô tả phương thức arc() 22

Hình 2.2. Ví dụ phương thức createLinearGradient()..... 24

Hình 2.3. Ví dụ phương thức createRadialGradient()..... 24

Hình 2.4. Ví dụ vẽ hình bằng SVG 26

Hình 2.5. Môi liên hệ JS, WebGL và GPU 27

Hình 2.6. Ví dụ sử dụng WebGL trên Canvas 28

Hình 2.7. Ví dụ WebGL - Biểu đồ TeeChart 3D 29

Hình 2.8. Ví dụ WebGL – Google Search (3D Graph)..... 30

Hình 2.9. Ví dụ WebGL – Trò chơi 3D Gwt Quake II 30

Hình 2.10. Ví dụ sử dụng Geolocation API và Google Maps 37

Hình 2.11. Quá trình bắt tay Websocket 42

Hình 2.12. Ví dụ kiểu input: color 44

Hình 2.13. Ví dụ kiểu input: date 44

Hình 2.14. Ví dụ kiểu input: datetime..... 45

Hình 2.15. Ví dụ kiểu input: datetime-local..... 45

Hình 2.16. Ví dụ kiểu input: email..... 45

Hình 2.17. Ví dụ kiểu input: number 46

Hình 2.18. Ví dụ kiểu input: range..... 46

Hình 2.19. Ví dụ kiểu input: search 46

Hình 2.20. Ví dụ kiểu input: tel..... 46

Hình 2.21. Ví dụ kiểu input: time 47

Hình 2.22. Ví dụ kiểu input: url 47

Hình 2.23. Ví dụ kiểu input: week 47

Hình 2.24. Ví dụ thành phần <datalist> 48

Hình 2.25. Ví dụ thành phần <keygen> 48

Hình 2.26. Ví dụ thành phần <output> 49

Hình 2.27. Ví dụ drag-and -drop..... 51

Hình 2.28. Thực hiện kéo thả.....	53
Hình 2.29. Ví dụ sử dụng Web Worker	55
Hình 3.1. Giao diện trang xem video	61
Hình 3.2. Giao diện trang upload	62
Hình 3.3. Giao diện ứng dụng Xác định vị trí.....	62
Hình 3.4. Ứng dụng vẽ đồ thị 3D TeeChart.....	63
Hình 3.5. Trò chơi đua phi thuyền HexGL	63

DANH MỤC BẢNG

Bảng 1.1. Dấu hiệu phân biệt Web 1.0 và Web 2.0	12
Bảng 2.1. Những kiểu nội dung trong HTML5	17
Bảng 2.2. Những thành phần vùng nội dung trong HTML5.....	18
Bảng 2.3. So sánh giữa Canvas và SVG	26
Bảng 2.4. Các trình duyệt hỗ trợ các chuẩn codec khác nhau	31
Bảng 2.5. Các thuộc tính của <video>	32
Bảng 2.6. Các thuộc tính của thẻ <audio>:	33
Bảng 2.7. Các sự kiện của <audio> và <video>	33
Bảng 2.8. Các thuộc tính của phương thức getCurrentPosition()	38
Bảng 2.9. Các thuộc tính mới của <form>	49
Bảng 2.10. Các thuộc tính mới của <input>	49
Bảng 2.11. Mô tả cấu trúc một file manifest.....	58

CHƯƠNG 1: TÌM HIỂU CÔNG NGHỆ WEB-BASED, WEB 2.0

1.1 Giới thiệu Web-based

Web-based technology hay công nghệ dựa trên nền tảng web là một thuật ngữ dùng để chỉ những ứng dụng hay phần mềm được sử dụng dựa trên nền tảng web. Tức là những ứng dụng hay phần mềm có thể truy cập thông qua trình duyệt trên hệ thống mạng như internet hay intranet.

Những ứng dụng web được xây dựng thông qua những ngôn ngữ mà các trình duyệt hỗ trợ như HTML, JavaScript... Những ứng dụng dựa trên nền tảng web hay ứng dụng web (web application) ngày càng trở lên rất phổ biến vì những ưu điểm vượt trội của nó, mà đặc biệt là ưu điểm to lớn đối với người sử dụng (hay người sử dụng cuối cùng) trên các máy trạm (clients).

Ưu điểm đối với phần máy trạm ở chỗ sử dụng những ứng dụng (application) hay phần mềm (chẳng hạn gmail, những điểm bán lẻ,...) mà không cần phải cài đặt chương trình gì mà chỉ cần chạy thông qua web. Bên cạnh những ưu điểm vượt trội về máy trạm, những ứng dụng web còn rất nhiều ưu điểm khác như: Tự động update chương trình thông qua việc update tại máy chủ, việc dùng trình duyệt làm việc có thể kết hợp với các ứng dụng web khác như mail, tìm kiếm. Người sử dụng có thể chạy chương trình trên mọi hệ điều hành như Windows, Linux, Mac... bởi chúng ta chỉ cần có mỗi trình duyệt để làm việc. Ngoài ra, máy tính của chúng ta cũng không cần đòi hỏi quá cao về cấu hình, đĩa trống...

1.2 Giới thiệu Web 2.0

1.2.1 Khái niệm

World Wide Web, gọi tắt là Web hoặc WWW, mạng lưới toàn cầu là một không gian thông tin toàn cầu mà mọi người có thể truy cập (đọc và viết) qua các máy tính nối với mạng Internet. Web được phát minh và đưa vào sử dụng vào khoảng năm 1990, 1991 bởi viện sĩ Viện Hàn lâm Anh Tim Berners-Lee và Robert Cailliau (Bi) tại CERN, Geneva, Switzerland.

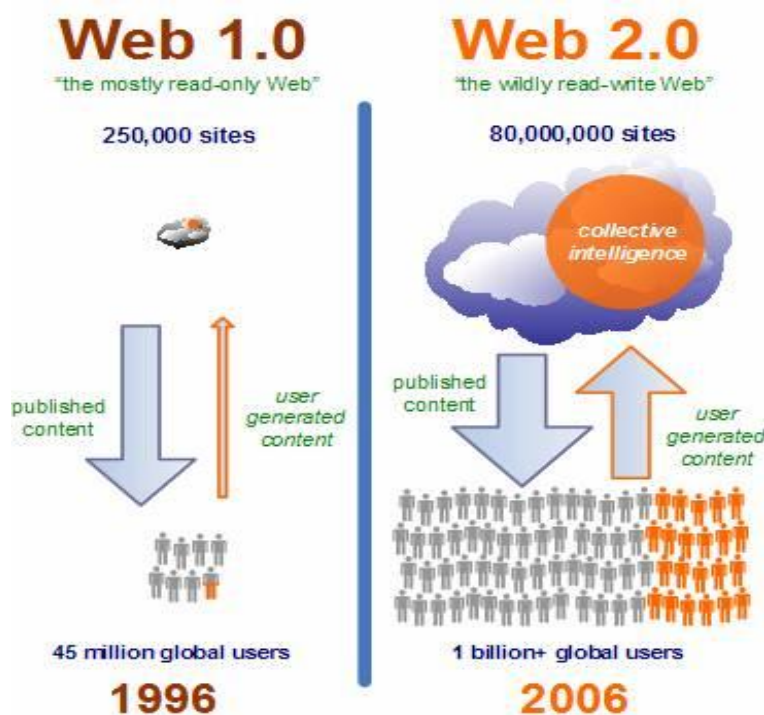
Các tài liệu trên World Wide Web được thể hiện bằng một hệ thống siêu văn bản (hypertext), đặt tại các máy tính trong mạng Internet. Người dùng phải sử dụng một chương trình được gọi là trình duyệt Web để xem siêu văn bản. Chương trình này sẽ nhận thông tin (documents) tại ô địa chỉ (address) do người sử dụng yêu cầu (thông tin trong ô địa chỉ được gọi là tên miền (domain name)), rồi sau đó chương trình sẽ tự động gửi thông tin đến máy chủ (web server) và hiển thị trên màn hình

máy tính của người xem. Người dùng có thể theo các liên kết siêu văn bản (hyperlink) trên mỗi trang web để nối với các tài liệu khác hoặc gửi thông tin phản hồi theo máy chủ trong một quá trình tương tác. Hoạt động truy tìm theo các siêu liên kết thường được gọi là duyệt Web.

1.2.2 Web 2.0

Web 1.0 gần như là những trang web chỉ phép người dùng đọc và đọc (read-only web). Ví dụ một trang tin, báo điện tử với những mục tin có sẵn cho chúng ta.

Web 2.0 thì làm được nhiều hơn thế. Khi nói tới web 2.0 người ta nhấn mạnh tới ảnh hưởng về xã hội của web hơn là các yếu tố kỹ thuật. Web 2.0 tạo cơ hội cho người dùng sử dụng web theo một cách khác so với trước kia. Họ không còn là những người tiếp nhận thông tin thụ động mà là người tham gia tạo nên nội dung của nó. Đó là các trang web có thể “đọc và viết” (read-and-write), tất nhiên phải hiểu theo nghĩa rộng của từ “viết” bao gồm cả hình ảnh, video và nhiều thứ khác tạo nên các nội dung web đa phương tiện ngày nay.



Hình 1.1. So sánh Web 1.0 và Web 2.0

Khái niệm Web 2.0 đầu tiên được Dale Dougherty, phó chủ tịch của O'Reilly Media, đưa ra tại hội thảo Web 2.0 lần thứ nhất do O'Reilly Media và MediaLive International tổ chức vào tháng 10/2004. Dougherty không đưa ra định nghĩa mà chỉ dùng các ví dụ so sánh phân biệt Web 1.0 và Web 2.0: "DoubleClick là Web 1.0; Google AdSense là Web 2.0. Ofoto là Web 1.0; Flickr là Web 2.0. Britannica Online là Web 1.0; Wikipedia là Web 2.0, v.v..."

Bảng 1.1. Dấu hiệu phân biệt Web 1.0 và Web 2.0

Mức độ	Web 1.0	Web 2.0
Mức độ tập trung	Tập trung một nơi	Phân tán nhiều nơi
Mức độ tương tác	Dành cho cá nhân	Dành cho cá nhân , tập thể, xã hội
Mức độ nội dung	Cung cấp nội dung	Cung cấp các dịch vụ và hệ giao tiếp lập trình ứng dụng (APIs)
Mức độ sử dụng	Đọc được	Đọc được, viết được
Mức độ liên kết	Truyền phát giữa các hệ thống	Đồng bộ giữa các hệ thống
Mức độ hệ thống	Hệ thống bao gồm cấu trúc, nội dung tạo ra đã có tính toán trước	Tự sản sinh, tự đề xuất
Mức độ dữ liệu	Tĩnh	Động
Mức độ truy xuất	Cứng nhắc, không linh hoạt	Quan hệ mềm dẻo, lỏng

Cùng với việc ra đời của Web 2.0 cũng không thể không kể đến việc các công nghệ nền tảng mới được phát triển nhằm giúp cho ứng dụng web trở nên “mạnh” hơn, nhanh hơn và dễ sử dụng hơn. Một số công nghệ phổ biến: AJAX tạo web có tính tương tác cao hơn với người dùng. RSS, RDF, *Atom* những giao thức giúp cung cấp nội dung và sử dụng chúng theo cách mà người dùng muốn. Liên quan đến dịch vụ web có một số giao thức truyền thông 2 chiều như REST, SOAP. Và để web trao đổi thông tin được an toàn hơn có giao thức HTTPS.

1.2.3 Kiến trúc cơ bản của Web

Web có kiến trúc hai tầng đó là một web client để hiển thị nội dung thông tin và web server để chuyển thông tin cho web client đó. Kiến trúc này phụ thuộc vào ba tiêu chuẩn chính: HTML cho mã hóa nội dung tài liệu, URL để đặt tên cho các đối tượng thông tin từ xa trong một không gian toàn cầu, và HTTP cho vận chuyển thông tin.

- *HyperText Markup Language (HTML)*: là một ngôn ngữ đánh dấu được thiết kế ra để tạo nên các trang web với các mẫu thông tin được trình bày trên World Wide Web. HTML được định nghĩa như là một phần nhỏ của SGML và được sử dụng trong các tổ chức cần đến các yêu cầu xuất bản phức tạp.

HTML đã trở thành một chuẩn Internet do tổ chức World Wide Web Consortium (W3C) duy trì.

- *Universal Resource Identifier (URI)*: là một giao thức địa chỉ IETF cho các đối tượng WWW. Có hai loại URI đó là: Universal Resource Names (URN) và Universal Resource Locators (URL).
- *HyperText Transfer Protocol (HTTP)*: Giao thức truyền tải siêu văn bản- là một trong năm giao thức chuẩn về mạng Internet, được dùng để liên hệ thông tin giữa Máy cung cấp dịch vụ (Web server) và Máy sử dụng dịch vụ (Web client) là giao thức Client/Server dùng cho World Wide Web-WWW, HTTP là một giao thức ứng dụng của bộ giao thức TCP/IP (các giao thức nền tảng cho Internet).

CHƯƠNG 2: TÌM HIỂU CÁC CÔNG NGHỆ TRONG HTML 5.0

HTML đóng vai trò rất quan trọng trong việc biểu diễn, lưu trữ và truyền tải thông tin trên internet. Và đã có quá trình phát triển qua nhiều giai đoạn. Dưới đây là cái nhìn tổng quan về HTML5 và những công nghệ của nó.

2.1 Sự phát triển của HTML

- *HTML 1.0*: Phiên bản đầu tiên của HTML được phát hành vào năm 1991 bởi Tim Berners-Lee với tên gọi HTML. HTML 1.0 cung cấp một nền tảng độc lập trong việc đánh dấu dữ liệu để trao đổi. Phiên bản chỉ bao gồm 20 thành phần (elements), 13 trong đó vẫn còn lại ở phiên bản HTML 4.01.
- *HTML+*: Năm 1993 Dave Ragget đã gợi ý một bản nháp cho HTML đó là HTML+. HTML+ kết hợp các thành phần đồ họa và hiển thị vào HTML. Các thành phần mũ trên, mũ dưới, chú thích, lề, chèn và xóa văn bản.
- *HTML 2.0*: Được phát hành năm 1994 và trở thành chuẩn chính thức đầu tiên của HTML – tiêu chuẩn cơ bản mà tất cả các trình duyệt áp dụng cho đến HTML 3.2. Tại phiên bản này được thêm các thành phần: INPUT, SELECT, OPTION và TEXTAREA cộng với các thành phần BR cho ngắt dòng. Nó cũng được thêm vào các thành phần META để mô tả chi tiết các tài liệu, rồi thay đổi những miêu tả về vùng đầu trang (head) và phần thân (body).
- *HTML 3.0*: Được phát hành bản nháp năm 1995. Phiên bản đã hỗ trợ bảng, các thành phần FIG, các thuộc tính ALIGN để căn lề và bổ sung thêm các thuộc tính ảnh nền, tab, chú thích, banner. Ở phiên bản này, CSS trở thành một đề nghị của W3C dùng trong tạo hình Web.
- *HTML 3.2*: Phát hành bản nháp đầu năm 1997, được thêm thẻ SCRIPT và Style. Cũng cung cấp những thành phần và thuộc tính mới cho trang web trở nên sinh động với ảnh động, màu sắc và âm thanh. đây là thời gian phổ biến của việc sử dụng khung trang (frame) và tự động chạy các tệp tin nhạc midi.
- *HTML 4.0*: Phát hành tháng 7-1997 và được chính thức vào 4-1998, giới thiệu các thành phần OBJECTS, và STYLE, DIV và SPAN để kết hợp với CSS
- *HTML 4.01*: Phát hành tháng 12-1999 và được W3C đề nghị sử dụng. HTML 4.01 hỗ trợ các tùy chọn đa phương tiện, các ngôn ngữ kịch bản, style, in ấn và tạo sự thuận tiện cho người dùng là người khuyết tật. HTML 4.01 có những bước đi lớn trong việc quốc tế hóa các văn bản với mục tiêu là làm cho trang web trở lên toàn cầu hóa.
- *XHTML 1.0*: Sự kết hợp giữa HTML và XML, được khuyến cáo sử dụng của W3C vào tháng 2-2000. XHTML có cú pháp chặt chẽ hơn HTML. Tuy nhiên

XHTML không được nhiều nhà phát triển sử dụng cũng vì chính sự chặt chẽ này.

- *HTML 5.0 (hay HTML5)*: Năm 2008, bản nháp đầu tiên của HTML5 được phát hành bởi Web Hypertext Application Technology (WHAT) Work Group. Và hiện tại WHATWG cùng với W3C đang cùng nhau phát triển phiên bản này. HTML5 như là một tiêu chuẩn HTML mới mà tất cả các nhà phát triển đang sử dụng.

2.2 Giới thiệu HTML 5.0

2.2.1 HTML5 là gì?

HTML5 (HyperText Markup Language): ngôn ngữ đánh dấu siêu văn bản phiên bản 5 hay nói cách khác HTML5 là một ngôn ngữ cho việc xây dựng cấu trúc và thể hiện nội dung trên web, một công nghệ cốt lõi của Internet. Nó là phiên bản mới nhất của chuẩn HTML và hiện tại vẫn còn đang trong giai đoạn phát triển. Sự cốt lõi của nó hướng tới sự cải tiến về mặt ngôn ngữ trong việc hỗ trợ các đa phương tiện mới nhất trong khi vẫn giữ cho con người có thể đọc nội dung một cách dễ dàng và các máy tính hay thiết bị có thể xử lý một cách thông nhất.

HTML5 được xây dựng để thỏa mãn bốn tiêu chí sau:

- *Khả năng tương thích*: HTML5 vẫn giữ lại các cú pháp truyền thống trước đây, và nếu một vài tính năng mới nào đó của HTML5 chưa được trình duyệt hỗ trợ thì nó phải có một cơ chế fall back để render trong các trình duyệt cũ. Đương nhiên là, HTML5 không thể xóa bỏ tất cả những gì đã có suốt hơn 20 năm chỉ trong một ngày. Mặc dù điều này cũng không đồng nghĩa với việc HTML5 hỗ trợ tất cả các trình duyệt, nhưng nếu bạn có một trình duyệt đủ cũ để không tương thích với HTML5, có lẽ đã đến lúc ta nên nâng cấp trình duyệt mới!
- *Tính tiện dụng*: đặt người dùng lên hàng đầu nên cú pháp của HTML5 khá thoải mái (dù chưa được chặt chẽ như XHTML), thiết kế hỗ trợ sẵn bảo mật, và sự tách biệt giữa phần nội dung và trình bày ngày càng thể hiện rõ: công việc định dạng hầu hết do CSS đảm nhiệm, HTML5 không còn hỗ trợ phần lớn các chức năng định dạng trong các phiên bản HTML trước đây.
- *Khả năng hoạt động xuyên suốt giữa các trình duyệt*: HTML5 cung cấp các khai báo đơn giản hơn và một API mạnh mẽ. Một ví dụ dễ thấy là khai báo DOCTYPE.

- *Khả năng truy xuất rộng rãi:* HTML5 mang lại sự hỗ trợ tốt hơn cho các ngôn ngữ trên thế giới và cho người khuyết tật, đồng thời cũng có thể hoạt động trên các thiết bị và nền tảng khác nhau.

HTML5 cung cấp:

- Các thẻ mô tả chính xác những gì chúng được thiết kế để chứa đựng.
- Tăng cường khả năng truyền thông trên mạng.
- Cải thiện khả năng lưu trữ chung.
- Các trình làm việc trên nền Web (Web Workers) để chạy các quá trình nền.
- Giao diện WebSocket để thiết lập kết nối liên tục giữa các ứng dụng cư trú và máy chủ.
- Truy vấn dữ liệu đã được lưu trữ tốt hơn.
- Cải thiện tốc độ nạp và lưu trang.
- Hỗ trợ cho CSS3 để quản lý giao diện người dùng đồ họa (GUI), có nghĩa là HTML5 có thể được định hướng nội dung.
- Cải thiện xử lý biểu mẫu trình duyệt.
- Một API cơ sở dữ liệu dựa trên-SQL cho phép lưu trữ cục bộ, phía máy khách..
- Canvas và video, để thêm đồ họa và video mà không cần cài đặt các plugin của bên thứ ba.
- Đặc tả Geolocation API (API định vị toàn cầu), sử dụng khả năng định vị của máy điện thoại thông minh để kết hợp các dịch vụ và các ứng dụng đám mây di động.
- Các biểu mẫu cải tiến làm giảm nhu cầu phải tải về mã JavaScript, cho phép truyền thông hiệu quả hơn giữa các thiết bị di động và các máy chủ điện toán đám mây.

CSS và JavaScript -sẽ là thiếu sót nếu nhắc tới HTML mà bỏ quên hai ngôn ngữ này. Cũng như HTML5, CSS3 vẫn đang trong giai đoạn phát triển, nhưng những tính năng của nó đã làm cho nhiều người phải háo hức. CSS3 cho phép thực hiện một cách dễ dàng những điều mà trước đây phải tốn rất nhiều công sức, chẳng hạn như các đường viền (border) với các góc cạnh được bo tròn, hay thậm chí là xoay đối tượng theo các hướng khác nhau. HTML5 và CSS3 có nhiều tiềm năng sẽ trở thành một cặp đôi ăn ý, nhất là khi mà HTML5 đang đi theo hướng tách biệt giữa nội dung và giao diện. Trách nhiệm trình bày một trang web sẽ từ từ được nhường lại cho CSS3.

Song song với chuyện đó, DOM trong HTML5 cũng được hoàn thiện với bộ Selectors API mới đơn giản hơn, cung cấp nhiều phương thức truy xuất chính xác

đến từng phần tử trên trang web mà không cần các vòng lặp phức tạp duyệt qua từng phần tử như trước. Cộng với việc các trình duyệt hiện đại đã tăng tốc thực thi JavaScript đáng kể, hơn nữa còn cung cấp các công cụ debug tiện lợi, việc phát triển ứng dụng với JavaScript đã trở nên nhẹ nhàng hơn bao giờ hết.

2.2.2 Những điểm mới trong HTML5

DOCTYPE và Character Set mới

Những phiên bản HTML trước có độ dài về khai báo DOCTYPE khá dài và gây khó khăn cho người lập trình để phải nhớ nó.

Ví dụ:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

HTML5 đã giải quyết vấn đề này bằng việc ta chỉ cần khai báo rất đơn giản như sau:

```
<!DOCTYPE html>
```

Tương tự với Character Set, với các phiên bản cũ:

```
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
```

Bây giờ:

```
<meta charset="utf-8">
```

Những thành phần mới và những thành phần bị loại bỏ

HTML5 giới thiệu những thành phần đánh dấu mới, được nhóm thành 7 kiểu nội dung khác nhau, được biểu diễn ở Bảng 2.1.

Bảng 2.1. Những kiểu nội dung trong HTML5

Kiểu nội dung	Miêu tả
Embedded	Kiểu nhúng, nội dung được nhập từ những nguồn khác vào trang web, ví dụ: <code>audio</code> , <code>video</code> , <code>canvas</code> và <code>iframe</code>
Flow	Những phần tử được sử dụng trong phần thân của trang web hay ứng dụng, ví dụ: <code>form</code> , <code>h1</code> và <code>small</code>
Heading	Vùng đầu trang, ví dụ: <code>h1</code> , <code>h2</code> và <code>hgroup</code>
Interactive	Những nội dung mà người dùng tương tác với, ví dụ: <code>audio controls</code> , <code>video controls</code> , <code>button</code> và <code>textarea</code>

Metadata	Những thành phần thường tìm thấy ở phần đầu trang. Phần tử này cung cấp thông tin về trang web, được sử dụng cho những phần trình bày ở sau, ví dụ: <code>script</code> , <code>style</code> và <code>title</code>
Phrasing	Văn bản và những phần tử dùng để định dạng văn bản, ví dụ: <code>mark</code> , <code>kbd</code> , <code>sub</code> và <code>sup</code>
Sectioning	Những phần tử dùng để xác định một vùng nào đó trong trang web, ví dụ: <code>article</code> , <code>aside</code> và <code>title</code>

Hầu hết những thành phần đều có thể được sử dụng trong CSS, ngoài ra một số khác có thể sử dụng thông qua các API kèm theo, ví dụ: `canvas`, `audio`, `video`.

Có nhiều thành phần bị loại bỏ đi trong HTML5 bởi vì chúng bị lỗi thời trong việc sử dụng CSS ví dụ: `big`, `center`, `font`, `basefont`...

Đánh dấu ngữ nghĩa (semantic markup)

Một kiểu nội dung mà bao gồm nhiều những thành phần HTML5 mới đó là kiểu vùng nội dung (section). Theo một phân tích của Google và Opera với hàng triệu trang web đã khám phá ra những tên ID chung cho các thẻ DIV và tìm thấy một số lượng lớn những tên được lặp lại. Ví dụ, nhiều người dùng thẻ DIV với ID="footer" để đánh dấu nội dung phần cuối trang. HTML5 cũng cấp những thành phần vùng nội dung như vậy được thể hiện ở Bảng 2.2.

Bảng 2.2. Những thành phần vùng nội dung trong HTML5

Thành phần	Miêu tả
<code>header</code>	Nội dung đầu trang, dùng cho một trang hoặc một vùng của một trang
<code>footer</code>	Nội dung cuối trang, dùng cho một trang hoặc một vùng của một trang
<code>section</code>	Một vùng trong một trang web
<code>article</code>	Nội dung bài viết độc lập
<code>aside</code>	Liên quan đến nội dung, chú thích
<code>nav</code>	Hỗ trợ định hướng

Tất cả những thành phần này đều có thể được định dạng với CSS. HTML5 đã tách riêng phần trình bày và nội dung, vì vậy bạn có thể tùy ý sử dụng CSS để tạo hình cho trang web của bạn trong HTML5.

Đơn giản hóa việc lựa chọn bằng sử dụng Selectors API mới

DOM trong HTML5 cũng được hoàn thiện với bộ Selectors API mới đơn giản hơn, cung cấp nhiều phương thức truy xuất chính xác đến từng phần tử trên trang web mà không cần các vòng lặp phức tạp duyệt qua từng phần tử như trước.

Những phương pháp chúng ta vẫn dùng để tìm các thành phần là: `getElementsById()`, `getElementsByName()` và `getElementsByTagName()`. Trong HTML5 có thêm 2 phương thức mới trong bộ Selector API là: `querySelector()` và `querySelectorAll()`. Hai phương thức mới này tìm tới các thành phần bằng kết hợp của một nhóm selector.

Miêu tả cụ thể về hai phương thức này theo W3C là:

- Phương thức `querySelector()`: Trả lại phần tử đầu tiên của trang mà được xác định bởi một hoặc nhiều luật selector.
- Phương thức `querySelectorAll()`: Trả lại tất cả phần tử mà được xác định bởi một hoặc nhiều luật.

Ta cũng có thể gửi nhiều hơn một luật tới hàm Selector API.

Ví dụ: Chọn phần tử trong tài liệu với lớp `highClass` hoặc lớp `lowClass`

```
var x = document.querySelector(".highClass", ".lowClass");
```

Bộ Selector API mới giúp cho chúng ta dễ dàng hơn trong việc chọn các section của trang web mà trước đó ta rất khó khăn để chọn nó. Ví dụ việc ta muốn xác định bất cứ ô nào của bảng khi ta di chuột qua nó.

window.JSON

JSON là một cách tương đối mới và ngày càng phổ biến trong cách thể hiện lưu trữ và trao đổi dữ liệu. JSON sử dụng cú pháp JavaScript để mô tả đối tượng dữ liệu, nhưng JSON là ngôn ngữ và nền tảng độc lập. JSON đang dần trở thành một chuẩn cho trao đổi dữ liệu trong các ứng dụng HTML5. Bộ API cho JSON có hai hàm cơ bản đó là `parse()` và `stringify()`, có nghĩa là phân tích và chuyển đổi chuỗi.

Những đối tượng JSON gốc là một phần của chuẩn ECMAScript 5, một thế hệ tiếp theo của ngôn ngữ JavaScript. Nó là một trong những phần đầu tiên của ECMAScript 5 được phổ biến rộng rãi. Mọi trình duyệt hiện đại bây giờ đều có

window.JSON, và chúng ta có thể thấy được khá nhiều ứng dụng HTML5 sử dụng JSON.

DOM mức 3

Các trình duyệt phổ biến đều đã hỗ trợ DOM mức 3, nhưng với Internet Explorer thì tới phiên bản IE9 mới hỗ trợ DOM mức 2 và 3. Vì vậy mã HTML5 đều có thể thực hiện được trên tất cả các phiên bản trình duyệt này. Bao gồm cả các phương thức `addEventListener()` và `dispatchEvent()`.

2.3 Những công nghệ mới trong HTML5

2.3.1 Canvas API

Thành phần Canvas cũng cấp một API cho việc vẽ các đối tượng đồ họa, hình ảnh, chữ, phủ màu...

Khởi tạo Canvas

Vùng canvas để vẽ là một vùng hình chữ nhật trên trang HTML, và được quy định là thẻ `<canvas>`. Mặc định thẻ `<canvas>` không có viền và bất cứ nội dung nào trong nó:

```
<canvas id="myCanvas" width="500" height="400"></canvas>
```

`<canvas>` luôn luôn có một thuộc tính `id` dùng để được tham chiếu tới nó khi ta dùng trong để vẽ trong JavaScript. Thuộc tính `width` và `height` xác định chiều rộng và chiều cao của khung vẽ canvas. Nếu không dùng 2 thuộc tính này thì canvas có kích thước mặc định là 300x150 pixel.

Để thêm đường viền, bạn sử dụng thuộc tính `style`.

Ví dụ:

```
<canvas id="myCanvas" width="200" height="100" style="border:1px solid #000000;"></canvas>
```

Vẽ trên Canvas với JavaScript

Mọi hành động vẽ đối tượng trên canvas ta phải thực hiện trong mã JavaScript.

Ví dụ:

```
<script>
var c=document.getElementById("myCanvas");
var ctx=c.getContext("2d");
ctx.fillStyle="#FF0000";
```

```
ctx.fillRect(0,0,150,75);
</script>
```

Giải thích:

Đầu tiên tìm thành phần <canvas> trong trang HTML:

```
var c=document.getElementById("myCanvas");
```

Sau đó gọi phương thức getContext() của nó và phải thêm chuỗi "2d" vào phương thức vì ta đang vẽ các đối tượng 2D:

```
var ctx=c.getContext("2d");
```

Đối tượng getContext("2d") đã được xây dựng sẵn trong HTML5 với nhiều thuộc tính và phương thức khác nhau cho ta thực hiện các đường vẽ, vẽ các hộp, hình tròn, văn bản, hình ảnh và nhiều hơn thế nữa.

Hai dòng lệnh tiếp theo thực hiện vẽ hình chữ nhật:

```
ctx.fillStyle="#FF0000";
ctx.fillRect(0,0,150,75);
```

Thuộc tính fillStyle có thể là một màu CSS, một màu gradient hoặc một mẫu tô. FillStyle mặc định là #000000, màu đen.

Phương thức fillRect(x,y,width,height) thực hiện vẽ hình chữ nhật được phủ với màu được định nghĩa trong fillStyle khai báo trước đó.

Tọa độ trong Canvas

Thành phần canvas trong HTML5 là một mạng lưới hai chiều. Góc trái trên cùng của canvas có tọa độ (0,0). Vì vậy phương thức fillRect() ở ví dụ trên có các tham số (0,0,150,75), có nghĩa là bắt đầu ở vị trí có tọa độ (0,0) vẽ một hình chữ nhật kích thước 150x57 pixel.

Đường vẽ trong Canvas

Để vẽ một đường thẳng trong canvas, chúng ta sử dụng hai phương thức sau:

- moveTo(x,y) : điểm bắt đầu của đường thẳng.
- .lineTo(x,y): điểm kết thúc của đường thẳng.

Và để cho đường thẳng hiện ra ta phải sử dụng phương thức stroke().

Ví dụ: Đoạn mã sau sẽ thực hiện vẽ một đường thẳng có điểm bắt đầu là (0,0) và điểm kết thúc là (200,100):

```
var c=document.getElementById("myCanvas");
var ctx=c.getContext("2d");
ctx.moveTo(0,0);
ctx.lineTo(200,100);
```

```
ctx.stroke();
```

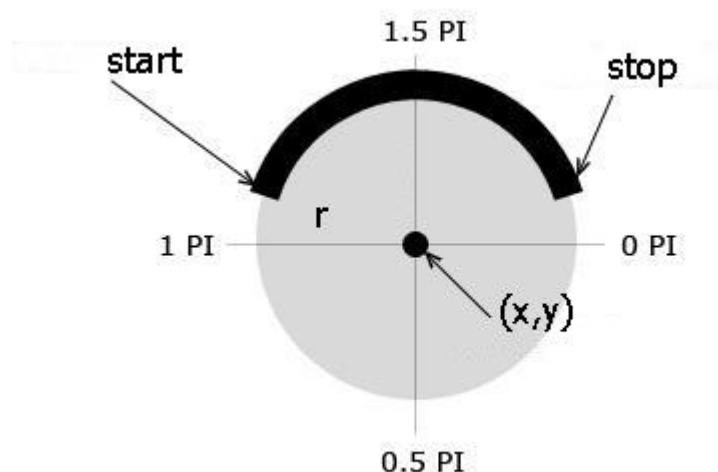
Để vẽ một đường tròn trong canvas, chúng ta sử dụng phương thức sau:

- `arc(x,y,r,start,stop)`

Và cũng để hiện được đường tròn của phương thức `arc()`, ta phải sử dụng một trong 2 phương thức sau: `stroke()` hoặc `fill()`.

Ví dụ: Vẽ một đường tròn với phương thức `arc(x,y,r,start,stop)` có tọa độ tâm $x=95$, $y=50$, bán kính $r=40$, vị trí bắt đầu vẽ là 0 và kết thúc đường tròn tại vị trí $2*\text{Math.PI}$.

```
var c=document.getElementById("myCanvas");
var ctx=c.getContext("2d");
ctx.beginPath();
ctx.arc(95,50,40,0,2*Math.PI);
ctx.stroke();
```



Hình 2.1. Mô tả phương thức `arc()`

Phương thức `arc()` sẽ thực hiện vẽ từ vị trí `start` tới vị trí `stop` theo chiều kim đồng hồ.

Hiển thị văn bản trong Canvas

Để vẽ các ký tự chữ trong canvas, ta có các thuộc tính và phương thức quan trọng sau:

- `font`: xác định phong chữ sẽ hiện thị
- `fillText(text,x,y)`: thực hiện vẽ `text` vào canvas.
- `strokeText(text,x,y)`: hiển thị văn bản lên canvas.

Ví dụ: Thực hiện vẽ đoạn “Hello World” lên canvas, sử dụng phong chữ “Arial”:

```
var c=document.getElementById("myCanvas");
var ctx=c.getContext("2d");
ctx.font="30px Arial";
ctx.fillText("Hello World",10,50);
```

Gradient trong Canvas

Màu gradient có thể được sử dụng để phủ màu cho các đối tượng hình chữ nhật, hình tròn, đường kẻ, văn bản,...

Có hai kiểu màu gradient:

- `createLinearGradient(x,y,x1,y1)`: tạo gradient theo đường thẳng.
- `createRadialGradient(x,y,r,x1,y1,r1)`: tạo gradient theo hình tròn.

Khi đã có đối tượng gradient, ta phải thêm hai hoặc nhiều hơn hai màu dừng cho gradient. Phương thức `addColorStop()` xác định màu dừng và vị trí của nó dọc theo Gradient. Vị trí gradient có thể ở bất kì giá trị nào giữa 0 tới 1.

Để sử dụng gradient, ta sử dụng thuộc tính `fillStyle` hoặc `strokeStyle` để đặt màu gradient, sau đó vẽ hình ta cần.

Ví dụ: sử dụng phương thức `createLinearGradient()` để vẽ hình chữ nhật có màu gradient từ đỏ sang trắng.

```
var c=document.getElementById("myCanvas");
var ctx=c.getContext("2d");

// Create gradient var
grd=ctx.createLinearGradient(0,0,200,0);
grd.addColorStop(0,"red");
grd.addColorStop(1,"white");

// Fill with gradient
ctx.fillStyle=grd;
ctx.fillRect(10,10,150,80);
```



Hình 2.2. Ví dụ phương thức createLinearGradient()

Ví dụ: sử dụng phương thức createRadialGradient():

```
var c=document.getElementById("myCanvas");
var ctx=c.getContext("2d");

// Create gradient var
grd=ctx.createRadialGradient(75,50,5,90,60,100);
grd.addColorStop(0,"red");
grd.addColorStop(1,"white");

// Fill with gradient
ctx.fillStyle=grd;
ctx.fillRect(10,10,150,80);
```



Hình 2.3. Ví dụ phương thức createRadialGradient()

Vẽ ảnh lên Canvas

Để vẽ một hình ảnh lên canvas, ta sẽ sử dụng phương thức sau:

- drawImage(image,x,y)

Ví dụ: Vẽ một ảnh có tên apple.jpg lên canvas.


```
//Tạo ảnh nguồn để vẽ


//Mã JavaScript thực hiện vẽ
<script>
var c=document.getElementById("myCanvas");
var ctx=c.getContext("2d");
var img=document.getElementById("apple");
ctx.drawImage(img,10,10);
</script>
```

2.3.2 Scalable Vector Graphics

SVG là gì?

- SVG là viết tắt của Scalable Vector Graphics
- SVG được sử dụng để vẽ các đối tượng vector dựa trên nền web
- SVG định nghĩa đồ họa trong định dạng XML
- Đồ họa SVG không mất chất lượng hình ảnh khi phóng to hoặc thay đổi kích thước ảnh.
- Mọi thành phần và thuộc tính trong SVG có thể tạo chuyển động.
- SVG được phát triển bởi W3C

Những điểm mạnh của SVG

Ưu điểm của việc sử dụng SVG so với các định dạng ảnh khác (như JPEG và GIF) là:

- Ảnh SVG có thể được tạo và chỉnh sửa với bất kỳ trình biên tập văn bản nào.
- Ảnh SVG có thể được tìm kiếm, lập chỉ mục, nén...
- Ảnh SVG có thể được co giãn kích thước
- Ảnh SVG có thể được in với chất lượng cao ở bất kỳ độ phân giải nào.
- Ảnh SVG có thể phóng to hoặc thu nhỏ mà không mất chất lượng ảnh.

Chèn SVG vào trang HTML

Trong HTML5, SVG có thể được chèn vào trang HTML với thẻ <svg>

Ví dụ:

```
<!DOCTYPE html>
<html>
<body>

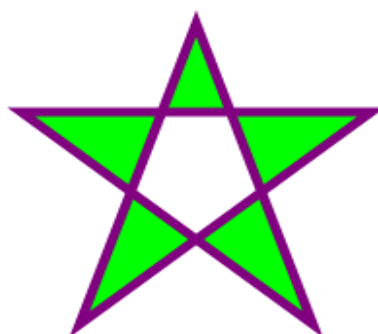
<svg xmlns="http://www.w3.org/2000/svg" version="1.1"
height="190">
```

```

<polygon points="100,10 40,180 190,60 10,60 160,180"
  style="fill:lime;stroke:purple;stroke-width:5;fill-
  rule:evenodd;">
</svg>
</body>
</html>

```

Kết quả:



Hình 2.4. Ví dụ vẽ hình bằng SVG

Điểm khác nhau giữa SVG và Canvas

SVG là một ngôn ngữ miêu tả đồ họa 2D trong XML. Còn canvas xử lý đồ họa 2D với JavaScript

SVG có nền tảng trên XML, mọi thành phần của nó đều trong SVG DOM, tuy nhiên ta có thể sử dụng JavaScript để xử lý các sự kiện cho các thành phần của SVG.

Trong SVG, mỗi hình được vẽ ra sẽ được coi như một đối tượng. Nếu những thuộc tính của một đối tượng SVG được thay đổi, trình duyệt sẽ tự động tạo lại hình đó. Còn canvas được tạo theo các điểm ảnh. Trong canvas một khi hình đồ họa được vẽ, trình duyệt không lưu giữ nhưng thông tin của hình đó. Nếu trình duyệt bị thay đổi thì ta phải vẽ lại hình đó.

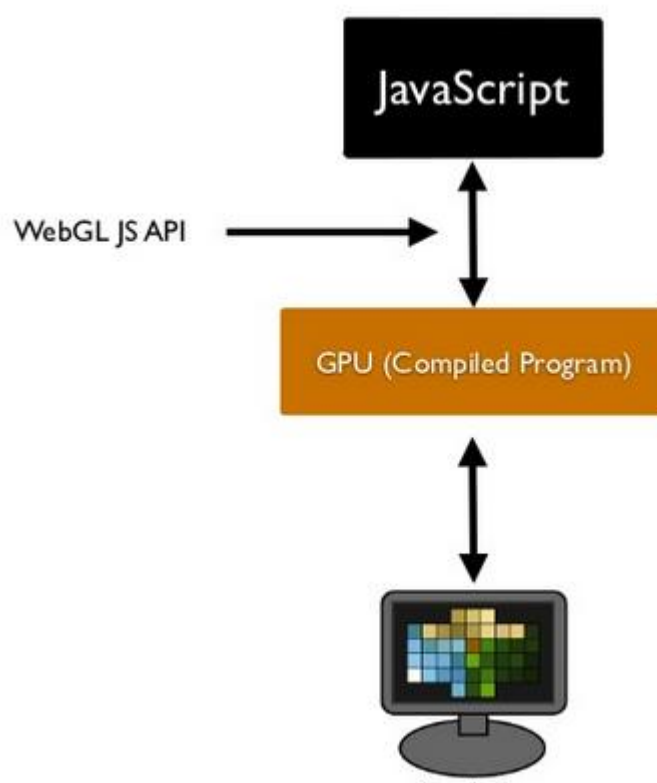
Bảng 2.3. So sánh giữa Canvas và SVG

Canvas	SVG
<ul style="list-style-type: none"> • Độ phân giải bị phụ thuộc • Không hỗ trợ xử lý sự kiện • Khả năng đọc văn bản thấp • Có thể lưu hình ảnh dưới định dạng .png hoặc .jpg • Phù hợp cho các việc tạo các game chuyên sâu 	<ul style="list-style-type: none"> • Độ phân giải không bị phụ thuộc • Hỗ trợ xử lý sự kiện • Phù hợp nhất cho các ứng dụng có kích thước hình lớn • Dạng hình chậm • Không phù hợp cho việc tạo ứng dụng game

2.3.3 WebGL – 3D trên Web

WebGL (Web-based Graphics Library) là một thư viện phần mềm mở rộng khả năng của ngôn ngữ JavaScript để cho phép nó tạo ra tương tác đồ họa 3D trong bất kỳ trình duyệt nào tương thích (Firefox, chrome...). Mã của WebGL thực hiện xử lý trên card xử lý đồ họa (GPU), lưu ý là card đồ họa phải có hỗ trợ shader.

WebGL là một context của phần tử canvas trong HTML5 cung cấp các hàm API 3D graphics. Được phát hành phiên bản đầu tiên 1.0 vào ngày 03 Tháng 3 năm 2011. WebGL được quản lý bởi tổ chức không lợi nhuận Khronos Group.



Hình 2.5. Môi liên hệ JS, WebGL và GPU

Chúng ta có thể dùng WebGL cho nhiều ứng dụng khác nhau như:

- Tạo game 3D
- Hiển thị dữ liệu trực quan
- Xử lý ảnh
- Tạo các ứng dụng 3D
- Và nhiều thứ khác liên quan đến đồ họa 3D cũng như 2D thông qua lập trình

Ví dụ: Thiết lập WebGL và sử dụng để tạo một màu nền xanh:

Đầu tiên là tạo một trang HTML5 có thành phần canvas:

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
<canvas id='canvas'></canvas>
<script>

</script>
</body>
</html>
```

Trong `<script>` thêm dòng sau để tạo một tham chiếu tới canvas vào một biến:

```
var c = document.getElementById('canvas');
```

Dòng lệnh sau sẽ lấy context vẽ của canvas, chú ý ta phải sử dụng context là: `experimental-webgl`:

```
var gl = c.getContext('experimental-webgl');
```

Tiếp đó sử dụng 2 phương thức của WebGL: `clearColor()` xác định màu nền của canvas, `clear()` để làm cho canvas không còn gì trong nó và `COLOR_BUFFER_BIT` để lấy màu ta đã định trước đưa vào canvas:

```
gl.clearColor(0, 0, 0.8, 1);
gl.clear(gl.COLOR_BUFFER_BIT);
```

Khi lưu và tải trang web trên ta sẽ được một hình chữ nhật có kích thước 300x150 (kích thước của canvas) có màu xanh như Hình 2.6:



Hình 2.6. Ví dụ sử dụng WebGL trên Canvas

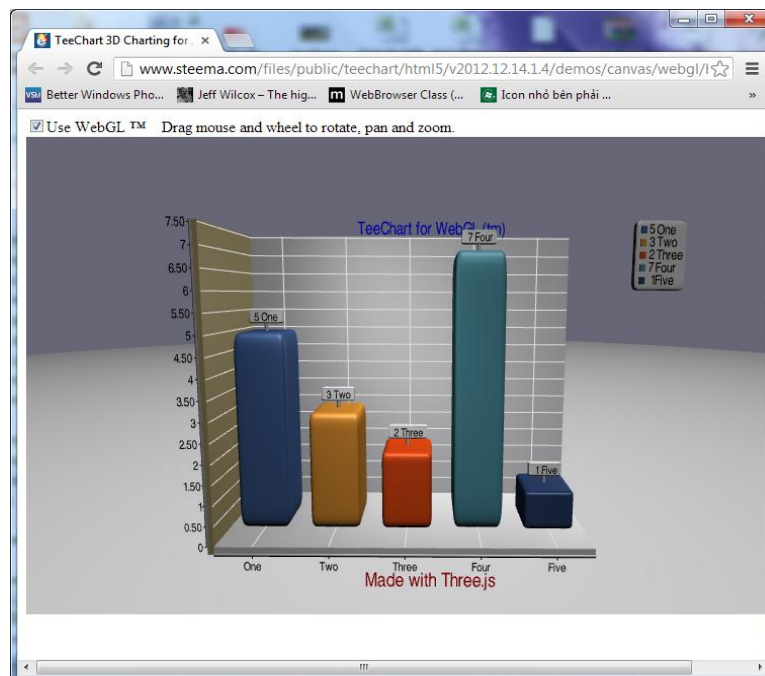
Ví dụ trên chỉ là ví dụ đơn giản để thiết lập và vẽ một hình 2D trong WebGL. WebGL còn nhiều khả năng phức tạp hơn thế để hiển thị các đồ họa 3D. Cũng vì sự phức tạp của cấu trúc lệnh WebGL mà đã có nhiều thư viện hỗ trợ nó,

làm cho cú pháp và cách biểu diễn của WebGL trở nên dễ hiểu hơn với người dùng. Một số thư viện đó là:

- Three.js – Một thư viện rất phổ biến. Nó giúp ta sử dụng không chỉ WebGL mà còn canvas, SVG dễ dàng và làm việc hơn.
- C3DL – Thư viện JS được phát triển bởi Mozilla để sử dụng với các ứng dụng WebGL
- PhiloGL - Một nền tảng JS hỗ trợ ứng dụng WebGL
- GLGE – Thư viện JS hỗ trợ ứng dụng WebGL
- O3D – Nền tảng phát triển bởi Google

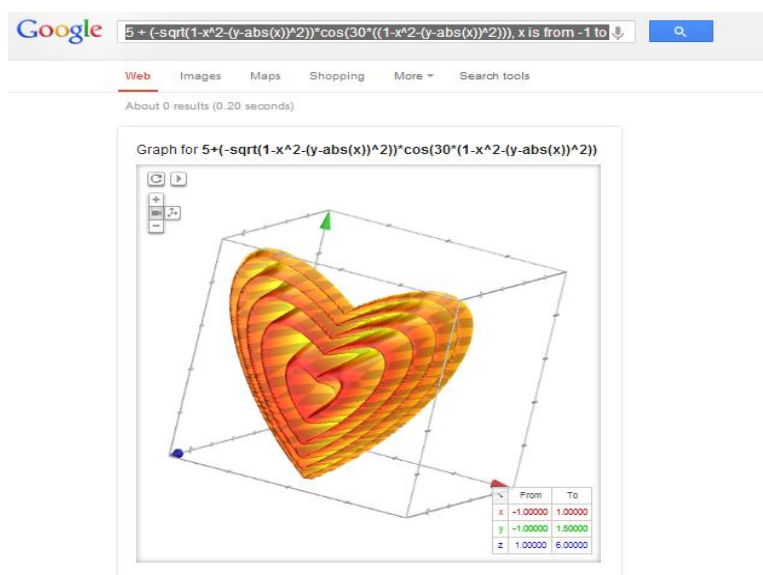
Một số ví dụ về khả năng của WebGL trong xây dựng ứng dụng 3D trên nền Web:

- Vẽ đồ thị 3D trên nền web



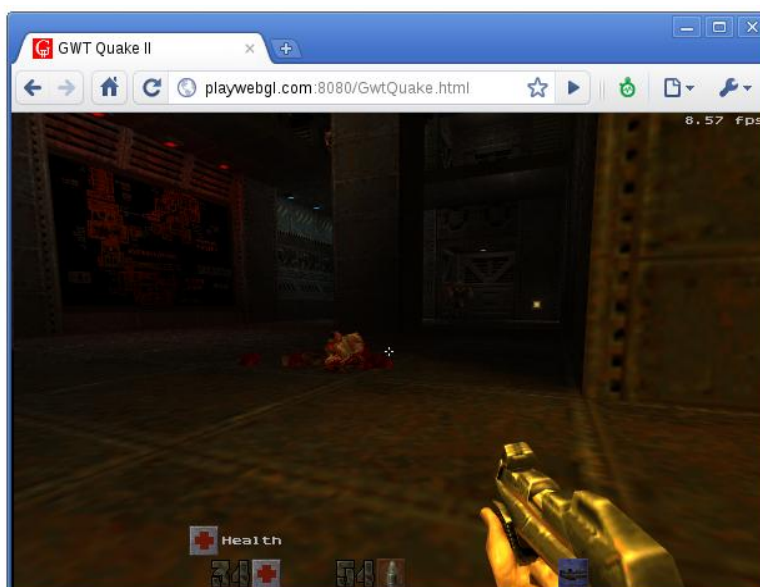
Hình 2.7. Ví dụ WebGL - Biểu đồ TeeChart 3D

- **Vẽ đồ thị 3D**



Hình 2.8. Ví dụ WebGL – Google Search (3D Graph)

- **Trò chơi 3D**



Hình 2.9. Ví dụ WebGL – Trò chơi 3D Gwt Quake II

2.3.4 Audio và Video

HTML5 cho phép chạy các tệp tin (file) âm thanh với thẻ <audio> và phim với thẻ <video> mà không cần tới bất kỳ một plug-in nào đối với trình duyệt. Thông qua các API mới, ta có thể truy cập, điều khiển và thao tác sâu tới các file âm thanh (audio) và phim (video) cũng như trạng thái mạng.

Hiện tại, vẫn chưa có một chuẩn codec nào cho các dữ liệu âm thanh và phim trên web. Tuy nhiên HTML5 cho phép người lập trình có thể tùy chọn nhiều nguồn dữ liệu đa phương tiện khác nhau. Bảng 2.4 là một số chuẩn codec mà các trình duyệt web phổ biến hiện tại hỗ trợ.

Bảng 2.4. Các trình duyệt hỗ trợ các chuẩn codec khác nhau

Trình duyệt	Định dạng dữ liệu và codec	
	Video	Audio
Google Chrome	Ogg (Theora and Vorbis) WebM (VP8 and Vorbis) MPEG 4 (H.264 and AAC)	Ogg (Vorbis, Opus) WebM (Vorbis) MP3 WAV (PCM) ACC
Firefox	Ogg (Theora and Vorbis) WebM (VP8 and Vorbis) MPEG 4 (H.264 and AAC)	Ogg (Vorbis, Opus) WebM (Vorbis) WAV (PCM) MP3 (chỉ trên Windows) AAC (chỉ trên Windows)
Internet Explorer	MPEG 4 (H.264 and AAC)	MP3 AAC
Opera	Ogg (Theora and Vorbis) WebM (VP8 and Vorbis)	Ogg (Vorbis) WebM (Vorbis) WAV (PCM)
Safari	MPEG 4 (H.264 and AAC)	Ogg (Vorbis) WAV (PCM) MP3 AAC

Nhúng phim (video)

Đoạn mã sau giúp ta nhúng một tập tin video có độ rộng và cao khi hiển thị là 300px và 200px với điều khiển và tự động chạy khi trang web được tải vào trong trang web:

```
<video width="300" height="200" controls autoplay>
<source src="video.ogg" type="video/ogg" />
<source src="video.mp4" type="video/mp4" />
Your browser does not support the <video> element.
</video>
```

Thẻ <source> dùng để xác định các kiểu định dạng nguồn file video của ta cùng với các thuộc tính của nó. Một thẻ <video> cho phép sử dụng nhiều thành phần nguồn và trình duyệt sẽ sử dụng định dạng đầu tiên mà nó công nhận.

Dòng thông báo: “Your browser does not support the <video> element. “sẽ hiện ra nếu như trình duyệt không hỗ trợ thẻ <video>.

Để nắm rõ hơn về thẻ <video>, Bảng 2.5 sau sẽ bao gồm tất cả các thuộc tính của thẻ này.

Bảng 2.5. Các thuộc tính của <video>

Thuộc tính	Giá trị	Mô tả
autoplay	autoplay	Xác định video sẽ tự động chạy ngay khi nó sẵn sàng
autobuffer	autobuffer	Xác định video sẽ được tải vào bộ đệm ngay khi không bật chế độ tự động chạy
controls	controls	Xác định thanh điều khiển video có hiển thị hay không.
height	<i>pixels</i>	Xác định chiều cao của player (khung hiển thị video)
loop	loop	Xác định video có được chạy lặp lại hay không
muted	muted	Tắt âm thanh của video
poster	<i>URL</i>	Xác định hình ảnh sẽ hiển thị khi video đang tải hoặc cho tới khi người dùng thực hiện động tác cho chạy video
preload	auto metadata none	Xác định video có được tải sẵn cùng trang web hay không.
src	<i>URL</i>	Xác định đường dẫn của file video
width	<i>pixels</i>	Xác định chiều rộng của player

Nhúng âm thanh (audio)

Tương tự như thẻ <video>, ví dụ sau giúp nhúng một nội dung âm thanh vào trang web:

```
<audio controls autoplay>
<source src="audio.ogg" type="audio/ogg" />
<source src="audio.wav" type="audio/wav" />
    Your browser does not support the <audio> element.
</audio>
```

Bảng 2.6. Các thuộc tính của thẻ <audio>:

Thuộc tính	Giá trị	Mô tả
autoplay	autoplay	Xác định nội dung audio sẽ tự động chạy ngay khi nó sẵn sàng
autobuffer	autobuffer	Xác định nội dung audio sẽ được tải vào bộ đệm ngay khi không bật chế độ tự động chạy
controls	controls	Xác định thanh điều khiển audio có hiển thị hay không.
loop	loop	Xác định audio có được chạy lặp lại hay không
preload	auto metadata none	Xác định nội dung audio có được tải sẵn cùng trang web hay không.
src	URL	Xác định đường dẫn của file audio

Xử lý sự kiện

Để sử dụng thêm nhiều chức năng hơn của 2 thành <video> và <audio>, ta sẽ sử dụng JS để điều khiển chúng. Bảng 2.7 sau cung cấp một số các sự kiện mà ta có thể dùng trong JS.

Bảng 2.7. Các sự kiện của <audio> và <video>

Sự kiện	Mô tả
abort	Xảy ra khi tải một file audio/video bị hủy bỏ
canplay	Xảy ra khi trình duyệt có thể bắt đầu chạy audio/video
ended	Xảy ra khi danh sách phát hiện tại hết

error	Xảy ra khi một lỗi xảy ra trong quá trình tải
loadeddata	Xảy ra khi trình duyệt đã tải xong khung hiện tại của audio/video
loadstart	Xảy ra khi trình duyệt bắt đầu tải audio/video
pause	Xảy ra khi audio/video bị tạm dừng
play	Xảy ra khi audio/video được bắt đầu chạy hoặc chạy sau khi bị tạm dừng
playing	Xảy ra khi audio/video đang chạy
progress	Xảy ra khi trình duyệt đang tải audio/video
ratechange	Xảy ra khi tốc độ chạy audio/video bị thay đổi
seeked	Xảy ra khi người dùng kết thúc di chuyển/nhảy tới một vị trí mới trong các audio/video
stalled	Xảy ra khi trình duyệt cố gắng lấy dữ liệu audio/video nhưng dữ liệu không khả thi
suspend	Xảy ra khi trình duyệt được lệnh đình chỉ việc tải audio/video
volumechange	Xảy ra khi người dùng thay đổi âm lượng
waiting	Xảy ra khi video dừng chạy bởi vì nó cần phải tải những khung hình tiếp theo lưu vào bộ nhớ

Ví dụ sau đây sẽ tạo một khung phát video bằng thẻ <video> và một nút thực hiện việc chạy video sử dụng JS:

```
<!DOCTYPE HTML>
<head>
<script type="text/javascript">
function PlayVideo() {
    var v = document.getElementsByTagName("video")[0];
    v.play();
}
</script>
</head>
<html>
<body>
<form>
<video width="300" height="200" src="clip.mp4">
```

```

    Your browser does not support the <video> element.
</video>
<input type="button" onclick="PlayVideo();" value="Play"/>
</form>
</body>
</html>

```

2.3.5 Geolocation API

Giới thiệu Geolocation API

HTML5 Geolocation API – Tính năng định vị cho phép người dùng chia sẻ vị trí của họ với các ứng dụng web bằng cách bắt lấy các giá trị tương đối của kinh độ và vĩ độ của người dùng khi được cấp quyền. Với tính năng định vị này của HTML5 sẽ rất có ích cho việc sử dụng với nhiều loại ứng dụng web bằng cách thêm thông tin vị trí tự động. Ví dụ như: người dùng có thể chia sẻ vị trí của họ với bạn bè trên các mạng xã hội, hay có thể tạo các chỉ dẫn đường hoặc tạo quảng cáo phù hợp dựa trên vị trí này. Geolocation API có thể được kết hợp với nhiều bản đồ khác nhau, chẳng hạn như Google Maps, Yahoo Maps, Bing Maps hoặc Nokia Ovi Maps.

Geolocation API xác định vị trí người dùng có thể thông qua nhiều nguồn khác nhau. Tuy nhiên không phải tất cả các nguồn đều xác định vị trí là chính xác vì chúng chỉ có tính tương đối. Những nguồn mà Geolocation API có thể dùng:

- Địa chỉ IP
- Global Positioning System (GPS)
- Wi-Fi kết hợp địa chỉ MAC từ RFID, Wi-Fi, và Bluetooth
- Thuê bao di động GSM hoặc CDMA
- Do người dùng tự định nghĩa

Sử dụng Geolocation API

Sử dụng phương thức `getCurrentPosition()` để có được vị trí của người dùng. Ví dụ dưới đây là một ví dụ đơn giản Geolocation trả về giá trị vĩ độ và kinh độ của vị trí người dùng (mà không có xử lý lỗi):

```

<!DOCTYPE html>
<html>
<body>
<p id="demo">Kích vào nút để nhận các giá trị tọa độ:</p>
<button onclick="getLocation()">Tọa độ</button>
<script>
var x=document.getElementById("demo");
function getLocation()
{

```

```

if (navigator.geolocation)
{
    navigator.geolocation.getCurrentPosition(showPosition);
}
else{x.innerHTML="Trình duyệt không hỗ trợ Geolocation.";}
}
function showPosition(position)
{
    x.innerHTML="Vĩ độ: " + position.coords.latitude +
    "<br>Kinh độ: " + position.coords.longitude;
}
</script>
</body>
</html>

```

Giải thích:

- Kiểm tra nếu Geolocation được trình duyệt hỗ trợ
- Nếu được hỗ trợ, sẽ chạy phương thức `getCurrentPosition()`, nếu không hiển thị tin nhắn tới người dùng.
- Nếu phương thức `getCurrentPosition()` được thực hiện, nó sẽ trả về một đối tượng được đưa vào tham số `showPosition`
- Hàm `showPosition()` sẽ hiển thị các giá trị Vĩ độ và Kinh độ

Xử lý lỗi

Tham số thứ 2 của phương thức `getCurrentPosition()` là `showError` được sử dụng để xử lý lỗi. Nó xác định một hàm nào đó sẽ được thực thi khi bị lỗi trong khi lấy vị trí người dùng;

```

function showError(error)
{
    switch(error.code)
    {
        case error.PERMISSION_DENIED:
            x.innerHTML="Người dùng từ chối yêu cầu sử dụng
Geolocation."
            break;
        case error.POSITION_UNAVAILABLE:
            x.innerHTML="Thông tin vị trí không khả thi."
            break;
        case error.TIMEOUT:
            x.innerHTML="Timeout."
            break;
        case error.UNKNOWN_ERROR:
            x.innerHTML="Lỗi không xác định."
            break;
    }
}

```


Ví dụ Hình 2.10 ta sử dụng những dữ liệu vĩ độ và kinh độ được trả lại để hiển thị vị trí trong Google Maps, sử dụng một ảnh tĩnh (Hình 2.10).

Phương thức `getCurrentPosition()`

Ngoài hai giá trị vĩ độ và kinh độ sẽ được trả lại nếu phương thức `getCurrentPosition()` được thực hiện thì có nhiều thuộc tính khác nữa của phương thức này nữa:

Bảng 2.8. Các thuộc tính của phương thức `getCurrentPosition()`

Thuộc tính	Mô tả
<code>coords.latitude</code>	Giá trị vĩ độ (số thập phân)
<code>coords.longitude</code>	Giá trị kinh độ (số thập phân)
<code>coords.accuracy</code>	Độ chính xác của vị trí
<code>coords.altitude</code>	Độ cao trung bình so với mực nước biển
<code>coords.altitudeAccuracy</code>	Tính chính xác của vị trí độ cao
<code>coords.heading</code>	Cung cấp thông tin về hướng tính theo độ, bắt đầu từ hướng bắc
<code>coords.speed</code>	Chỉ ra tốc độ tương đối m/s
<code>timestamp</code>	Trả lại thời gian khi dữ liệu về vị trí được lấy ra

Những phương thức thú vị khác trong Geolocation

`watchPosition()` – Trả lại vị trí hiện tại của người dùng và tiếp tục cập nhật vị trí khi người dùng di chuyển (giống như hệ thống GPS trong ô tô).

`clearWatch()` – Dừng phương thức `watchPosition()`.

2.3.6 Communication APIs

HTML5 cung cấp hai khối giao tiếp thời gian thực quan trọng, đó là: Cross Document Messaging và XMLHttpRequest cấp 2. Cả hai khối này đã thêm nhiều tùy chọn giao tiếp mới để cho phép các ứng dụng HTML5 phục vụ nhiều tên miền khác nhau một cách an toàn.

Cross Document Messaging

Cross Document Messaging là một API cho phép các trang tài liệu giao tiếp với nhau qua những nguồn khác nhau hay những tên miền nguồn khác. Sử dụng phương thức `postMessage()` của API giao tiếp này, những tin nhắn text có thể được gửi đi từ tên miền này tới tên miền khác. Điều này đòi hỏi phải có đối tượng `Window` để nhận tin nhắn đó. Tin nhắn có thể được đăng tới:

- những frame hoặc iframe khác trong cùng cửa sổ gửi
- những cửa sổ mà JS sẽ mở ra khi tin nhắn gửi đi
- của sổ mẹ của tin nhắn gửi đi
- cửa sổ đã mở tin nhắn gửi đi

Khi tin nhắn gửi đi, `events` sẽ nhận các thuộc tính sau:

- `data` – Dữ liệu hoặc nội dung của các tin nhắn gửi đến.
- `origin` – Một chuỗi bao gồm nguồn gốc của tin nhắn, tên miền và cổng.
- `source` – Một tham chiếu tới của sổ của tin nhắn gốc. Chính xác hơn đó là một đối tượng `WindowProxy`

Sử dụng postMessage()

Ta sẽ xét một ví dụ, ta muốn tài liệu A giao tiếp với tài liệu B được đặt trong một iframe hoặc một cửa sổ popup. Mã JS sau là cho tài liệu A:

```
var o = document.getElementsByTagName('iframe')[0];
o.contentWindow.postMessage('Hello B',
'http://documentB.com/');
```

Để nhận tin nhắn từ tài liệu A, ta thêm một sự kiện để nghe. Sử dụng thuộc tính `origin` để kiểm tra tên miền của nơi gửi có đúng với đã thiết lập từ trước hay không. Sau đó tài liệu B sẽ xem tin nhắn và hiện thị tin nhắn đó hoặc có thể trả lời lại tin nhắn tới tài liệu A. Mã JS sau sẽ thực hiện việc này:

```
function receiver(event) {
if (event.origin == 'http://documentA.com') {
    if (event.data == 'Hello B') {
event.source.postMessage('Hello A, how are you?',
event.origin);
    }
else {
alert(event.data);
    }
}
}
window.addEventListener('message', receiver, false);
```

Security

Để bảo đảm sự an toàn cho ứng dụng tránh nhiễm các mã độc hại từ các tên miền không an toàn khi sử dụng Cross-Document Messaging thì ta cần phải kiểm tra kỹ thuộc tính `origin`. Thêm vào đó định dạng dữ liệu gửi tới cũng phải được kiểm tra trước xem có phù hợp với định dạng mà mình mong muốn nhận hay không.

XMLHttpRequest cấp 2

XMLHttpRequest là một đối tượng được sử dụng để trao đổi dữ liệu với một máy chủ ở đằng sau. Điều đó có nghĩa nó có thể cập nhật một hoặc nhiều phần nào đó của trang web mà không phải tải lại trang đó.

Đoạn mã sau giúp tạo một đối tượng XMLHttpRequest:

```
var xhr = new XMLHttpRequest();
```

Sau đó gửi một yêu cầu tới máy chủ:

```
xhr.open('GET', 'example.php');  
xhr.send();
```

Sau đó đợi phản hồi từ máy chủ.

XMLHttpRequest mức 2 - phiên bản mới nhất của XMLHttpRequest – đã có nhiều cải tiến tập trung vào những phần sau:

- Cross-origin XMLHttpRequest
- Sự kiện Progress
- Dữ liệu nhị phân

Cross-origin XMLHttpRequest

Đối tượng XMLHttpRequest có thể gửi yêu cầu tới nhiều máy chủ khác nhau. Được gọi là cross-origin resource sharing (CORS). Nhưng để có thể sử dụng được CORS thì trình duyệt phải hỗ trợ nó và máy chủ phải đồng ý với yêu cầu cross-origin

```
xhr.open('GET', 'http://other.server/and/path/to/script');
```

Progress event

Sự kiện progress sẽ gửi lại thông tin về tiến trình đang thực hiện một việc gì đó của đối tượng XMLHttpRequest. Có hai khả năng của progress là: quá trình tải về và tải lên. Sự kiện progress cho tải về thuộc về đối tượng XMLHttpRequest, trong khi đó đối với tải lên thì thuộc về đối tượng XMLHttpRequest.upload. Chúng ta có thể thiết lập hàm callback cho sự kiện progress như sau:


```
xhr.onprogress = updateProgress;  
xhr.upload.onprogress = updateProgress;
```

Trong hàm callback ta sử dụng một số thuộc tính của sự kiện progress như:

```
function updateProgress(event) {  
    if (event.lengthComputable){  
        var percentComplete = event.loaded / event.total;  
    }  
}
```

Đoạn mã trên, `event.total` tương ứng với tổng số byte được chuyển đi, `event.loaded` nghĩa là số byte đã chuyển, nếu `event.lengthComputable` là `false` thì `event.total=0`.

Có năm sự kiện khác liên quan đến sự kiện progress:

- `load` : chuyển dữ liệu đã thành công
- `abort`: chuyển dữ liệu bị hủy bởi người dùng
- `error`: có lỗi khi chuyển
- `loadstart`: chuyển dữ liệu bắt đầu
- `loadend`: chuyển dữ liệu hoàn thành nhưng chưa chắc việc đã thành công hay không

Nhận dữ liệu nhị phân

Trong phiên bản XMLHttpRequest cũ chỉ có thể nhận dữ liệu text từ máy chủ, với XMLHttpRequest 2 thì có thể nhận dữ liệu nhị phân. Các dữ liệu sẽ được xử lý như văn bản và cần chuyển đổi chúng thành dữ liệu nhị phân (đây là cách ghi đè MIMEType):

```
for (var i = 0, len = binStr.length; i < len; ++i) {  
    var c = binStr.charCodeAt(i);  
    var byte = c & 0xff;  
}
```

2.3.7 WebSocket API

WebSocket là một tính năng giao tiếp mạnh mẽ nhất trong HTML5, nó xác định một kênh truyền thông giao tiếp song công (full-duplex) hai chiều giữa client và server bằng cách sử dụng một TCP socket để tạo một kết nối hiệu quả và ít tốn kém. Dữ liệu truyền tải thông qua giao thức HTTP (thường dùng với kỹ thuật AJAX) chứa nhiều dữ liệu không cần thiết trong phần header. Một header request/response của HTTP có kích thước khoảng 871 byte, trong khi với WebSocket kích thước này chỉ là 2 byte (sau khi đã kết nối).

Giao thức bắt tay của WebSocket

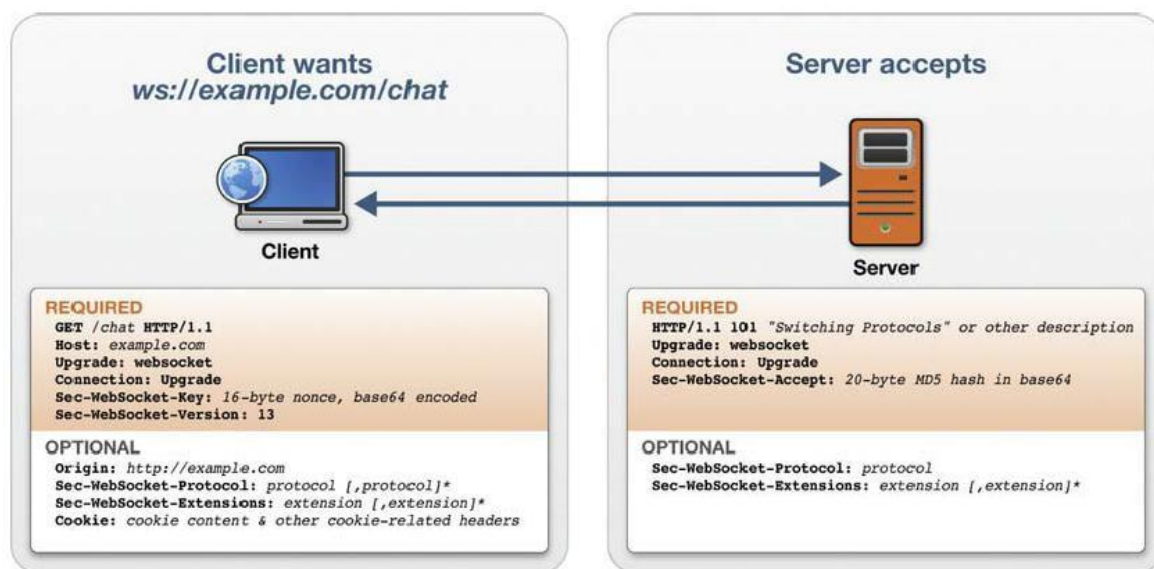
Để thực hiện kết nối, client phải gửi một WebSocket handshake request đến server. Server sẽ gửi trả lại WebSocket handshake response như ví dụ sau:

Từ client tới server:

```
GET /chat HTTP/1.1
Host: example.com
Connection: Upgrade
Sec-WebSocket-Protocol: sample
Upgrade: websocket
Sec-WebSocket-Version: 13
Sec-WebSocket-Key: 7cxQRnWs91xJW9T0QLSuVQ==
Origin: http://example.com
```

Từ server tới client:

```
HTTP/1.1 101 WebSocket Protocol Handshake
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Accept: 7cxQRnWs91xJW9T0QLSuVQ==
WebSocket-Protocol: sample
```



Hình 2.11. Quá trình bắt tay WebSocket

Sau khi đã thiết lập kết nối, các thông điệp WebSocket có thể được gửi qua lại giữa client và server trong chế độ full-duplex. Điều này có nghĩa là các thông điệp văn bản có thể được gửi theo hai hướng cùng một lúc. Mỗi tin nhắn được bắt đầu với một byte 0x00, kết thúc với byte 0xFF và dữ liệu UTF-8 ở giữa.

Sử dụng WebSocket

Để mở một kết nối WebSocket, đơn giản chỉ việc gọi constructor WebSocket, ví dụ:

```
var connection = new WebSocket('ws://example.com/chat', ['soap', 'xmpp']);
```

Chú ý: ws: là giao thức cho kết nối WebSocket, cũng có wss: cho kết nối WebSocket an toàn giống như https: được sử dụng cho kết nối HTTP an toàn. Tham số thứ 2 (ví dụ trên là: ['soap', 'xmpp']) tùy chọn là giao thức phụ, đây là một chuỗi hoặc một mảng chuỗi.

Để gửi dữ liệu tới server ta sử dụng phương thức send():

```
connection.onopen = function () {
    connection.send('Tin nhắn cần gửi');
};
```

Để lưu lại lỗi:

```
connection.onerror = function (error) {
    console.log('WebSocket Error ' + error);
};
```

Để lưu tin nhắn từ server

```
connection.onmessage = function (e) {
    console.log('Server: ' + e.data);
};
```

Để đóng kết nối WebSocket:

```
connection.close();
```

2.3.8 Forms API

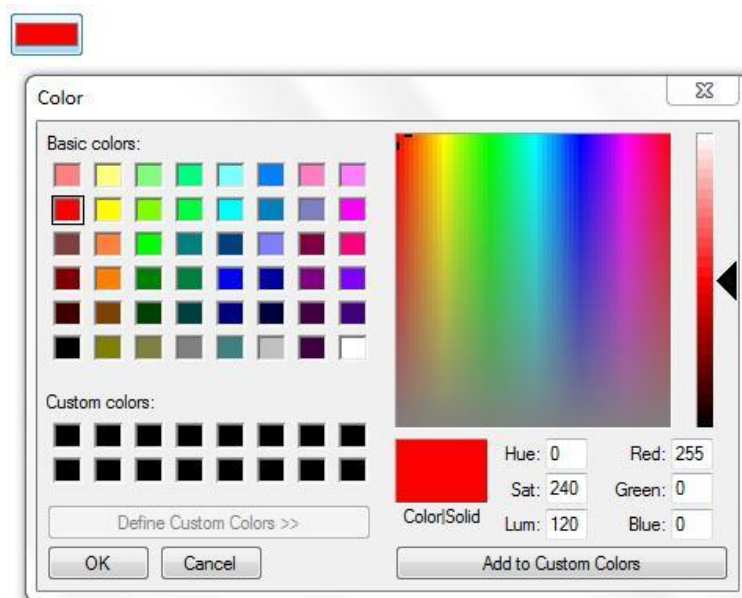
Thành phần Form và các thuộc tính của Form trong HTML5 cung cấp một mức độ đánh dấu ngữ nghĩa lên một mức cao hơn so với HTML4, và nó cũng loại bỏ nhiều mã JS tẻ nhạt và tạo kiểu (styling) không cần thiết bị bắt buộc trong HTML4. Các đặc điểm của form trong HTML5 tạo tính năng động và dễ dùng hơn cho người sử dụng, nó đáp ứng ngay lập tức với dữ liệu mà người dùng nhập vào. Có khả năng làm việc ngay trên những trình duyệt mà các mã kịch bản như JS bị vô hiệu hóa.

Những kiểu input mới

- color
- number
- datetime
- datetime-local
- week
- email
- month
- number
- range
- search
- tel
- time
- url

Kiểu input: color- được sử dụng cho phép người dùng chọn một màu và trả lại giá trị hexa của màu đó.

`<input id="color" name="color" type="color">`



Hình 2.12. Ví dụ kiểu input: color

Kiểu input: date – cho phép người sử dụng chọn ngày

Ngày sinh: `<input type="date" name="bday">`



Hình 2.13. Ví dụ kiểu input: date

Kiểu input: datetime –cho phép xác định một thời gian chính xác gồm cả ngày và giờ (bao gồm timezone)

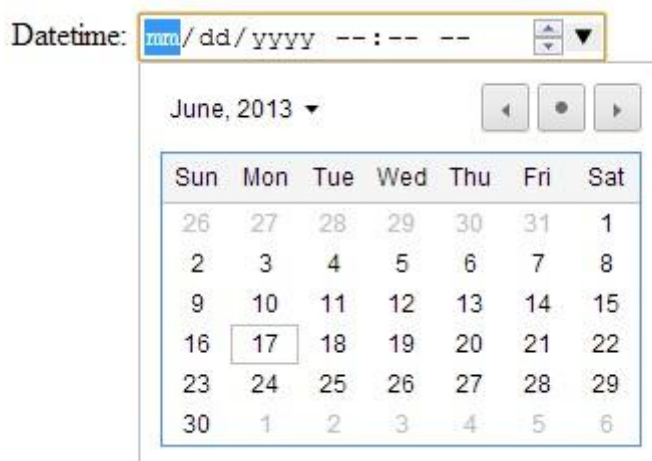
Datetime: `<input type="datetime" name="bday">`



Hình 2.14. Ví dụ kiểu input: datetime

Kiểu input: datetime-local - cho phép xác định một thời gian chính xác gồm cả ngày và giờ (không bao gồm timezone)

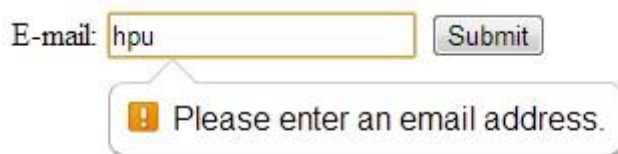
```
Datetime:<input type="datetime" name="bday">
```



Hình 2.15. Ví dụ kiểu input: datetime-local

Kiểu input: email – được sử dụng để nhập địa chỉ email, trình duyệt sẽ tự kiểm tra nếu dữ liệu nhập không phải email sẽ có thông báo.

```
E-mail: <input type="email" name="email">
```



Hình 2.16. Ví dụ kiểu input: email

Kiểu input: month – cho phép người dùng chọn một tháng nào đó trong một năm.

```
<input type="month" name="bdaymonth">
```

Kiểu input: number – cho phép dữ liệu nhập vào có giá trị số, ta có thể thiết lập hạn chế những số được chấp nhận.

Quantity (between 1 and 5):

```
<input type="number" name="quantity" min="1" max="5">
```

Quantity (between 1 and 5):

Hình 2.17. Ví dụ kiểu input: number

Kiểu input: range – cũng giống như kiểu input number nhưng nó cung cấp một điều khiển đơn giản cho người dùng chọn giá trị số.

Value:

Value:

Hình 2.18. Ví dụ kiểu input: range

Kiểu input: search – được sử dụng để nhập giá trị cho tìm kiếm

```
<input type="search" name="googlesearch">
```

Search:

Hình 2.19. Ví dụ kiểu input: search

Kiểu input: tel – sử dụng cho nhập số điện thoại

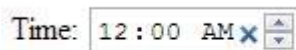
Telephone:



Hình 2.20. Ví dụ kiểu input: tel

Kiểu input: time – cho nhập giá trị giờ

Time: `<input type="time" name="usr_time">`



Hình 2.21. Ví dụ kiểu input: time

Kiểu input: url – cho nhập giá trị là một địa chỉ URL, trình duyệt sẽ kiểm tra tính hợp lệ của URL khi nhập xong.

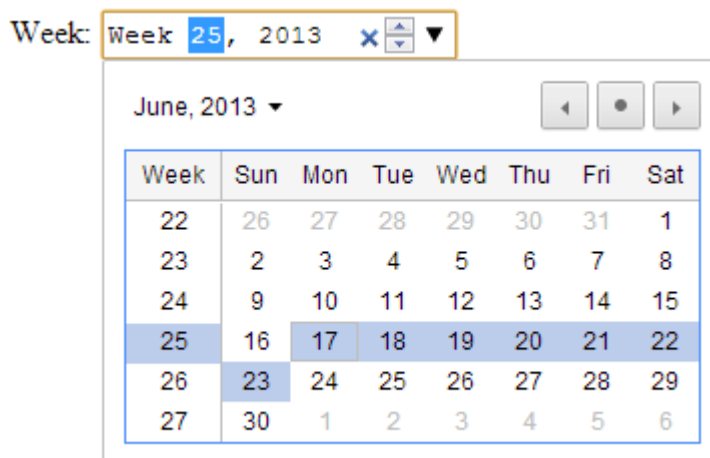
`<input type="url" name="homepage">`



Hình 2.22. Ví dụ kiểu input: url

Kiểu input: week – cho phép chọn một tuần nào đó của một năm

Week: `<input type="week" name="year_week">`



Hình 2.23. Ví dụ kiểu input: week

Những thành phần mới của Form

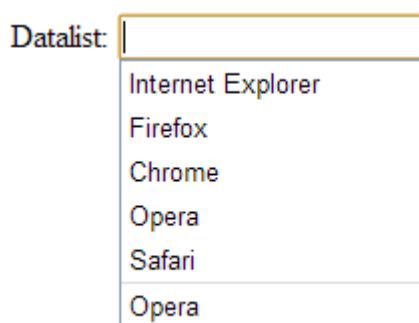
HTML5 có ba thành phần form mới:

- `<datalist>`
- `<keygen>`
- `<output>`

Thành phần `<datalist>` được sử dụng để cung cấp đặc điểm “autocomplete” cho thành phần `<input>`. Người dùng sẽ nhìn thấy một danh sách thả xuống các tùy chọn đã được tạo trước sẵn. Ví dụ:

```
<input list="browsers">
Datalist:
<datalist id="browsers">
  <option value="Internet Explorer">
  <option value="Firefox">
  <option value="Chrome">
  <option value="Opera">
  <option value="Safari">
</datalist>
```

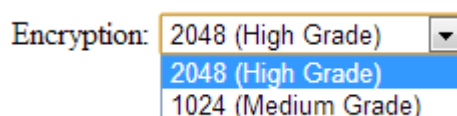
Kết quả:



Hình 2.24. Ví dụ thành phần `<datalist>`

Thành phần `<keygen>` - cung cấp một cách an toàn để xác thực người dùng. Khi form được gửi đi, 2 khóa sẽ được tạo ra, một là private và một là public. Khóa private được lưu tại máy khách và khóa public được gửi tới server. Khóa public có thể được dùng để tạo một chứng chỉ khách để xác thực người dùng trong tương lai.

Encryption: `<keygen name="security">`

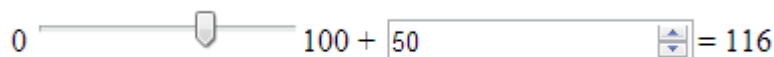


Hình 2.25. Ví dụ thành phần `<keygen>`

Thành phần <output> - hiển thị kết quả tính toán, giống như tính toán trên các script. Ví dụ sau thực hiện tính tổng hai số a, b được lấy từ 2 kiểu input range:

```
<form oninput="x.value=parseInt(a.value)+parseInt(b.value)">0
<input type="range" id="a" value="50">100 +
<input type="number" id="b" value="50">=
<output name="x" for="a b"></output>
</form>
```

Kết quả:



Hình 2.26. Ví dụ thành phần <output>

Những thuộc tính mới của form

Bảng 2.9. Các thuộc tính mới của <form>

Thuộc tính	Mô tả
autocomplete	Kích hoạt chức năng autocomplete cho form. Có giá trị “on” hoặc “off”
novalidate	Khi được sử dụng sẽ không kiểm tra giá trị được nhập vào khi gửi đi

Bảng 2.10. Các thuộc tính mới của <input>

Thuộc tính	Mô tả
autocomplete	Kích hoạt chức năng autocomplete cho <input>. Có giá trị “on” hoặc “off”
autofocus	Tự động focus con trỏ vào vị trí <input> nó được chỉ định.
form	Xác định một hoặc nhiều <input> thuộc form nào đó
formaction	Được sử dụng cho kiểu submit và image. Xác định URL sẽ xử lý form khi gửi đi
formenctype	Được sử dụng cho kiểu submit và image. Xác định dữ liệu form sẽ được mã hóa như thế nào khi gửi đi (chỉ sử dụng với form có method=”post”)
formmethod	Được sử dụng cho kiểu submit và image. Xác định phương thức HTTP sẽ gửi dữ liệu form tới URL action
formnovalidate	Được sử dụng với kiểu submit. Không kiểm tra tính hợp lệ của <input>
formtarget	Được sử dụng cho kiểu submit và image. Xác định một tên hay

	một từ khóa cho biết nơi hiển thị các response nhận được sau khi gửi form đi
height và width	Chỉ được sử dụng cho kiểu image. Xác định chiều cao và chiều rộng của <input>
list	Thuộc tính này sẽ sử dụng một danh sách được tạo ra từ <datalist> đưa vào <input>
min và max	Xác định giá trị tối thiểu và tối đa của <input>
multiple	Cho phép người dùng được phép nhập nhiều hơn một giá trị trong <input>
pattern	Cho phép người dùng đặt một mẫu có thể được kết hợp với các giá trị được nhập vào bởi người sử dụng trước khi được gửi đi.
placeholder	Hiển thị một gợi ý ngắn miêu tả giá trị nên được nhập vào một <input>
required	Xác định trường <input> đó phải được điền trước khi gửi đi
step	Được sử dụng cho các kiểu: number, range, date, datetime, datetime-local, month, time và week. Xác định bước nhảy khi tăng hoặc giảm giá trị.

2.3.9 Drag-and-Drop

Tính năng kéo-thả là một tính năng rất phổ biến. Nó là khi ta kéo một đối tượng thả nó tới một vị trí khác. Và trong HTML5, kéo-thả là một chuẩn mới và bất kỳ thành phần nào cũng có thể kéo được (draggable).

Đoạn mã sau là ví dụ đơn giản về kéo thả:

```
<!DOCTYPE HTML>
<html>
<head>
<script>
function allowDrop(ev)
{
ev.preventDefault();
}

function drag(ev)
{
ev.dataTransfer.setData("Text",ev.target.id);
}

function drop(ev)
```

```

{
ev.preventDefault();
var data=ev.dataTransfer.getData("Text");
ev.target.appendChild(document.getElementById(data));
}
</script>
</head>
<body>
<div id="div1" ondrop="drop(event)"
ondragover="allowDrop(event)"></div>


</body>
</html>

```

Thêm giao diện bằng CSS ta được một giao diện kéo-thả như Hình 2.27:



Hình 2.27. Ví dụ drag-and-drop

Làm cho một thành phần có thể kéo được

Từ ví dụ đoạn mã tạo ra giao diện kéo thả như hình trên, để làm cho một thành phần nào đó có thể kéo được, ta thiết lập thuộc tính `draggable="true"`. Theo ví dụ thì thiết lập thành phần `` là ảnh logo có thể kéo được;

```
<img draggable="true">
```

Sử dụng thuộc tính ondragstart và phương thức setData()

Sau khi thiết lập thành phần draggable, ta phải xác định những gì sẽ xảy ra khi thành phần đó được kéo. Trong ví dụ trên, thuộc tính ondragstart gọi hàm drag(event) để xác định dữ liệu gì được kéo.

Phương thức dataTransfer.setData() sẽ đặt kiểu dữ liệu và giá trị của dữ liệu được kéo. Trong trường hợp này, kiểu dữ liệu là “Text” và giá trị là id của thành phần(“drag1”):

```
function drag(ev)
{
    ev.dataTransfer.setData("Text",ev.target.id);
}
```

Thuộc tính ondragover

Thuộc tính ondragover xác định nơi dữ liệu được kéo sẽ thả xuống. Mặc định các thành phần và dữ liệu không thể được thả vào những thành phần khác. Để cho phép thả như vậy, ta phải chặn thiết lập mặc định đó bằng việc gọi phương thức event.preventDefault() cho thuộc tính ondragover:

```
event.preventDefault();
```

Thuộc tính ondrop

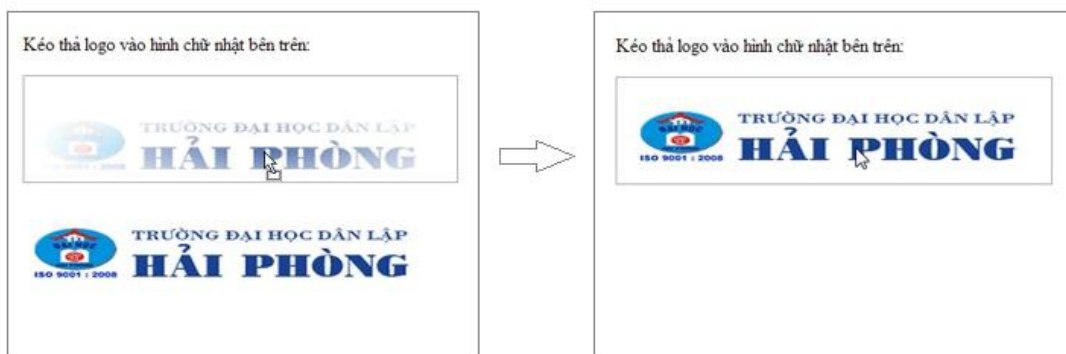
Khi dữ liệu được thả, một sự kiện thả sẽ xảy ra. Trong ví dụ trên, thuộc tính ondrop gọi hàm drop(event):

```
function drop(ev)
{
    ev.preventDefault();
    var data=ev.dataTransfer.getData("Text");
    ev.target.appendChild(document.getElementById(data));
}
```

Giải thích:

- Đầu tiên gọi phương thức preventDefault() để chặn lại thiết lập mặc định của trình duyệt để xử lý dữ liệu (mặc định là mở như một liên kết khi thả)
- Lấy dữ liệu được kéo với phương thức dataTransfer.getData(“Text”). Phương thức này sẽ trả lại bất kỳ dữ liệu gì mà được đặt cùng kiểu trong phương thức setData().
- Xác định dữ liệu được kéo là id (“drag1”)
- Thêm thành phần được kéo vào thành phần thả.

Thực hiện kéo thả ta được như Hình 2.28:



Hình 2.28. Thực hiện kéo thả

2.3.10 Web Workers API

Web Worker là gì?

Khi thực thi một script trong một trang HTML thì trang sẽ không thực hiện giao tiếp với server cho đến khi script hoàn thành. Một web worker là một JS chạy dưới nền Web, độc lập với các script khác và không làm ảnh hưởng đến hiệu suất của trang. Ta có thể làm mọi thứ như click chuột, quét chọn,.. trong khi web worker đang chạy.

Để biết rõ hơn về web worker ta sẽ làm một ví dụ đơn giản nhảy số tăng dần:

Kiểm tra trình duyệt có hỗ trợ Web Worker

Trước khi tạo một web worker, ta phải kiểm tra xem trình duyệt đang dùng có hỗ trợ không:

```
if (typeof(Worker) !== "undefined")
{
    // Có hỗ trợ
    // Viết code.....
}
else
{
    // Không hỗ trợ..
}
```

Tạo một file Web Worker

Chúng ta tạo ra một script đếm. Script này được viết trong file demo_workers.js:

```

var i=0;
function timedCount()
{
    i=i+1;
    postMessage(i);
    setTimeout("timedCount()",500);
}
timedCount();

```

Phần quan trọng của đoạn code trên chính là phương thức `postMessage()`, nó dùng để gửi thông tin đến trang HTML.

Tạo một đối tượng Web Worker

Chúng ta đã có một file web worker, giờ ta cần gọi nó ra từ trang HTML. Đoạn mã sau kiểm tra worker có hay chưa, nếu chưa, tạo web worker mới và chạy code trong file “demo_workers.js”:

```

if (typeof(w)=="undefined")
{
    w=new Worker("demo_workers.js");
}

```

Sau đó bạn có thể gửi và nhận thông tin từ web worker bằng đoạn mã sau:

```

w.onmessage=function(event){
    document.getElementById("result").innerHTML=event.data;
};

```

Khi web worker trả lại cho ta thông tin, đoạn mã trong sự kiện được thực thi. Dữ liệu từ web worker gửi đến được lưu trong `event.data`.

Hủy Web Worker

Khi đối tượng web worker được tạo ra, nó sẽ tiếp tục hoạt động cho tới khi ta dùng phương thức `terminate()` để hủy nó:

```

w.terminate();

```

Ví dụ hoàn chỉnh

```

<!DOCTYPE html>
<html>
<body>
<p>Count numbers: <output id="result"></output></p>
<button onclick="startWorker()">Start Worker</button>
<button onclick="stopWorker()">Stop Worker</button>
<br>
<br>

```

```
<script>
var w;
function startWorker()
{
  if (typeof(Worker) !== "undefined")
  {
    if (typeof(w) === "undefined")
    {
      w = new Worker("demo_workers.js");
    }
    w.onmessage = function (event) {
      document.getElementById("result").innerHTML = event.data;
    };
  }
  else
  {
    document.getElementById("result").innerHTML = "Trình duyệt
    không hỗ trợ Web Worker.";
  }
}

function stopWorker()
{
  w.terminate();
}
</script>
</body>
</html>
```

Giao diện như Hình 2.29:

Count numbers: 15

Start Worker

Stop Worker

Hình 2.29. Ví dụ sử dụng Web Worker

2.3.11 Storage APIs

Web Storage là gì?

Với HTML5, các trang web có thể lưu trữ dữ liệu tại máy khách trong trình duyệt của người dùng. Trước đây, điều này được thực hiện với *cookies*. Tuy nhiên với *Web Storage* thì an toàn và nhanh hơn nhiều.

Sự xuất hiện của *Web Storage* là một điểm nhấn cho việc ra đời các ứng dụng web có khả năng tương tác và nạp dữ liệu tức thì trên trình duyệt. Một hiệu quả nữa là dung lượng truyền tải qua mạng có thể được giảm thiểu đáng kể. Với những ứng dụng web có cơ sở dữ liệu nhỏ gọn, lập trình viên có thể thực hiện việc cache “âm thầm” cơ sở dữ liệu xuống client và sau đó người dùng có thể thoải mái tra cứu mà không cần request đến server

Dữ liệu được lưu trữ trong một cặp *key/value*, và một trang web chỉ có thể truy cập vào dữ liệu do chính nó lưu trữ.

localStorage and sessionStorage

Có hai đối tượng mới cho lưu trữ dữ liệu trên máy client:

- *localStorage* – lưu trữ dữ liệu không giới hạn thời gian
- *sessionStorage* – lưu trữ dữ liệu cho một phiên làm việc *session*

Trước khi sử dụng web storage, ta phải kiểm tra trình duyệt có hỗ trợ *localStorage* và *sessionStorage*:

```
if (typeof (Storage) !== "undefined")
{
    // localStorage and sessionStorage được hỗ trợ!
    // Code .....
}
else
{
    // Trình duyệt không hỗ trợ..
}
```

Đối tượng localStorage

Đối tượng *localStorage* lưu trữ dữ liệu không giới hạn thời gian, dữ liệu sẽ không bị xóa khi trình duyệt đóng.

Ví dụ:

```
localStorage.school="HPU";
document.getElementById("result").innerHTML="School: "
+ localStorage.school;
```

Giải thích:

- Tạo một cặp *key/value* với *key*="school" và *value*="HPU"
- Nhận giá trị của khoa "school" và chèn vào thành phần có *id*="result"

Đối tượng sessionStorage

Đối tượng sessionStorage sử dụng tương đương với localStorage nhưng chỉ khác một điều là dữ liệu của nó chỉ lưu trữ cho một phiên làm việc. Dữ liệu sẽ mất khi người dùng đóng trình duyệt.

Ví dụ: Đếm số lần người dùng kích chuột vào 1 nút, trong phiên làm việc hiện tại:

```
if (sessionStorage.clickcount)
{
    sessionStorage.clickcount=Number(sessionStorage.clickcount)+1
;
}
else
{
    sessionStorage.clickcount=1;
}
document.getElementById("result").innerHTML="Bạn đã kích " +
sessionStorage.clickcount + " lần trong phiên làm việc này.";
```

2.3.12 Tạo Offline Web Applications

Với HTML5 thật dễ dàng tạo những ứng dụng web offline mà không cần đến kết nối internet. Application Cache có 3 lợi thế sau:

- Duyệt web offline – người dùng vẫn có thể sử dụng được ứng dụng mà không cần đến kết nối mạng internet.
- Tốc độ - cache có tốc độ nhanh
- Giảm tải cho server - Trình duyệt chỉ cần tải những cập nhật từ server.

HTML5 Cache Manifest

Ví dụ: Ví dụ sau là một tài liệu HTML với cache manifest (cho duyệt web offline):

```
<!DOCTYPE HTML>
<html manifest="demo.appcache">

<body>
The content of the document.....
</body>

</html>
```

Cơ bản về Cache Manifest

Để kích hoạt application cache, ta chèn thuộc tính manifest vào thẻ <html>:

```
<!DOCTYPE HTML>
<html manifest="demo.appcache">
...
</html>
```

Nếu một trang có thuộc tính manifest sẽ được lưu lại trong bộ nhớ cache khi người dùng truy cập đến nó. Nếu như trang web không thiết lập thuộc tính manifest thì nó sẽ không được lưu cache trừ khi trong file manifest có dòng lệnh thiết lập trực tiếp cho nó.

Phần mở rộng của file manifest là “.appcache”. Để sử dụng file manifest thì cần phải thiết lập máy chủ đúng kiểu MIME-type là “text/cache-manifest”.

Manifest File

File manifest là một file văn bản đơn giản, nó nói cho trình duyệt biết phải cache cái gì và không cache gì.

File manifest có 3 phần:

- **CACHE MANIFEST** – Danh sách những file sẽ được cache sau lần truy cập đầu tiên
- **NETWORK** – Danh sách những file bắt buộc phải có kết nối internet và không bao giờ lưu cache.
- **FALLBACK** – Liệt kê danh sách những trang dự phòng nếu một trang không thể truy cập được.

Bảng 2.11. Mô tả cấu trúc một file manifest

Phần	Khai báo	Mô tả
CACHE MANIFEST	CACHE MANIFEST /theme.css /logo.gif /main.js ...	- Dòng đầu tiên là bắt buộc khai báo từ khóa: “CACHE MANIFEST”. - Những dòng bên trái có quy định 3 file là css, js, gif sẽ được tải về máy từ website.
NETWORK	NETWORK login.php ...	- Những phần sau dòng NETWORK yêu cầu phải có kết nối internet, ví dụ như file login.php không được lưu cache và không bao giờ được làm việc offline.

	NETWORK *	-Dấu (*) yêu cầu tất cả các file đều phải có kết nối internet.
FALLBACK	FALLBACK /html/ /offline.html	- Xác định file “offline.html” sẽ được dùng trong trường hợp không có kết nối internet.

Cập nhật Cache

Một khi ứng dụng được cache nó vẫn được cache cho tới khi xảy ra một trong các điều sau:

- Người dùng xóa cache của trình duyệt
- File manifest được chỉnh sửa
- Bộ nhớ cache ứng dụng được cập nhật theo chương trình

Ví dụ: Một file cache manifest:

```
CACHE MANIFEST
# 2012-02-21 v1.0.0
/theme.css
/logo.gif
/main.js
```

```
NETWORK:
login.asp
```

```
FALLBACK:
/html/ /offline.html
```

Dòng bắt đầu với ký tự “#” là một dòng chú thích. Một ứng dụng cache chỉ cập nhật khi file manifest bị thay đổi. Nếu bạn sửa một hình ảnh hoặc một hàm JS, những thay đổi đó không bị ảnh hưởng đến cache. Cập nhật ngày, phiên bản vào dòng ghi chú như một cách để trình duyệt cache trang web.

Ghi chú

Bạn phải cẩn thận với những gì bạn cache. Một file bị cache thì trình duyệt sẽ tiếp tục hiển thị phiên bản cache trước đó, cho dù bạn thay đổi file trên máy chủ thì máy trạm vẫn hiện bản cache cũ. Do đó, mồn cache được cập nhật thì bạn phải thay đổi file manifest.

Lưu ý: Trình duyệt có thể có những giới hạn dung lượng cache khác nhau (một vài trình duyệt giới hạn dung lượng cache là 5MB cho một website).

CHƯƠNG 3: XÂY DỰNG ỨNG DỤNG

3.1 Giới thiệu ứng dụng

Từ những công nghệ hấp dẫn của HTML5 trình bày ở trên, em đã tạo được 2 ứng dụng sau:

- **Website chia sẻ video :** Website cho phép người dùng tải lên những video của họ và chia sẻ với người khác qua đường link của video đã tải lên đó.
- **Xác định vị trí:** Ứng dụng giúp xác định vị trí của người dùng.

Những ứng dụng này được xây dựng sử dụng những thành phần sau của HTML5:

- Thành phần <video> được sử dụng để phát các clip trên website
- Tính năng drag-and-drop của HTML5 dùng để kéo-thả file rồi upload
- XMLHttpRequest mức 2 được dùng để tải file video lên máy chủ
- Các thuộc tính mới của form, ứng dụng trong form nhập thông tin video khi upload
- Tính năng Geolocation để xác định vị trí.

3.2 Yêu cầu phần cứng

Ứng dụng **Website chia sẻ video** cần một máy chủ có dung lượng ổ cứng lớn để lưu trữ dữ liệu video người dùng tải lên và một đường truyền tốt giúp người dùng upload cũng như download video nhanh.

3.3 Yêu cầu phần mềm

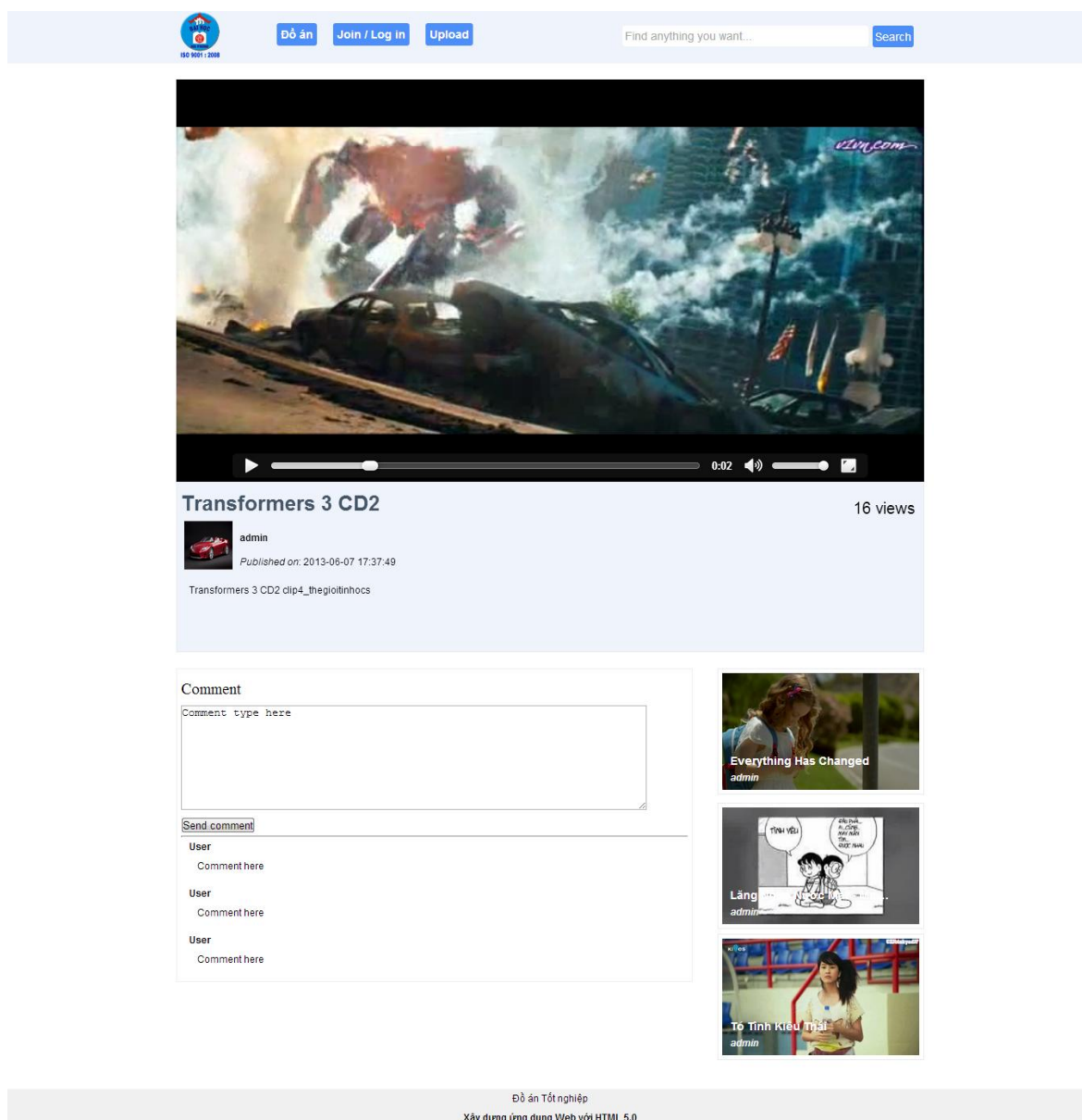
- Hệ điều hành: Windows / Linux
- Máy chủ web: Apache
- Ngôn ngữ xây dựng: HTML5, PHP, JavaScript, CSS
- Hệ quản trị cơ sở dữ liệu: MySQL

3.4 Một số giao diện chính

Sau đây là một số giao diện chính của các ứng dụng.

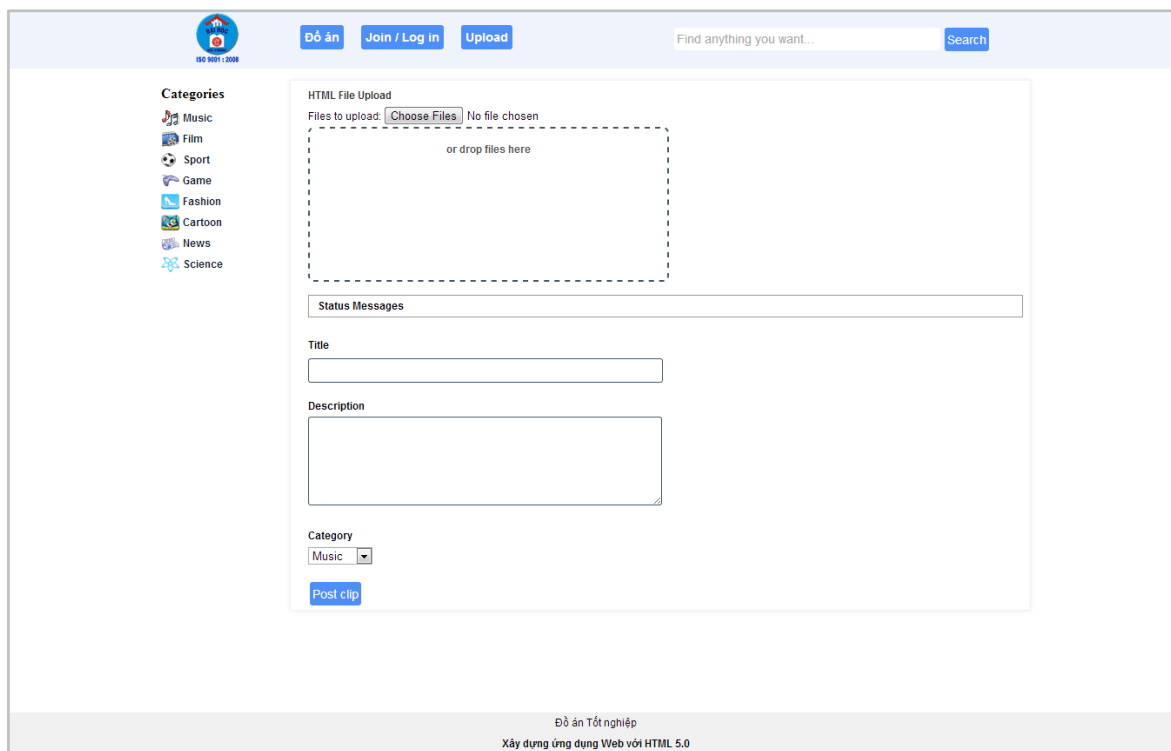
3.4.1 Website chia sẻ video và Xác định vị trí

Giao diện xem video (Hình 3.1) sử dụng thành phần <video> của HTML5



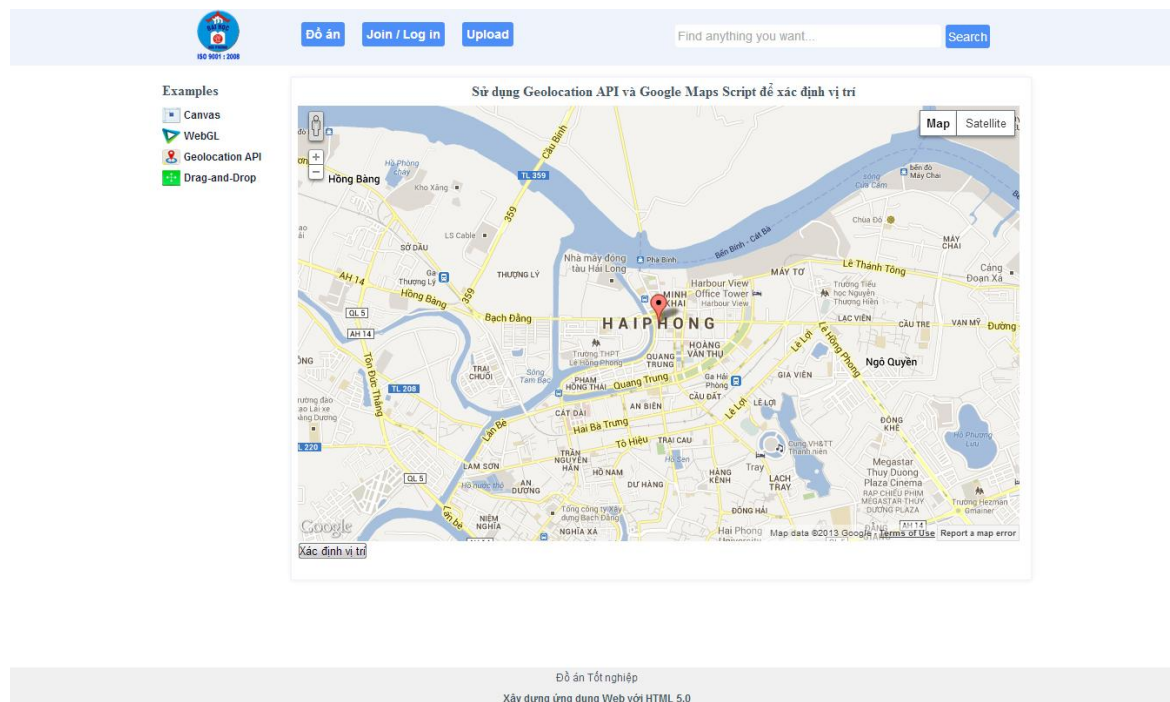
Hình 3.1. Giao diện trang xem video

Giao diện trang upload (Hình 3.2) sử dụng tính năng drag-and-drop, XMLHttpRequest mức 2 và các thuộc tính mới của form trong HTML5.



Hình 3.2. Giao diện trang upload

Giao diện ứng dụng Xác định vị trí (Hình 3.3) sử dụng tính năng Geolocation của HTML5 và API Google Maps Script để hiển thị vị trí người dùng.

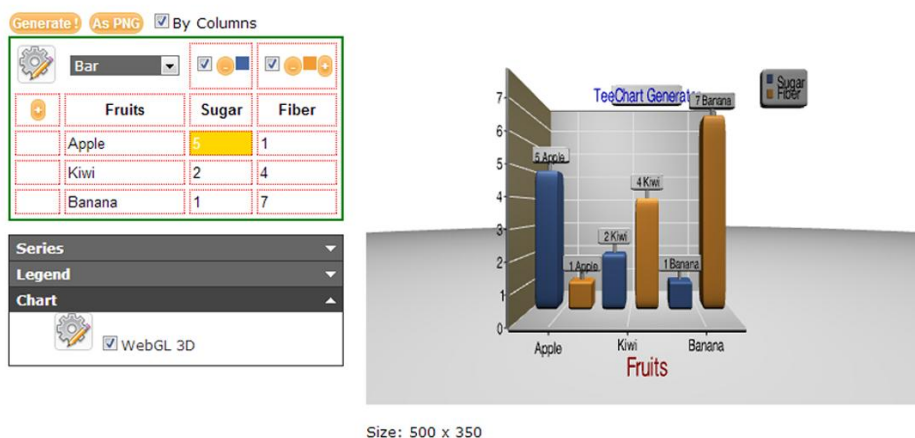


Hình 3.3. Giao diện ứng dụng Xác định vị trí

3.4.2 Một số ứng dụng khác

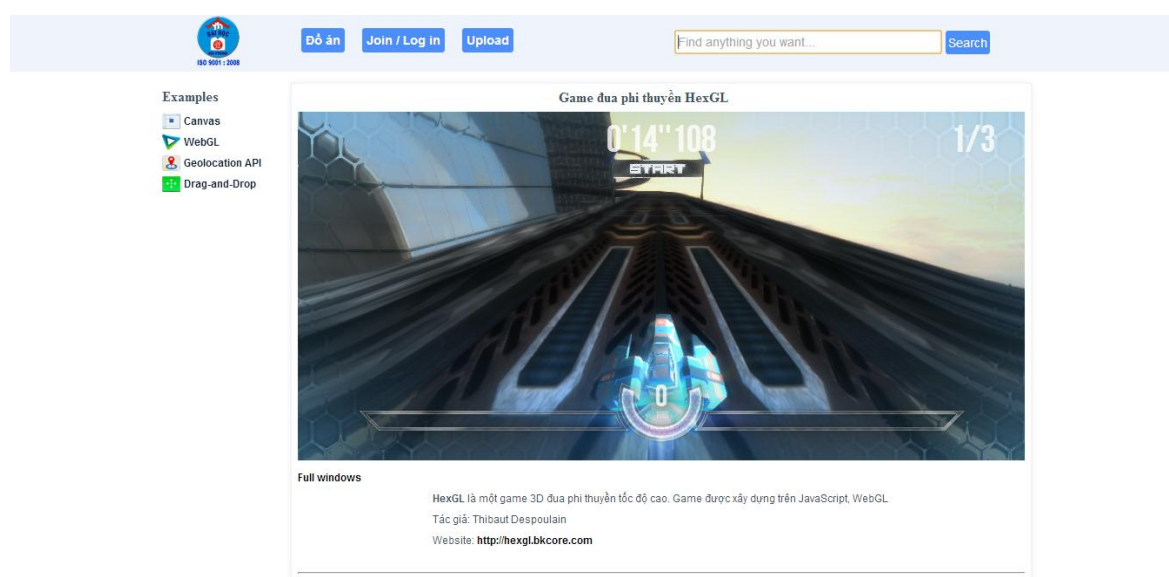
HTML5 được đánh ra cao về trình diễn đồ họa 2D cũng như 3D ngay trên trình duyệt, sau đây là một số ứng dụng khác em sưu tầm được từ nhiều nguồn khác nhau mà thể hiện rất rõ khả năng này của HTML5.

Ứng dụng vẽ biểu đồ 3D TeeChart (Hình 3.4) của Steema Software, sử dụng thành phần <canvas> và WebGL.



Hình 3.4. Ứng dụng vẽ đồ thị 3D TeeChart

Trò chơi đua phi thuyền HexGL (Hình 3.5) của Thibaut Despoulain sử dụng <canvas>, WebGL với thư viện three.js.



Hình 3.5. Trò chơi đua phi thuyền HexGL

KẾT LUẬN

Qua quá trình thực hiện đồ án tốt nghiệp với đề tài “**Xây dựng ứng dụng Web với HTML 5.0**” em thấy mình đã đạt được một số kết quả nhất định và đã thu được một số kết quả sau:

- Nắm được những kiến thức cơ bản về Web và HTML. Bên cạnh đó là những công nghệ mới trong HTML5 cũng với CSS, JavaScript.
- Rút ra được một số kinh nghiệm và bài học cơ bản về cách làm việc khoa học, chủ động nghiên cứu các công nghệ mới. Có thể áp dụng kiến thức đã được học vào thực tiễn, đồng thời thu thập được rất nhiều những kiến thức khác từ quá trình làm đồ án.
- Vận dụng những kiến thức đã được học vào việc nghiên cứu áp dụng cho những vấn đề mới gặp phải trong quá trình làm việc một cách hiệu quả. Thử nghiệm ngôn ngữ PHP để tạo ra ứng dụng minh họa cho đề tài.

Đối với HTML5 và những công nghệ mà nó mang đến, mặc dù vẫn chưa được hoàn chỉnh nhưng sẽ trở thành một chuẩn mới cho thế giới web trong tương lai. Giờ đây, một số công nghệ đó giúp ta chỉ cần một trình duyệt hỗ trợ HTML5 là có thể làm mọi điều trên đó mà không cần một plugin của hãng thứ ba nào (như Flash, Silverlight,...).

TÀI LIỆU THAM KHẢO

- [1]. **Flanagan, David.***JavaScript: The Definitive Guide, 6th Edition*. s.l. : O'Reilly Media, 2011. p. 1100.
- [2]. **Hogan, Brian P.***HTML5 and CSS3: Develop with Tomorrow's Standards Today*. s.l. : Pragmatic Bookshelf, 2011. p. 280. 978-1934356685.
- [3]. **Lubbers, Peter, Salim, Frank and Albers, Brian.***Pro HTML5 Programming, 2nd Edition*. s.l. : Apress, 2011. p. 352. 978-1430238645.
- [4]. **Lawson, Bruce; Sharp, Remy.***Introducing HTML5 (2nd Edition)*. s.l. : New Riders, 2011. p. 312. 978-0321784421.
- [5]. W3Schools. [Online] <http://www.w3schools.com/html>.
- [6]. IBM developerWorks. [Online]
<http://www.ibm.com/developerworks/vn/library/wa-html5fundamentals/>.
- [7]. Wikipedia - Web 2.0. [Online] http://en.wikipedia.org/wiki/Web_2.0.
- [8]. Wikipedia - HTML5. [Online] <http://en.wikipedia.org/wiki/HTML5>.