

**ỦY BAN NHÂN DÂN THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG CAO ĐẲNG KINH TẾ KỸ THUẬT
THÀNH PHỐ HỒ CHÍ MINH**

ISO



ISO 9001 - 2008

GIÁO TRÌNH

MÔN HỌC: XÂY DỰNG ỨNG DỤNG WEB

NGHỀ: HỆ THỐNG THÔNG TIN

TRÌNH ĐỘ: CAO ĐẲNG

(Ban hành kèm theo Quyết định số: /QĐ-CDKTKT

ngày tháng năm 2020 của Hiệu trưởng

Trường Cao đẳng Kinh tế - Kỹ thuật Thành phố Hồ Chí Minh)

Thành phố Hồ Chí Minh, năm 2020

**ỦY BAN NHÂN DÂN THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG CAO ĐẲNG KINH TẾ KỸ THUẬT
THÀNH PHỐ HỒ CHÍ MINH**



**GIÁO TRÌNH
MÔN HỌC: XÂY DỰNG ỨNG DỤNG WEB
NGHỀ: HỆ THỐNG THÔNG TIN
TRÌNH ĐỘ: CAO ĐẲNG**

THÔNG TIN CHỦ NHIỆM ĐỀ TÀI

Họ tên: Nguyễn Gia Quang Đăng

Học vị: Thạc sỹ

Đơn vị: Khoa Công nghệ thông tin

Email: nguyengiaquangdang@hotec.edu.vn

TRƯỞNG KHOA

**TỔ TRƯỞNG
BỘ MÔN**

**CHỦ NHIỆM
ĐỀ TÀI**

Lê Như Dzi

Võ Đào Thị Hồng Tuyết

Nguyễn Gia Quang Đăng

**HIỆU TRƯỞNG
DUYỆT**

Tháng 10 năm 2020

TUYÊN BỐ BẢN QUYỀN

Tài liệu này thuộc loại sách giáo trình nên các nguồn thông tin có thể được phép dùng nguyên bản hoặc trích dùng cho các mục đích về đào tạo và tham khảo.

Mọi mục đích khác mang tính lệch lạc hoặc sử dụng với mục đích kinh doanh thiếu lành mạnh sẽ bị nghiêm cấm.

Tailieu.vn

LỜI GIỚI THIỆU

Giáo trình này được biên soạn dựa trên chương trình chi tiết môn học bậc cao đẳng chuyên ngành **Hệ thống thông tin** Khoa Công nghệ thông tin của Trường Cao đẳng Kinh tế Kỹ thuật Thành Phố Hồ Chí Minh.

Ngày nay, do sự phát triển không ngừng của mạng internet toàn cầu, HTML cũng ngày càng trở nên phức tạp và hoàn thiện hơn để đáp ứng được những yêu cầu mới nảy sinh trong quá trình phát triển đó (như âm thanh, hình ảnh động, v.v...). Người ta gọi đó là những phiên bản của HTML và đánh số để biểu thị. HTML 2, HTML 2+, HTML 3, HTML 5,... là để chỉ những phiên bản sau này.

Một trong những điểm mạnh của HTML là một văn bản bất kỳ nếu tuân thủ tiêu chuẩn HTML đều có thể hiện lên màn hình hay in ra, tóm lại là hiểu được, bởi bất kỳ loại phần mềm hay máy tính nào mà người đọc có, không phân biệt trình duyệt nào (NetScape trên Windows hay Lynx trên UNIX, thậm chí cho người khiếm thị bằng phần mềm đặc biệt).

Đây là quyển giáo trình được biên soạn lần thứ nhất cho môn học này tại khoa Công nghệ thông tin của nhà trường. Nhằm cung cấp kiến thức nền tảng về ngôn ngữ HTML, Javascript, C#, giúp học sinh sinh viên nắm vững và vận dụng các kỹ thuật phổ biến viết ứng dụng trên web chạy trực tuyến trên mạng.

Từ đó, sinh viên có thể tự học các kiến thức chuyên sâu hơn. Trong tài liệu này tác giả sử dụng phương pháp lập trình trên ngôn ngữ C# truy xuất trên môi trường web. Qua đó, giúp sinh viên nắm bắt kiến thức và kỹ năng thực hành cơ bản để vận dụng viết được các ứng dụng trong thực tiễn.

Trong quá trình biên soạn chắc chắn giáo trình sẽ còn nhiều thiếu sót và hạn chế. Rất mong nhận được sự đóng góp ý kiến quý báu của học sinh sinh viên và các bạn đọc để giáo trình ngày một hoàn thiện hơn.

Thành phố Hồ Chí Minh, ngày.....tháng 10 năm 2020

Tác giả biên soạn
Nguyễn Gia Quang Đăng

MỤC LỤC

LỜI GIỚI THIỆU	1
GIÁO TRÌNH MÔN HỌC.....	6
Chương 1: Giới thiệu chung.....	7
1.1. Nền tảng của Web ngôn ngữ HTML	7
1.1.1. Cơ bản về ngôn ngữ HTML	7
1.1.2. Định dạng Text	8
1.1.3. Liên kết các trang web (Link)	10
1.1.4. Danh sách (List).....	11
1.2. Các giao thức sử dụng cho Web.....	12
1.3. Các mô hình ứng dụng Web.....	13
1.3.1. Tất cả trong một Server.....	13
1.3.2. Tách riêng Database Server.....	14
1.3.3. Sử dụng Load Balancer (Reverse Proxy).....	15
1.3.4. Dùng HTTP Accelerator (Caching Reverse Proxy)	15
1.3.5. Cấu hình Master-Slave Database Replication	16
Chương 2: Định dạng trang web với CSS	18
2.1. Mô hình 3 lớp trong thiết kế Web	18
2.2. Các loại CSS.....	19
2.3. CSS In-line, out-line	20
2.3.1. Các thành phần CSS.....	20
2.3.2. Nhóm nhiều đối tượng	21
2.3.3. Thuộc tính Class	21
2.3.4. Thuộc tính ID.....	22
2.3.5. Ghi chú trong CSS	23
2.3.6. Sử dụng CSS trong trang HTML.....	23
2.3.7. Các ví dụ áp dụng	25
2.4. Bài tập áp dụng	28
Chương 3: Ngôn ngữ JavaScript	30
3.1. Giới thiệu về JavaScript	30
3.2. Sử dụng JavaScript trong trang HTML	30
3.2.1. Cú pháp cơ bản của lệnh	31

3.2.2. Hiển thị một dòng văn bản	31
3.2.3. Hiển thị hộp thoại thông báo –Lệnh alert()	32
3.2.4. Giao tiếp với người sử dụng – Lệnh prompt()	33
3.2.5. Hỏi đáp người sử dụng – Lệnh confirm()	34
3.3. Biến, kiểu dữ liệu, biểu thức, toán tử.....	35
3.3.1. Biến	35
3.3.2. Kiểu dữ liệu	35
3.3.3. Lệnh, khối lệnh trong JavaScript	36
3.3.4. Toán tử & Biểu thức trong JavaScript	37
3.4. Câu lệnh điều kiện, lệnh lặp for.....	39
3.4.1. Cấu trúc lập trình rẽ nhánh (Điều Kiện)	39
3.4.2. Cấu trúc lặp.....	40
3.5. Câu lệnh while, switch, break, continue	41
3.6. Biến mảng, hàm	43
3.6.1. Mảng.....	43
3.6.2. Hàm	45
3.7. DOM	50
3.8. Sự kiện.....	55
3.8.1. Khái niệm sự kiện và xử lý sự kiện	55
3.8.2. Một số sự kiện trong JavaScript	57
3.8.3. Các sự kiện có sẵn của một số đối tượng	57
3.9. Bài tập áp dụng	59
Chương 4: Kịch bản trình chủ.....	60
4.1. Giới thiệu về ngôn ngữ lập trình Web động.....	60
4.2. Cài đặt cấu hình Web Server IIS	60
4.3. Giới thiệu Visual Studio.....	63
4.3.1. Tìm hiểu về .Net Platform	63
4.3.2. Tìm hiểu về .Net Framework.....	64
4.3.3. Tìm hiểu về ASP.Net	68
4.3.4. Những ưu điểm của ASP.Net	69
4.3.5. Quá trình xử lý tập tin ASPX	70
4.4. Tạo ứng dụng Web trên Visual Studio	70
4.4.1. Khởi động ASP.NET	70
4.4.2. Phân loại tập tin trong ASP.Net.....	73

4.5. Các thành phần giao diện VS	74
4.5.1. Solution Explorer.....	74
4.5.2. Property/Toolbox	75
4.6. Server Controls - Các điều khiển chuẩn	76
4.6.1. HTML Control.....	76
4.6.2. ASP.Net Web Control.....	77
4.7. Điều khiển GridView	91
4.7.1. Các thao tác định dạng lưới.....	91
4.7.2. Xử lý sắp xếp.....	97
4.7.3. Xử lý phân trang	100
4.7.4. Tùy biến các cột.....	101
4.7.5. Cập nhật dữ liệu trực tiếp trên lưới.....	104
4.8. Điều khiển DataList	110
4.8.1. Sử dụng DataList để hiển thị dữ liệu	110
4.8.2. Cập nhật dữ liệu với DataList.....	114
4.9. Điều khiển Repeater.....	121
4.10. Bài tập áp dụng	125
Chương 5: Xây dựng và quản lý ứng dụng.....	126
5.1. Đối tượng Request	126
5.2. Đối tượng Response.....	127
5.2.1. Đối tượng Response	127
5.2.2. Ví dụ xử lý cho phép người dùng download file.....	128
5.3. Session.....	129
5.3.1. Thuộc tính & Phương thức.....	130
5.3.2. Sử dụng biến toàn cục với Session	131
5.4. Application	132
5.4.1. Sử dụng biến Application.....	132
5.4.2. Duyệt qua tập hợp biến chứa trong Application.....	132
5.5. Cookies.....	132
5.5.1. Giới thiệu.....	132
5.5.2. Làm việc với Cookies	133
5.6. Global.asax	134
5.6.1. Cấu trúc tập tin Global.asax	134
5.6.2. Các sự kiện trong tập tin Global.asax	135

5.7. Web.config	137
5.7.1 Cấu trúc tập tin web.config	137
5.7.2. Các cấu hình mặc định	139
5.8. Bài tập áp dụng	141
Chương 6: Xây dựng các thành phần truy xuất dữ liệu.....	142
6.1. Tìm hiểu về ADO.NET	142
6.2. Giới thiệu.....	142
6.3. Kiến trúc ADO.NET	143
6.4. Minh họa tạo kết nối cơ sở dữ liệu	144
6.5. Các đối tượng trong ADO.NET	144
6.6. Đối tượng Connection.....	146
6.7. Đối tượng Command.....	147
6.8. Đối tượng DataReader	150
6.9. Đối tượng DataAdapter.....	150
6.10. Đối tượng DataSet	152
6.11. Đối tượng DataTable.....	157
6.12. Bài tập	160
Chương 7: Web Services	161
7.1. Tìm hiểu về Web services	161
7.2. Xây dựng Web services	163
7.2.1. Tạo Web Services trong Visual studio .Net.....	163
7.2.2. Kiểm tra Web Service	165
7.3. Sử dụng Web services.....	167
7.3.1. Sử dụng Web Services do người dùng xây dựng	167
7.3.2. Sử dụng Web Services được cung cấp miễn phí trên mạng.....	168
7.4. Xây dựng Web services truy xuất dữ liệu.....	169
7.5. Bài tập	171
TÀI LIỆU THAM KHẢO	172
DANH MỤC HÌNH ẢNH	173
DANH MỤC BẢNG BIỂU	175

GIÁO TRÌNH MÔN HỌC

Tên môn học: **Xây dựng ứng dụng web**

Mã môn học: **MH3101346**

Vị trí, tính chất, ý nghĩa và vai trò của môn học:

- Vị trí: là môn học chuyên ngành, được bố trí sau môn học Thiết kế Web, học kỳ 4.
- Tính chất: là môn học bắt buộc
- Ý nghĩa và vai trò của môn học: giúp cho người học trang bị được kiến thức về lập trình trên môi trường ứng dụng web.

Mục tiêu của môn học:

 Về kiến thức:

- Trình bày được các mô hình lập trình Web
- Trình bày được các giao thức của công nghệ Web
- Trình bày được các thành phần của ngôn ngữ JavaScript
- Trình bày được các chức năng của Web Server Control
- Trình bày được các kiến trúc của công nghệ ADO.NET
- Trình bày được qui trình kết nối dữ liệu

 Về kỹ năng:

- Vận dụng được ngôn ngữ JavaScript để viết các kịch bản
- Vận dụng được ngôn ngữ lập trình trong xử lý dữ liệu.
- Vận dụng được Hệ quản trị CSDL SQL Server tương tác với Web.
- Sử dụng và đánh giá được các phương pháp truy vấn dữ liệu.
- Sử dụng được các đối tượng của ADO.NET
- Xây dựng được ứng dụng Web có sử dụng cơ sở dữ liệu

 Về năng lực tự chủ và trách nhiệm:

- Cẩn thận, nghiêm túc trong nghiên cứu
- Có khả năng xem các tài liệu hướng dẫn, đọc sách
- Cẩn thận và chính xác khi làm việc với cơ sở dữ liệu
- Có khả năng vận dụng môn học lập trình ứng dụng trên Web để giải quyết vấn đề trong thực tế.

Nội dung của môn học:

CHƯƠNG 1: GIỚI THIỆU CHUNG

Giới thiệu: Khái niệm tổng quan về cấu trúc ngôn ngữ web HTML

Mục tiêu:

- Trình bày được các khái niệm về mô hình ứng dụng Web
- Trình bày được các giao thức sử dụng cho Web
- Sử dụng được ngôn ngữ HTML để thiết kế trang Web

Nội dung chính:

1.1. Nền tảng của Web ngôn ngữ HTML

1.1.1. Cơ bản về ngôn ngữ HTML

* Cấu trúc

HTML bao gồm 3 thẻ để xác định cấu trúc của trang web bao gồm:

`<html></html>`

`<head></head>`

`<body></body>`

* Thẻ định dạng văn bản

Mặc dù có rất nhiều thẻ để định dạng văn bản, những thẻ sau đây là những thẻ cơ bản nhất mà gần như bất cứ một trang web nào cũng phải sử dụng:

`<p>`: bắt đầu một đoạn văn bản mới

`<br>`: xuống dòng

`<h1></h1>`, `<h2></h2>`,...: đặt dòng văn bản nằm trong cặp thẻ là tiêu đề (heading).

* Thẻ ghi chú

Cũng như các ngôn ngữ lập trình, để cho phép người viết trang web đặt những ghi chú dành riêng cho mình vào trong trang web, HTML cung cấp thẻ ghi chú. Đây là thẻ đặc biệt so với những thẻ khác:

`<!-- nội dung ghi chú -->`

Ghi nhớ thẻ qua ý nghĩa HTML 4.0 có tương đối nhiều thẻ, để nhớ được nhiều, người viết thường phải hiểu được ý nghĩa tên của mỗi thẻ. Các thẻ trong HTML thường là viết tắt của những từ gọi nhớ như: **P**aragraph, **B**Reack,...

1.1.2. Định dạng Text

* Định dạng kiểu chữ

Trong các tài liệu, văn bản chúng ta thường sử dụng các kiểu chữ đậm, nghiêng, gạch dưới,... ví dụ sau minh họa các thẻ được dùng định dạng kiểu chữ:

```
<html>
<body>
<b>In đậm</b> <br>
<strong>In rất đậm </strong><br>
<font size="4">c</font><big>hữ lớn </big><br>
<em>nhấn mạnh</em><br>
<i>in nghiêng </i><br>
<small>chữ nhỏ</small><br>
Công thức hoá học của nước: H<sub>2</sub>O<br>
X bình phương: X<sup> 2 </sup>
</body>
</html>
```

Để xem code HTML của một trang web đã có từ IE, trên menu View, chọn mục Source. Bạn có thể học hỏi được nhiều điều bằng cách xem code HTML của những trang web được thiết kế chuyên nghiệp nhưng hãy nhớ rằng những trang web đẹp luôn được viết rất công phu và thường sử dụng nhiều công cụ (tool) hỗ trợ.

* Font chữ, màu sắc và canh lề

```
<font face="..." size="..." color="#HHHHHH">...</font>
<p align="left/right/center">
```

Ví dụ 1:

```
<p align="center">
<font face="Algerian" size="5">Computer Joke </font>
<p><u>Kỹ thuật viên</u>:
<font face="Arial">Máy tính của anh có ổ đĩa mềm chứ ?</font>
<p><u>Khách</u>:
<i><font face="Times New Roman">Tôi không nhìn thấy bên trong. Có chữ
&quot; <b>Intel Pentium <font color="#FF0000">Inside</font>
</b>&quot;</font></i>
</p>
```

Thuộc tính của một thẻ

Một thông tin định dạng có thể gồm nhiều chi tiết, trong ví dụ trên, font chữ sẽ hiển thị cho một chuỗi văn bản được chỉ định qua thẻ tuy nhiên, font chữ lại gồm nhiều chi tiết như: tên font, kích thước, màu sắc,... Các thông tin chi tiết được gọi là các thuộc tính của thẻ. Một thẻ có thể có nhiều thuộc tính. Bạn nên đặt giá trị của thuộc tính trong dấu ngoặc kép.

Định dạng trước nội dung văn bản Web browser sẽ không quan tâm đến cách bạn trình bày đoạn code HTML trong file .html mà chỉ dựa vào các thẻ để trình bày nội dung trang web.

Thẻ <pre> được dùng khi bạn muốn yêu cầu web browser "tôn trọng" các khoảng trắng và xuống dòng trong đoạn code HTML của mình.

Ví dụ 2:

```
<!DOCTYPE html>
<html>
<head>
<title>Vi du the pre trong HTML</title>
</head>
<body>
<pre>
Phan van ban
```

```
nay se duoc giu nguyen
dinh dang, khoang cach le va
vi tri cua cac tu.
```

```
</pre>
</body>
</html>
```

1.1.3. Liên kết các trang web (Link)

URL: (Uniform Resource Locator), là một đường dẫn được dùng trên Internet để chỉ tới một trang web cụ thể nào đó. Thuật ngữ thường dùng thay cho url là : "địa chỉ"

Domain name: Là tên dễ nhớ của một địa chỉ. Những tên này được quản lý bởi một tổ chức quốc tế, đảm bảo không có hai địa chỉ khác nhau nào có cùng tên. Nếu bạn muốn website của mình có một tên gọi nhớ để mọi người có thể truy cập, bạn sẽ phải đem tên đó đi đăng ký.

Trong domain name, phần cuối cùng dùng để phân loại các website:

- * Com : commercial – website thương mại, kinh doanh
- * Edu : education – website về giáo dục, đào tạo
- * Gov : government – website của chính phủ
- * vn, uk, au, ... : vietnam, united kingdom, austratlia – website của quốc gia nào.

* Tạo liên kết

HTML dùng thẻ <a> (anchor) để tạo liên kết tới một trang web. Thẻ <a> có ba thuộc tính chính là:

- * href : địa chỉ của trang web muốn liên kết
- * target : cửa sổ sẽ hiển thị trang web
- * name : tên của mỗi liên kết

Ví dụ 3:

```
<a href="http://www.yahoo.com">Liên kết tới Yahoo!</a>
```

Thuộc tính target chỉ ra cửa sổ sẽ dùng để mở trang web mới. Nếu không đặt giá trị cho target, trang web bạn đang xem sẽ bị thay thế bằng trang web mới. Để mở trang web trong một cửa sổ mới, đặt target="_blank"

***. Liên kết trong cùng trang web**

Nếu như cho bạn được quyền đặt tên cho các thẻ của HTML, có lẽ bạn sẽ thay <a> bằng <l>

(Link) thì đúng hơn. Tuy nhiên <a> thực sự mang ý nghĩa là một mỏ neo (anchor) khi bạn dùng để liên kết tới một đoạn văn bản nào đó trong chính bản thân trang web.

Thuộc tính name của <a> dùng để đặt tên cho đoạn văn bản sẽ liên kết tới. Chú ý, giá trị của name có dấu # đứng trước.

Ví dụ 4:

```
<a href="#EndOfPage">Đến cuối trang</a>
```

```
..... <!-- nhiều dòng -->
```

```
<a name="#EndOfPage">cuối trang</a>
```

*** Liên kết với địa chỉ email**

Để cho phép người đọc gửi mail cho bạn bằng cách click vào liên kết, gán giá trị "mailto:địa chỉ email" cho thuộc tính href.

1.1.4. Danh sách (List)

Danh sách gồm 2 loại: có thứ tự và không có thứ tự

Danh sách trong HTML tương tự như định dạng Bullets and Numbering trong Word. Thông thường, chúng ta ít phân biệt giữa danh sách có thứ tự và không có thứ tự. Với danh sách có thứ tự, mỗi mục sẽ được đánh thứ tự 1, 2, 3 hay a, b, c, ... trong khi với danh sách không có thứ tự, mỗi mục sẽ bắt đầu bằng dấu -, □, ■, à, o, ...

Trong HTML, mỗi mục trong danh sách được bắt đầu bằng thẻ . Các mục trong danh sách lại được đặt trong một thẻ danh sách. HTML có các thẻ danh sách:

 : ordered list – danh sách có thứ tự

 : unordered list – danh sách không có thứ tự

Ví dụ 5:

Nội dung môn học lập trình web cơ bản

```
<ol>
```

```
<li>HTML </li>
```

```
<li>JavaScript </li>
```

```
</ol>
```

Kết quả:

Nội dung môn học lập trình web cơ bản

1. HTML

2. JavaScript

Ví dụ:

Nội dung môn học lập trình web cơ bản

```
<ul>
```

```
<li>HTML </li>
```

```
<li>JavaScript </li>
```

```
</ul>
```

Kết quả:

Nội dung môn học lập trình web cơ bản

- HTML
- JavaScript

Thuộc tính type của các thẻ danh sách cho phép bạn định lại các số thứ tự hay bullet hiển thị đầu mỗi mục trong danh sách. Các giá trị của type:

Bảng 1-1: Kết quả bảng danh sách

- Order list	- Unorder list
"A" : A, B, C, ...	"disk" : ●
"a" : a, b, c, ...	"circle" : ○
"I" : I, II, III, ...	"square" : ■
"i" : i, ii, iii, ...	
"1" : 1, 2, 3, ... (mặc định)	

1.2. Các giao thức sử dụng cho Web

Thuộc tính HREF (tham chiếu siêu văn bản) được dùng để chỉ địa chỉ hay URL của tài liệu hoặc file được liên kết.

Cú pháp của HREF là:

Hypertext

Trong đó,

Giao thức – Đây là loại giao thức. Một số giao thức thường dùng:

http – giao thức truyền siêu văn bản

telnet – mở một phiên telnet

gopher – tìm kiếm file

ftp – giao thức truyền file

mailto – gửi thư điện tử

Host.domain – Đây là địa chỉ Internet của máy chủ

Port - Cổng phục vụ của máy chủ đích

HyperText – Đây là văn bản hay hình ảnh mà người dùng cần nhấp vào để kích hoạt liên kết

1.3. Các mô hình ứng dụng Web

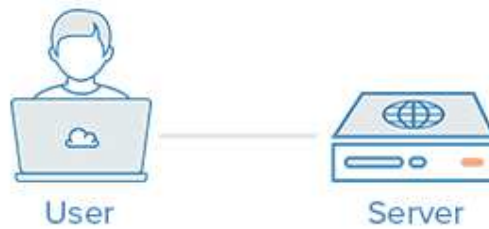
Có rất nhiều mô hình cài đặt Server dành cho ứng dụng (Application Server) và mỗi mô hình lại có sự ưu việt khác nhau. Để lựa chọn mô hình Server tối ưu cho ứng dụng của mình bạn cần dựa vào các yếu tố như hiệu suất, khả năng mở rộng, tính sẵn có, độ tin cậy, giá cả và khả năng quản lý.

Dưới đây là 5 mô hình Cloud Server dành cho Application phổ biến nhất để tham khảo. Các mô hình này có thể sử dụng kết hợp lẫn nhau và phù hợp với từng môi trường cũng như từng loại ứng dụng khác nhau.

1.3.1. Tất cả trong một Server

Đây là mô hình đơn giản nhất khi tất cả tài nguyên của ứng dụng đều được đặt chung trên một máy chủ duy nhất. Một gói tài nguyên ứng dụng bao gồm các web server, application server và database server. Chẳng hạn như gói LAMP bao gồm Linux, Apache, MySQL và PHP trên cùng một server.

Single Server



Hình 1-1 Sử dụng chung phần tài nguyên server

Ưu điểm của mô hình này là sự đơn giản, nhanh chóng khi cài đặt ứng dụng. Tuy nhiên nó chỉ là mô hình cơ bản và sẽ gây khó khăn trong việc mở rộng cũng như tách biệt các thành phần của ứng dụng với nhau. Ứng dụng và database của nó sử dụng chung phần tài nguyên của cloud server (CPU, bộ nhớ, I/O,...) khiến hiệu suất suy giảm và khó xác định lỗi.

1.3.2. Tách riêng Database Server

Mô hình này tách biệt hệ thống quản lý database với phần còn lại để tránh tranh chấp tài nguyên giữa ứng dụng và database. Tính bảo mật cũng có thể tăng cường bằng cách đặt database trong một private network.

Separate Database Server



Hình 1-1 Tách biệt hệ thống quản lý database

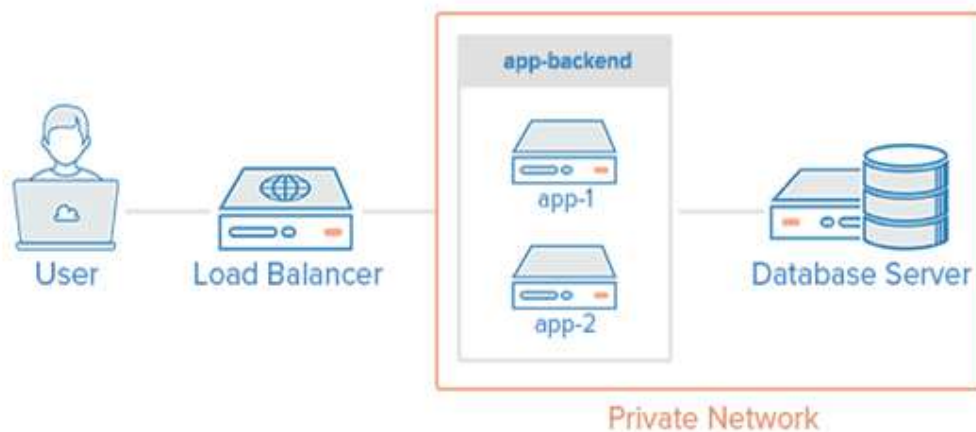
Ưu điểm của mô hình này là hiệu suất sẽ được cải thiện do ứng dụng và database sử dụng tài nguyên riêng. Ngoài ra có thể tăng thêm tài nguyên cho server khi có nhu cầu mở rộng. Tuy nhiên nó đòi hỏi cài đặt phức tạp hơn mô hình all-in-one và nếu như hai server ứng dụng và database có độ trễ lớn (do khoảng cách quá xa hoặc băng thông quá thấp so với lượng dữ liệu truyền tải) thì hiệu suất sẽ bị suy giảm.

1.3.3. Sử dụng Load Balancer (Reverse Proxy)

Để cải thiện hiệu suất và độ tin cậy trong môi trường máy chủ, chúng ta có thể phân phối việc xử lý request trên nhiều máy chủ bằng load balancer. Nếu một trong các máy chủ bị lỗi, load balancer sẽ gửi các yêu cầu xử lý lưu lượng đến các máy chủ khác cho đến khi máy chủ bị lỗi hoạt động trở lại bình thường.

Load balancer cũng có thể sử dụng để phục vụ cho nhiều ứng dụng thông qua cùng một tên miền và cổng bằng một layer application reverse proxy. Các phần mềm có khả năng load balancer reverse proxy có thể kể đến HAProxy, Nginx hay Varnish.

Load Balancer



Hình 1-2 Mô hình máy chủ load balancer

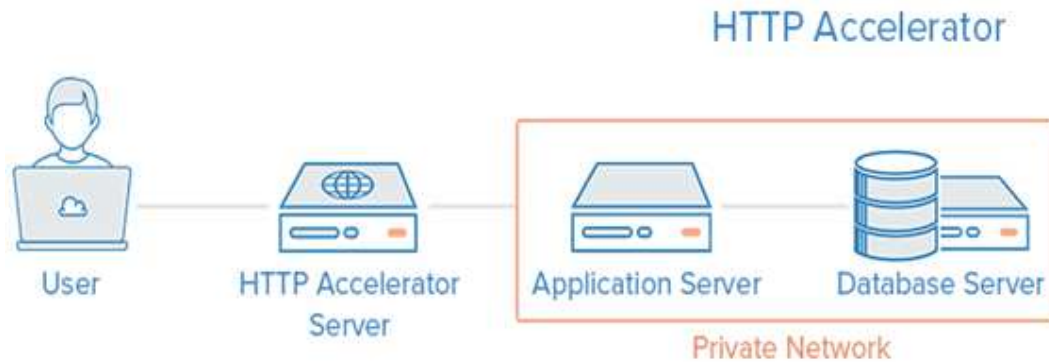
Ưu điểm của mô hình này là có thể mở rộng bằng cách thêm nhiều máy chủ, và chống lại DDOS bằng cách hạn chế kết nối của người dùng xuống mức tần suất hợp lý. Tuy nhiên nếu load balancer không đủ hiệu suất hoặc bị cấu hình kém thì có thể trở thành nút cổ chai gây tắc nghẽn hệ thống cloud.

Ngoài ra điểm yếu chí mạng của mô hình này là nếu máy chủ load balancer bị lỗi thì cả hệ thống sẽ sập theo. Các vấn đề như nơi thực hiện chấm dứt SSL và xử lý đòi hỏi session cũng cần phải xem xét.

1.3.4. Dùng HTTP Accelerator (Caching Reverse Proxy)

Các kỹ thuật như dùng HTTP accelerator hay caching HTTP reverse proxy có thể giảm thời gian tải nội dung từ server đến người dùng. HTTP accelerator sẽ lưu trữ nội dung mà ứng dụng trả về người dùng vào bộ nhớ lần đầu và nếu sau này có yêu cầu truy cập tương tự, HTTP accelerator chỉ cần lấy nội dung trong bộ nhớ ra để trả về người

dùng mà không cần tương tác với web server nữa. Các phần mềm tăng tốc HTTP trên cloud server có thể kể đến Varnish, Squid hay Nginx.



Hình 1-3 Mô hình Accelerator

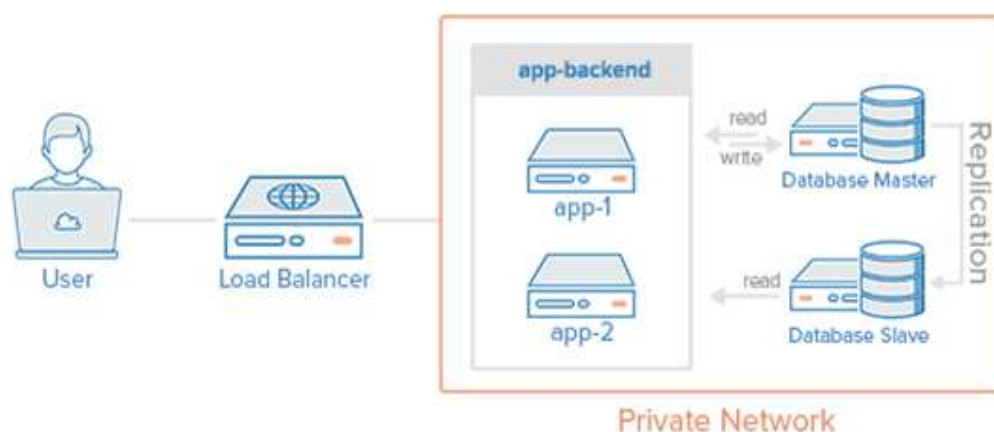
Mô hình này rất hữu ích trong việc cải thiện hiệu suất các website có nội dung nặng hay có nhiều tài nguyên được truy cập thường xuyên, do cho phép giảm tải CPU trên web server thông qua bộ nhớ cache và nén. HTTP accelerator server có thể sử dụng như một load balancer server, và các phần mềm caching cũng được dùng để bảo vệ hệ thống chống lại các cuộc tấn công DDOS.

Hai điểm đáng chú ý khi sử dụng HTTP accelerator đó là cần điều chỉnh để có được hiệu suất tốt nhất, và nếu tỷ lệ tái sử dụng bộ nhớ cache thấp thì hiệu suất cloud server có thể bị suy giảm.

1.3.5. Cấu hình Master-Slave Database Replication

Nếu như hệ thống của bạn có số yêu cầu đọc thực hiện nhiều hơn số yêu cầu ghi (ví dụ như CMS) thì có thể cấu hình master-slave database replicate để cải thiện hiệu suất. Mô hình này đòi hỏi một master database và một hay nhiều slave database. Yêu cầu cập nhật dữ liệu sẽ được gửi cho master và các yêu cầu đọc dữ liệu sẽ được phân phối trên các slave.

Master-Slave Database Replication



Hình 1-4 cấu hình master-slave database replicate

Ưu điểm của mô hình này là cải thiện hiệu suất đọc dữ liệu từ database của ứng dụng. Tuy nhiên ứng dụng cần xác định nút database nào để ghi vào nút nào để đọc. Ngoài ra việc cập nhật dữ liệu cho các slave là không đồng bộ nên có khả năng dữ liệu ứng dụng đọc được không phải là mới nhất. Điểm yếu của mô hình này là không có phương án dự phòng cho trường hợp master bị lỗi, nếu như xảy ra điều này thì quá trình cập nhật dữ liệu sẽ bị gián đoạn cho đến khi master phục hồi.