

# CÔNG NGHỆ SỬ DỤNG HỆ THỐNG QUẢN LÝ CÔNG VĂN

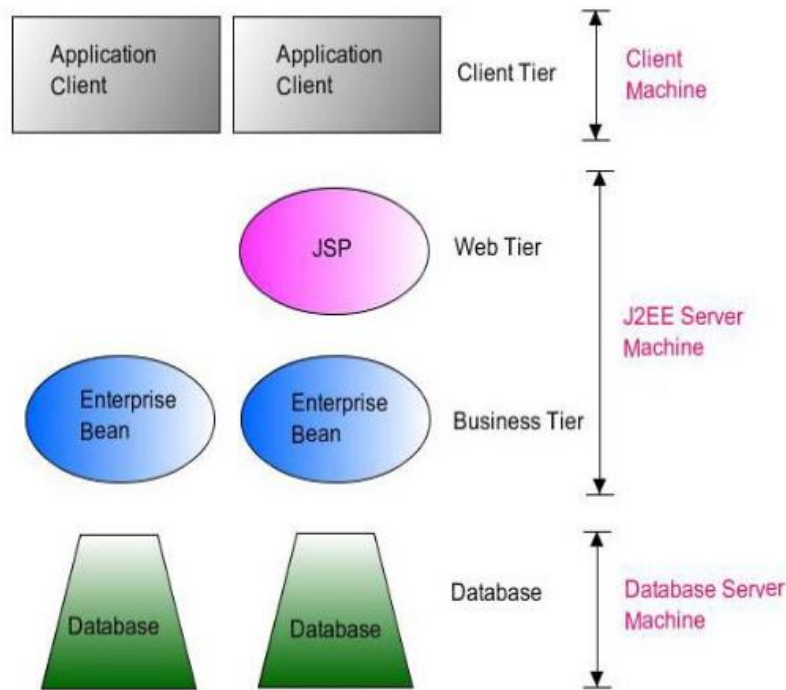
## 1. Tổng quan J2EE:

### 1.1 Khái niệm

Java 2 Platform Enterprise Edition viết tắt là J2EE, định nghĩa một chuẩn để phát triển những ứng dụng thương mại đa tầng (multitier enterprise applications). J2EE platform chứa đựng những tính năng tốt của Java 2 Platform như "viết một lần, chạy ở mọi nơi".

### 1.2 Kiến trúc của J2EE

J2EE nền tảng sử dụng một mô hình ứng dụng phân tán đa tầng.



**Hình 1- Mô hình đa tầng của J2EE**

Các công nghệ trong J2EE :

- **Java Database Connectivity (JDBC) API:** JDBC là một tập hợp các giao diện cho phép các ứng dụng Java truy cập vào cơ sở dữ liệu bất kỳ. API này có cùng một mục đích như ODBC (Open Database Connectivity) Microsoft.
- **Remote Method Invocation (RMI):** RMI là một API cho phép các đối tượng Java để giao tiếp từ xa với các đối tượng khác.

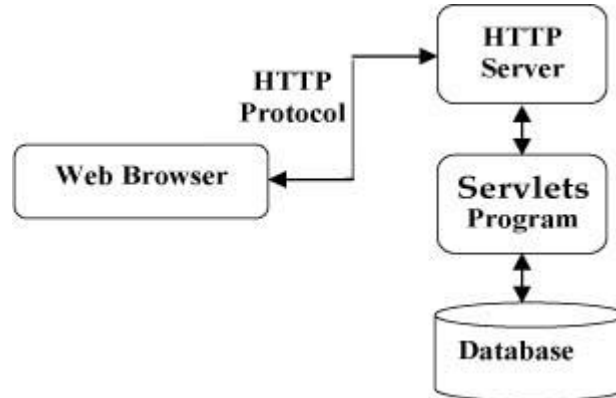
- **Java IDL:** IDL (Interface Definition Language) là một ngôn ngữ nền tảng tiêu chuẩn khai báo độc lập được sử dụng để xác định các giao diện đối tượng triển khai cung cấp và đối tượng khách hàng gọi. Java IDL cho phép bất kỳ đối tượng Java để giao tiếp với các đối tượng khác trong ngôn ngữ bất kỳ bằng phương tiện của IDL.
- **Enterprise Java Beans:** Đóng gói các logic thương mại (business logic) và hỗ trợ thực hiện giao tác và bảo mật.
- **Servlets và Java Server Pages (JSP):** Công nghệ dùng trong xây dựng những ứng dụng web.
- **Java Message Service (JMS):** Tập APIs cho phép ứng dụng tạo, gửi, nhận, và đọc thông điệp bất đồng bộ.
- **Java Transaction API (JTA):** JTA cho phép những ứng dụng J2EE thực hiện những giao tác phân tán (distributed transaction).
- **JavaMail:** Cho phép thành phần java gửi và nhận mail.
- **Java API cho xử lý XML (Java API for XML Processing -JAXP):** XML là một định dạng dữ liệu cho các tài liệu thay thế cấu trúc trên Web. JAXP cho phép các ứng dụng Java phân tích và chuyển đổi tài liệu XML. XML được sử dụng nhiều trong J2EE như là một định dạng dữ liệu.
- **Java Naming và Directory Interface (JNDI):** JNDI là một giao thức cung cấp một tập API chuẩn cho phép truy cập tới những dịch vụ naming và directory.
- **Java Persistence API (JPA) :** Cho phép ánh xạ giữa một lớp java với các cột trong một bảng.

## 2. Servlet

### 2.1 Khái niệm

Servlet là một chương trình java chạy trên máy chủ web. Trong các ứng dụng thực tế, Servlet đóng vai trò như một lớp trung gian giữa client và server. Nó sẽ tiếp nhận các yêu cầu từ client (browser) và gọi đến các ứng dụng trên server (hoặc thao tác với cơ sở dữ liệu) và trả kết quả về các file jsp. Servlet được sử dụng để xây dựng các ứng dụng web động.

## 2.2 Kiến trúc

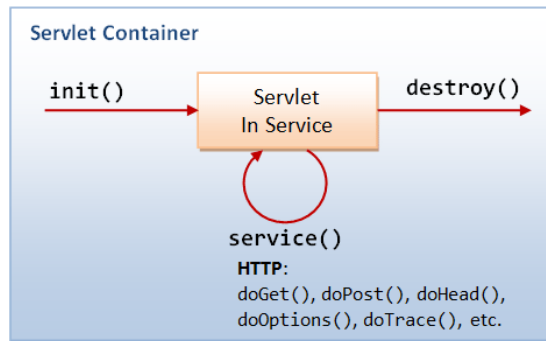


**Hình 2- Kiến trúc của servlet**

## 2.3 Các bước thực thi của servlet

- Servlet nhận các yêu cầu được gửi từ client (browser)
- Servlet đọc các dữ liệu được gửi đến từ client, như cookies, các tham số...
- Xử lý dữ liệu đó và tạo ra kết quả. quá trình xử lý có thể servlet sẽ gọi đến một ứng dụng server, hay thao tác với cơ sở dữ liệu (như lấy dữ liệu, thêm mới, cập nhật, xóa).
- Trả về kết quả cho client (browser). dữ liệu trả về có thể là 1 file html, hay các dữ liệu lưu trong bộ nhớ.

Bản chất của Servlet là một lớp java được kế thừa từ lớp `javax.servlet.http.HttpServlet`.



**Hình 3 - Vòng đời của servlet.**

Vòng đời của một servlet được hiểu là toàn bộ quá trình từ lúc khởi tạo cho đến lúc hủy servlet đó đi. Một vòng đời của servlet trải qua 4 giai đoạn:

### **Giai đoạn 1: khởi tạo bằng cách gọi phương thức init().**

Định nghĩa một method init() trong servlet:

```
public void init() throws ServletException {  
    // Initialization code...  
}
```

- Phương thức init() được gọi đến một lần duy nhất, nó được gọi đến khi ta click vào 1 link, hay có một redirect từ một servlet khác đến. Mục đích của init() là nạp servlet vào bộ nhớ của máy chủ.
- Sau khi đã gọi method init() thì mỗi lần sử dụng servlet đó, máy chủ sẽ điều hướng trực tiếp vào các hàm doGet() hoặc doPost() tùy theo yêu cầu từ client. mà không phải gọi init() lại nữa.

### **Giai đoạn 2: Gọi phương thức service() để xử lý các request từ client.**

Cú pháp:

```
public void service(ServletRequest request, ServletResponse response)  
    throws ServletException, IOException{  
}
```

Sau khi hàm init() được gọi thành công. hàm service() sẽ tự động được gọi nhằm mục đích kiểm tra các yêu cầu từ phía client thuộc loại nào của http: GET, POST, PUT, DELETE để điều hướng vào các method tương ứng lần lượt là: doGet, doPost, doPut, doDelete

Có hai method được thường xuyên sử dụng là doGet, doPost.

- **doGet() Method**

Một request từ client dưới dạng một link, hay một form có method = "get" thì servlet sẽ được xử lý trong method **doGet()**.

Cú pháp:

```
public void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    // Servlet code
}
```

- **doPost() Method**

Khi submit một form từ client đến servlet và method của form đó là POST thì servlet sẽ được xử lý ở method doPost().

Cú pháp:

```
public void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    // Servlet code
}
```

### **Giai đoạn 3: Các servlet được hủy khi gọi phương thức destroy().**

Để giải phóng tài nguyên, cụ thể như: đóng kết nối đến cơ sở dữ liệu, đóng file đang mở, xóa các dữ liệu lưu trong cookies, session.

Cú pháp:

```
public void destroy() {
    // Finalization code...
}
```

### **Giai đoạn 4: Cuối cùng, tình dọn rác garbage collected trong java sẽ thu hồi các tài nguyên liên quan đến servlet đó.**

### 3. Java Server Page

#### 3.1 Khái niệm

JSP là một công nghệ cho phép chúng ta dễ dàng viết các trang web động (chúng ta cũng có thể viết các trang web tĩnh với JSP). Cụ thể, một trang JSP là một trang HTML (hay XML) trong đó có thể trộn các mã Java, tức các thành phần JSP, cho phép thực hiện nội dung động. Trang JSP có phần mở rộng .jsp.

#### 3.2 Ưu điểm

\*Lợi thế của việc sử dụng JSP khi so sánh với các công nghệ khác:

- **So với Active Server Pages (ASP):** Lợi thế của JSP có thể coi là gấp đôi. Đầu tiên, các phần động được viết bằng Java, không phải bằng Visual Basic hoặc ngôn ngữ MS khác, vì thế nó mạnh mẽ hơn và dễ dàng để sử dụng hơn. Thứ hai, nó thích hợp cho các Hệ điều hành khác, không chỉ là Microsoft Web Server.
- **So với Pure Servlets:** Nó tiện lợi hơn khi viết (và sửa đổi) HTML, vì có nhiều lệnh `println` hơn.
- **So với Server-Side Includes (SSI):** SSI chỉ dành cho các thể đơn giản, không dành cho các chương trình “thực” mà sử dụng Form Data, tạo kết nối Database.
- **So với JavaScript:** JavaScript có thể tạo HTML động trên Client nhưng lại tương tác khó khăn với Web Server để thực hiện các tác vụ phức tạp như truy cập Database và xử lý hình ảnh, ...
- **Với Static HTML:** HTML thông thường không thể chứa thông tin động.

### 4. Spring MVC Framework

#### 4.1 Giới thiệu

- Spring Framework là một ứng dụng mã nguồn mở phổ biến cho Java EE phát triển dễ dàng hơn. Nó bao gồm một container, một framework cho các thành phần quản lý, và một bộ các snap-in dịch vụ cho các giao diện người dùng web, giao dịch,... Một phần của Spring Framework là Spring Web MVC, một mô hình MVC mở rộng để tạo các ứng dụng web.

- MVC Framework thì cài đặt đầy đủ đặc tính của MVC pattern để xây dựng các ứng dụng Web. MVC framework thì cấu hình thông qua giao diện và chứa được một số kỹ thuật view bao gồm: JSP, Velocity, Tiles và generation of PDF và Excel file. . Spring MVC framework cung cấp sự phân biệt rõ ràng giữa domain model và web form.
- Spring Framework như thùng chứa nhẹ và hỗ trợ nhiều framework và các thư viện. Nó cho phép các nhà phát triển trộn và kết hợp nhiều framework khi phát triển và triển khai các ứng dụng. Các Spring MVC có những lợi thế từ Spring framework và cung cấp một lần tốt nhất cho framework và hiệu quả cho sự phát triển của các ứng dụng web.
- Các MVC Spring là một framework phát triển web dựa trên các mẫu thiết kế MVC (Model View Controller). Các tính năng của framework Spring MVC là Pluggable công nghệ View và Injection dịch vụ vào điều khiển.

#### 4.2. Ưu điểm

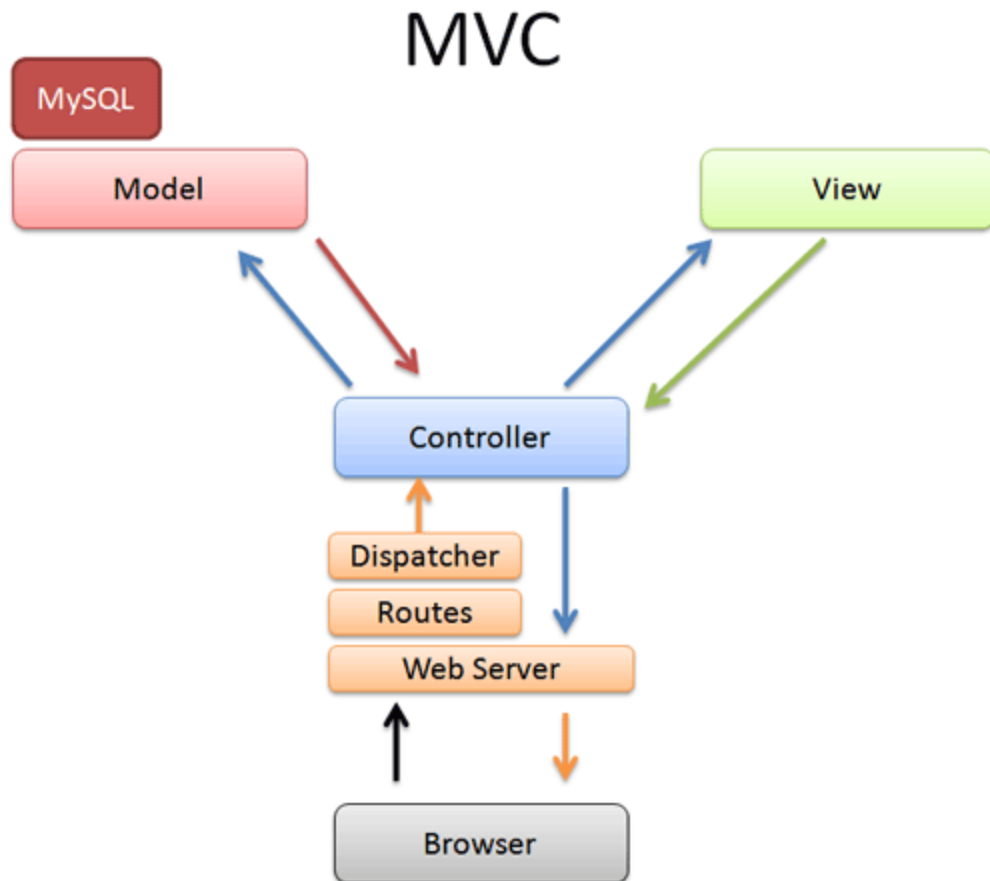
- Tất cả các framework đã được tích hợp rất tốt vào Spring.
- Hoạt động rất tốt khi áp dụng theo kiến trúc MVC.
- Sử dụng cơ chế plug-in.
- Kết hợp rất tốt với các O/R (object-relational) Mapping frameworks như là Hibernate.
- Dễ Testing ứng dụng.
- Ít phức tạp hơn so với các framework khác.
- Cộng đồng người sử dụng rất nhiều.

#### 4.3. Spring Web MVC

##### 4.3.1 Mô hình MVC

- **Khái niệm:** Mô hình MVC (Model - View - Controller) là một kiến trúc phần mềm hay mô hình thiết kế được sử dụng trong kỹ thuật phần mềm. Làm ứng dụng tách ra 3 thành phần khác nhau Model, View và Controller, giúp các chuyên gia có thể dễ dàng dựa vào mô hình để trao đổi và xử lý những nghiệp vụ một cách nhanh chóng. Mỗi thành phần có một nhiệm vụ riêng biệt và độc lập với các thành phần khác.
- **Đặc điểm:**
  - + Giúp cho ứng dụng dễ bảo trì.
  - + Module hóa các chức năng và được xây dựng nhanh chóng.
  - + Dễ dàng thêm các tính năng mới, và thay đổi các tính năng cũ.
  - + MVC cho phép thay đổi trong 1 phần của ứng dụng mà không ảnh hưởng đến các phần khác.
- **Ưu điểm:**
  - + MVC làm cho ứng dụng trở nên trong sáng, giúp lập trình viên phân tách ứng dụng thành ba lớp một cách rõ ràng. Điều này sẽ rất giúp ích cho việc phát triển những ứng dụng xét về mặt lâu dài cho việc bảo trì và nâng cấp hệ thống.

+ MVC hiện đang là mô hình lập trình tiên tiến bậc nhất hiện nay, điều mà các framework vẫn đang nỗ lực để hướng tới sự đơn giản và yếu tố lâu dài cho người sử dụng.



**Hình 4– Mô hình MVC**

#### **4.3.2 Model (Mô hình)**

Model gồm các lớp java có nhiệm vụ:

- Biểu diễn data và cho phép truy cập tới để get và set data trong (JAVABEAN), Thường thì phần layer này mô phỏng 1 cách đầy đủ đối tượng từ thế giới thực.
- Nhận các yêu cầu từ khung nhìn.
- Thi hành các yêu cầu đó (tính toán, kết nối CSDL ...).
- Trả về các giá trị tính toán theo yêu cầu của Controller.

#### **4.3.3 View (khung nhìn)**

Bao gồm các mã tương tự như JSP, HTML, CSS, XML, Javascript, JSON... để hiển thị giao diện người dùng, các dữ liệu trả về từ Model thông qua Controller...



#### **4.3.4 Controller (Bộ điều khiển)**

- Đồng bộ hoá giữa View và Model. Tức là với một trang JSP này thì sẽ tương ứng với lớp java nào để xử lý nó và ngược lại, kết quả sẽ trả về trang jsp nào. Nó đóng vai trò điều tiết giữa View và Model.
- Như vậy, chúng ta có thể tách biệt được các mã java ra khỏi mã html. Do vậy, nó đã giải quyết được các khó khăn đã nêu ra trong Mô hình 1. Người thiết kế giao diện và người lập trình java có thể mang tính chất độc lập tương đối.
- Việc debug hay bảo trì sẽ dễ dàng hơn, việc thay đổi các theme của trang web cũng dễ dàng hơn ...

### **5. Hibernate**

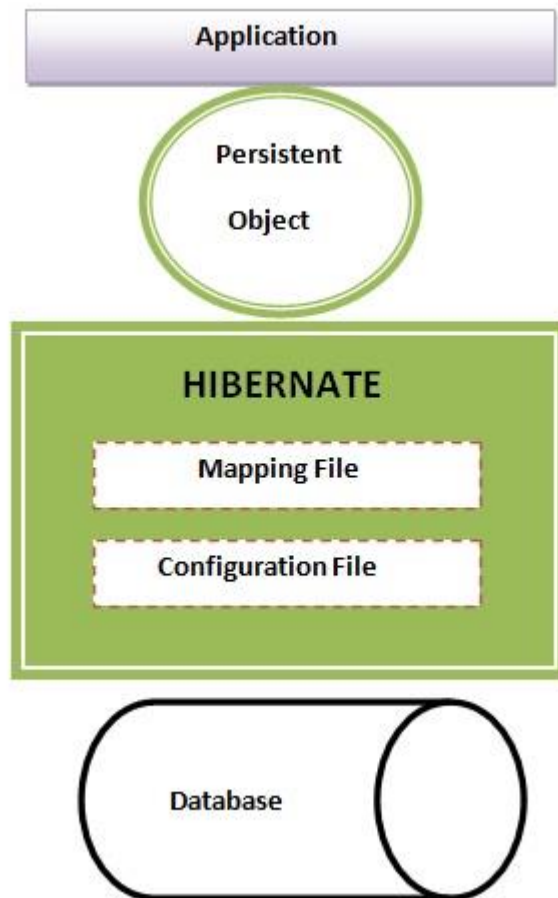
#### **5.1 Khái niệm**

Hibernate là một framework cung cấp API cho phép ứng dụng có thể thao tác với dữ liệu trong database.

#### **5.2 Cài đặt**

- Hibernate được build thành 1 tập tin JAR. Chỉ cần download tập tin này vào lưu nó vào thư mục LIB của project mà bạn đang phát triển.
- Download JDBC Driver cho database (cũng được build thành 1 tập tin JAR).
- Thêm các thành phần liên quan của Hibernate vào classpath.

### 5.3. Kiến trúc



**Hình 5 – Kiến trúc Hibernate**

### 5.4. Sử dụng

- Mỗi table trong database là một object trong Hibernate. Do đó, bạn cần có một java bean cho mỗi table trong database. Các java bean này sẽ có các getters / setters và một số ngoại lệ theo quy ước của Hibernate.
- Để Hibernate biết một object được “ánh xạ” (mapping) như thế nào với table trong database, Hibernate yêu cầu bạn cung cấp tập tin đặc tả gọi là mapping file. Theo quy ước của Hibernate, các mapping file này có đuôi là .hbm.xml và phải hợp lệ với DTD (Document Type Definition) mà Hibernate đã đưa ra. Trong các mapping file này, bạn sẽ đặc tả các mối quan hệ giữa property của object với field của table. Tức là bạn giúp cho Hibernate hiểu “mối quan hệ giữa các object” tương ứng như thế nào với “mối quan hệ giữa các field”.
- Tiếp theo nữa, bạn cần có 1 tập tin để Hibernate bắt đầu: hibernate.cfg.xml. Có thể nói tập tin này được load lên đầu tiên khi bạn “khởi động” Hibernate. Trong tập tin này, bạn đặc tả các thông tin sau:

- + SQL Dialects.
- + JDBC connection property: connection (url, driver class name, user name, password, pool size).
- + Hoặc là datasource property: datasource JNDI name, user name, password.
- + Hibernate configuration: show sql, ...
- + Hibernate cache configuration.
- + Hibernate transaction configuration.
- + Miscellaneous configuration.

### **5.5. Lợi ích khi sử dụng Hibernate (với HQL – Hibernate Query Language)**

- Tìm kiếm và sắp xếp nhanh.
- Làm việc được với dữ liệu lớn.
- Làm việc trên nhóm dữ liệu.
- Joining, aggregating.
- Chia sẻ nhiều người dùng và nhiều vùng.
- Giải quyết tương tranh (Transaction)
- Hỗ trợ cho nhiều ứng dụng.
- Bảo đảm toàn vẹn.
- Ràng buộc nhiều cấp độ.
- Tách biệt giao tác.

## **6. Javascript**

### **6.1. Khái niệm**

JavaScript là một ngôn ngữ chương trình máy tính động. Được sử dụng phổ biến nhất như là một phần của các trang web, cho phép Client-Side script tương tác với người sử dụng và tạo các trang web động. Nó là một ngôn ngữ chương trình thông dịch với các khả năng hướng đối tượng.

### **6.2. Chức năng**

- JavaScript dùng để viết các trang HTML sinh động.
- JavaScript xuất những text một cách động cho các trang HTML.
- JavaScript bắt và xử lý các sự kiện từ giao tiếp của người sử dụng Webbrowser .
- JavaScript có thể đọc và viết các phần tử cơ bản hay nội dung của trang HTML.
- JavaScript có thể được sử dụng để kiểm tra dữ liệu trước khi submit.
- JavaScript có thể cung cấp thông tin về browser.
- Tạo cookies.

### **6.3 Hạn chế**

- Client-side JavaScript không cho phép đọc và ghi các file, bởi vì lý do bảo mật.
- JavaScript không được sử dụng cho việc kết nối mạng các ứng dụng bởi vì không có những hỗ trợ có sẵn.

- JavaScript không có bất kỳ khả năng đa luồng hoặc đa xử lý.

## 7. JQuery

### 7.1 Khái niệm

JQuery là một bộ công cụ tiện ích JavaScript làm đơn giản hóa các tác vụ đa dạng với việc viết ít code hơn. JQuery làm đơn giản hóa việc truyền tải HTML, xử lý sự kiện, tạo hiệu ứng động và tương tác Ajax.

### 7.2 Tính năng

- **Thao tác DOM:** jQuery giúp dễ dàng lựa chọn các phần tử DOM để *traverse* một cách dễ dàng như sử dụng CSS, và chỉnh sửa nội dung của chúng bởi sử dụng phương tiện Selector mã nguồn mở, mà được gọi là Sizzle.
- **Xử lý sự kiện:** jQuery giúp tương tác với người dùng tốt hơn bằng việc xử lý các sự kiện đa dạng.
- **Hỗ trợ AJAX:** jQuery giúp phát triển một site giàu tính năng và phản hồi tốt bởi sử dụng công nghệ AJAX.
- **Hiệu ứng:** jQuery đi kèm với rất nhiều các hiệu ứng đa dạng và đẹp mắt có thể sử dụng trong các Website của mình.
- **Gọn nhẹ:** jQuery là thư viện gọn nhẹ, có kích cỡ khoảng 19KB (gzipped).
- **Được hỗ trợ hầu hết bởi các trình duyệt hiện đại:** jQuery được hỗ trợ hầu hết bởi các trình duyệt hiện đại, và làm việc tốt trên IE 6.0+, FF 2.0+, Safari 3.0+, Chrome và Opera 9.0+.
- **Cập nhật và hỗ trợ các công nghệ mới nhất:** jQuery hỗ trợ CSS3 Selector và cú pháp XPath cơ bản.
- **2.7.3 Sử dụng**
  - Có hai cách để sử dụng jQuery:
  - **Cài đặt nội bộ:** Có thể tải jQuery Library trên thiết bị nội bộ của bạn và include nó trong HTML code.
  - **Sử dụng từ CDN (CDN Based Version):** Có thể include thư viện jQuery vào trong HTML code một cách trực tiếp từ Content Delivery Network (CDN).

## 8. Ajax

### 8.1 Khái niệm

AJAX là thuật ngữ viết tắt của Asynchronous Javascript and XML (sJS và XML không đồng bộ). AJAX có thể đọc là “trao quyền cho javascript” và thông qua js để cung cấp một công nghệ phía client-script để gọi ngầm một lệnh background để phía server thực hiện và nhận thông tin trả về, update thông tin của trang nhanh mà không cần phải load lại cả trang, tránh được mất thời gian load lại những cái không cần update.

### 8.2 Ưu điểm

- Nó giúp việc thiết kế web đa dạng hơn và tăng tính tương tác của website với người dùng.
- Nó sử dụng các công nghệ đã có sẵn nên dễ học và sử dụng.

- Được hỗ trợ trong các trình duyệt phổ biến hiện nay

### 8.3 Hạn chế

- Không thể bookmark nó vào favourite trên trình duyệt hay gởi link đến cho bạn bè, vì tất cả quá trình nó thực hiện ngầm và không hiển thị trên address
- Không thể hiển thị nội dung trên các trang tìm kiếm vì các trang tìm kiếm hiện nay vẫn chưa hỗ trợ tìm vì rất khó tìm và gần như không thể tìm đc.
- Không thể sử dụng nút back vì back cũng là chính nó.
- Với một số trình duyệt, do nhu cầu bảo mật, sẽ tắt chức năng thực hiện javascript nên ajax không thể chạy, hay trong một vài host, không hỗ trợ vào sâu cấu hình server nên hay bị lỗi “Access denied”.

## 9. HTML/HTML5

### 9.1 Khái niệm

HTML (HyperText Markup Language) là ngôn ngữ đánh dấu siêu văn bản được sử dụng để tạo một trang web.

HTML5 là một chuẩn mới và là thế hệ tiếp theo của HTML.

### 9.2. Ưu điểm của HTML5

\* Một số điểm mới trên HTML5 :

- Giảm thiểu nhu cầu sử dụng các plugins bên ngoài, với HTML5, sẽ không cần đến các công nghệ độc quyền. Ngôn ngữ web HTML5 ra đời nhằm mục đích giảm bớt sự phụ thuộc và cần thiết của những công nghệ ứng dụng Internet độc quyền như Adobe Flash, Microsoft Silverlight và Sun JavaFX. HTML5 cho phép xem video mà không cần sử dụng một plugin như Flash hoặc Silverlight.
- Thêm các thẻ đánh dấu mới để dần thay thế cho các mã lập trình. Điểm đặc trưng đầu tiên của một ngôn ngữ đánh dấu (markup language) là các thẻ. Với HTML5 được bổ sung rất nhiều cái mới, từ các thẻ tổ chức nội dung (article, aside, title...) đến các thẻ hỗ trợ tương tác và multimedia (video, audio...). Trong HTML5 cũng xuất hiện một khái niệm gọi là semantic markup, tức là các thẻ có mang ngữ nghĩa. Các thẻ này ra đời từ việc khảo sát các trang web và nhận diện một số thói quen đặt tên phổ biến (một số phần của trang web thường luôn được đặt một cái tên như “header”, “footer”, “nav”). Ngoài sự rõ ràng, sử dụng các semantic markup còn có thể đem lại lợi thế khi các công cụ tìm kiếm trong tương lai tận dụng chúng để phân loại kết quả.
- Công cụ thiết kế mới  
HTML5 và CSS3 cũng làm cho các ứng dụng Web và các trang Web hấp dẫn hơn. HTML5 có các tính năng mới được thêm vào giúp cho việc xây dựng ứng dụng web dễ dàng hơn rất nhiều. Ví dụ như nhiều màu sắc hơn và hỗ trợ đường cong, việc làm mờ, góc tròn (thay vì ép buộc các nhà thiết kế web sử dụng các file ảnh để tạo ra các góc tròn), và dĩ nhiên là cả việc lưu trữ offline. Giúp trang Web trở nên dễ nhìn hơn, bắt mắt hơn và làm cho mọi thứ trở nên tiến gần tới những gì mà nhà thiết kế tưởng tượng.

- Khả năng hoạt động xuyên suốt giữa các trình duyệt HTML5 cung cấp các khai báo đơn giản hơn và một API mạnh mẽ, so với các phiên bản trước, đặc tả HTML5 dài hơn đáng kể nhằm chi tiết hóa mọi hành vi để đảm bảo chúng thống nhất giữa các trình duyệt khác nhau.

## 10. CSS

### 10.1 Khái niệm

CSS (Cascading Style Sheets) là một ngôn ngữ quy định cách trình bày của các thẻ html trên trang web, là file có phần mở rộng là .css và có tác dụng tách riêng phần định dạng (style) ra khỏi nội dung HTML.

Việc sử dụng kỹ thuật CSS trong thiết kế web giúp chúng ta dễ dàng quản lý được nội dung trong HTML, tiết kiệm được nhiều thời gian trong chỉnh sửa đặc biệt là các dự án lớn được kết cấu từ nhiều trang HTML.

### 10.2 Cú pháp

Để bắt đầu một tài liệu css, chúng ta sử dụng cặp thẻ **<style></style>**.

Ví dụ :

```
<style type="text/css">
```

```
selector{
```

```
    - Thuộc tính 1
```

```
    - Thuộc tính 2
```

```
    - Thuộc tính 3
```

```
    .....
```

```
    - Thuộc tính n
```

```
}
```

```
</style>
```

Trong đó :

- Selector là đối tượng lựa chọn, có thể là thẻ html hoặc là các class và id

- Thuộc tính css là các thuộc tính quy định cho thẻ html, ví dụ như **width** quy định chiều dài, **height** quy định chiều cao cho thẻ html.

### 10.3 Cách nhúng css vào html

\* Có 3 cách nhúng css vào thẻ html như sau :

**a. Chèn vào phần head của website, trong cặp thẻ <head></head>**

```
<head>
```

```
<style type="text/css">
```

```
selector{
```

```
    - Thuộc tính 1
```

```
    - Thuộc tính 2
```

```
    - Thuộc tính 3
```

```
    .....
```

```
    - Thuộc tính n
```

```
}
```

```
</style>
```

<head>

**b. Nhúng css vào một file css riêng sau đó nhúng vào website**

Cú pháp :

<link href="Đường dẫn đến file css" rel="stylesheet" type="text/css" />

Ví dụ : <link href="styles/style.css" rel="stylesheet" type="text/css" />

**c. Chèn trực tiếp vào thẻ html**

Ví dụ : <p style="color:red;text-decoration:underline">Đây là thẻ P, có màu chữ đỏ và chữ gạch chân</p>

## 11. MySQL

### 11.1 Khái niệm

Mysql là hệ quản trị dữ liệu. Chính yếu tố phát triển trong cộng đồng mã nguồn mở nên mysql đã qua rất nhiều sự hỗ trợ của những lập trình viên yêu thích mã nguồn mở. Mysql cũng có cùng một cách truy xuất và mã lệnh tương tự với ngôn ngữ SQL. Nhưng Mysql không bao quát toàn bộ những câu truy vấn cao cấp như SQL. MySQL có 2 tùy chọn:

- **MySQL Community Edition** là phiên bản download miễn phí các cơ sở dữ liệu mã nguồn mở phổ biến nhất thế giới. Nó được phát hành theo giấy phép GPL và được hỗ trợ bởi một cộng đồng rất lớn và tích cực của các nhà phát triển mã nguồn mở.
- **MySQL Editions** tính phí cung cấp hiệu suất cao, các ứng dụng cơ sở dữ liệu mở rộng hơn.

\*Những tính năng mà Mysql Community cung cấp

- Pluggable Storage Engine Architecture
- Nhiều Máy lưu trữ:
- MySQL Replication: để cải thiện hiệu suất ứng dụng và khả năng mở rộng
- MySQL Phân vùng để cải thiện hiệu suất và quản lý các ứng dụng cơ sở dữ liệu lớn
- Thủ tục lưu trữ để nâng cao hiệu quả phát triển
- Kích hoạt để thực thi các quy tắc kinh doanh phức tạp ở cấp cơ sở dữ liệu
- Lọc để đảm bảo các thông tin nhạy cảm không bị tổn hại
- Schema hiệu để theo dõi mức độ sử dụng / ứng dụng tiêu thụ tài nguyên
- Information Schema cung cấp dễ dàng truy cập vào các siêu dữ liệu
- MySQL Connectors (ODBC, JDBC, .NET, vv) cho các ứng dụng xây dựng trong nhiều ngôn ngữ
- MySQL Workbench cho các mô hình trực quan, phát triển và quản trị SQL

### 11.2 Ưu điểm

- **Tốc độ:** MySQL rất nhanh. Những nhà phát triển cho rằng MySQL là cơ sở dữ liệu nhanh nhất mà bạn có thể có.
- **Dễ sử dụng:** MySQL tuy có tính năng cao nhưng thực sự là một hệ thống cơ sở dữ liệu rất đơn giản và ít phức tạp khi cài đặt và quản trị hơn các hệ thống lớn.
- **Hỗ trợ ngôn ngữ truy vấn:** MySQL hiểu SQL, là ngôn ngữ của sự chọn lựa cho tất cả các hệ thống cơ sở dữ liệu hiện đại. Có thể truy cập MySQL bằng cách sử dụng

dụng các ứng dụng mà hỗ trợ ODBC (Open Database Connectivity - một giao thức giao tiếp cơ sở dữ liệu được phát triển bởi Microsoft).

- **Năng lực:** Nhiều client có thể truy cập đến server trong cùng một thời gian. Các client có thể sử dụng nhiều cơ sở dữ liệu một cách đồng thời. Có thể truy cập MySQL tương tác với sử dụng một vài giao diện để bạn có thể đưa vào các truy vấn và xem các kết quả: các dòng yêu cầu của khách hàng, các trình duyệt Web...
- **Kết nối và bảo mật:** MySQL được nối mạng một cách đầy đủ, các cơ sở dữ liệu có thể được truy cập từ bất kỳ nơi nào trên Internet do đó bạn có thể chia sẻ dữ liệu của bạn với bất kỳ ai, bất kỳ nơi nào. Nhưng MySQL kiểm soát quyền truy cập cho nên người mà không nên nhìn thấy dữ liệu của bạn thì không thể nhìn được.
- **Tính linh động:** MySQL chạy trên nhiều hệ thống UNIX cũng như không phải UNIX chẳng hạn như Windows hay OS/2. MySQL chạy được các với mọi phần cứng từ các máy PC ở nhà cho đến các máy server.
- **Sự phân phối rộng:** MySQL rất dễ dàng đạt được, chỉ cần sử dụng trình duyệt web của bạn. Nếu bạn không hiểu làm thế nào mà nó làm việc hay tò mò về thuật toán, bạn có thể lấy mã nguồn và tìm tòi nó. Nếu bạn không thích một vài cái, bạn có thể thay đổi nó.
- **Sự hỗ trợ:** Có thể tìm thấy các tài nguyên có sẵn mà MySQL hỗ trợ. Cộng đồng MySQL rất có trách nhiệm. Họ trả lời các câu hỏi trên mailing list thường chỉ trong vài phút. Khi lỗi được phát hiện, các nhà phát triển sẽ đưa ra cách khắc phục trong vài ngày, thậm chí có khi trong vài giờ và cách khắc phục đó sẽ ngay lập tức có sẵn trên Internet.



# Mục lục

|   |    |
|---|----|
| 1. Tổng quan J2EE: .....                | 1  |
| 1.1 Khái niệm .....                     | 1  |
| 1.2 Kiến trúc của J2EE .....            | 1  |
| 2. Servlet .....                        | 2  |
| 2.1 Khái niệm .....                     | 2  |
| 2.2 Kiến trúc .....                     | 3  |
| 2.3 Các bước thực thi của servlet ..... | 3  |
| 3. Java Server Page .....               | 6  |
| 3.1 Khái niệm .....                     | 6  |
| 3.2 Ưu điểm .....                       | 6  |
| 4. Spring MVC Framework .....           | 6  |
| 4.1 Giới thiệu .....                    | 6  |
| 4.2. Ưu điểm .....                      | 7  |
| 4.3. Spring Web MVC .....               | 7  |
| 5. Hibernate .....                      | 9  |
| 5.1 Khái niệm .....                     | 9  |
| 5.2 Cài đặt .....                       | 9  |
| 5.3. Kiến trúc .....                    | 10 |
| 5.4. Sử dụng .....                      | 10 |
| 6. Javascript .....                     | 11 |
| 6.1. Khái niệm .....                    | 11 |
| 6.2. Chức năng .....                    | 11 |
| 6.3 Hạn chế .....                       | 11 |
| 7. JQuery .....                         | 12 |
| 7.1 Khái niệm .....                     | 12 |
| 7.2 Tính năng .....                     | 12 |
| 8. Ajax .....                           | 12 |
| 8.1 Khái niệm .....                     | 12 |
| 8.2 Ưu điểm .....                       | 12 |
| 8.3 Hạn chế .....                       | 13 |
| 9. HTML/HTML5 .....                     | 13 |

|                                    |    |
|------------------------------------|----|
| 9.1 Khái niệm .....                | 13 |
| 9.2. Ưu điểm của HTML5 .....       | 13 |
| 10. CSS .....                      | 14 |
| 10.1 Khái niệm .....               | 14 |
| 10.2 Cú pháp.....                  | 14 |
| 10.3 Cách nhúng css vào html ..... | 14 |
| 11. MySQL .....                    | 15 |
| 11.1 Khái niệm .....               | 15 |
| 11.2 Ưu điểm .....                 | 15 |