Description: Description: Description: khung hoa van - lam bia**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

------------o0o------------



**THỰC TẬP TỐT NGHIỆP**

**Đề tài: Thiết kế website bán mỹ phẩm cho cửa hàng Lan Hương sử dụng Servlet, JSP và MySQL**

**Giảng viên hướng dẫn : Ths. Phạm Thế Anh**

**Lớp : KTPM 3 – K11**

**Sinh viên thực hiện : Phạm Quốc Cường**

**Hà Nội - 2020**

Description: Description: Description: khung hoa van - lam bia**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

------------o0o------------

****

**THỰC TẬP TỐT NGHIỆP**

**Đề tài: Thiết kế website bán mỹ phẩm cho cửa hàng Lan Hương sử dụng Servlet, JSP và MySQL**

GV hướng dẫn : Ths. Phạm Thế Anh

Lớp : KTPM 3 - K11

Thành viên thực hiện : Phạm Quốc Cường

**Hà Nội - 2020**

MỤC LỤC

[LỜI CẢM ƠN 1](#_Toc28726645)

[CHƯƠNG 1. CƠ SỞ LÝ THUYẾT 2](#_Toc28726646)

[1.1 TỔNG QUAN VỀ INTERNET, WORLD WIDE WEB 2](#_Toc28726647)

[1.1.1 Internet là gì? 2](#_Toc28726648)

[1.1.2 World Wide Web 3](#_Toc28726649)

[1.2 NGÔN NGỮ JAVA 5](#_Toc28726650)

[1.2.1 Lịch sử ra đời của ngôn ngữ JAVA 5](#_Toc28726651)

[1.2.2 Java là gì? 6](#_Toc28726652)

[1.2.3 Java được đùng để làm gì? 7](#_Toc28726653)

[1.3 GIỚI THIỆU VỀ JAVA SERVLET 8](#_Toc28726654)

[1.3.1 Java Servlet là gì? 8](#_Toc28726655)

[1.3.2 Ưu điểm của Servlet 11](#_Toc28726656)

[1.3.3 Môi trường thực hiện Servlet 12](#_Toc28726657)

[1.4 GIỚI THIỆU VỀ JSP 13](#_Toc28726658)

[1.4.1 JSP là gì? 13](#_Toc28726659)

[1.4.2 Cơ chế hoạt động của trang JSP 15](#_Toc28726660)

[1.4.3 Các cú pháp cơ bản của JSP 15](#_Toc28726661)

[1.4.4 Các thẻ xử lý nhúng và chuyển hướng giữa các trang 17](#_Toc28726662)

[1.4.5 Sử dụng các đối tượng trong trang JSP 20](#_Toc28726663)

[1.5 GIỚI THIỆU HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MYSQL 21](#_Toc28726664)

[1.5.1 Giới thiệu MySQL 21](#_Toc28726665)

[1.5.2 Mục đích sứ dụng cơ sở dữ liệu: 22](#_Toc28726666)

[CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG 24](#_Toc28726667)

[CHƯƠNG 3. CÀI ĐẶT 25](#_Toc28726668)

[KẾT LUẬN 26](#_Toc28726669)

# LỜI CẢM ƠN

# CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

1. **TỔNG QUAN VỀ INTERNET, WORLD WIDE WEB**
   1. Internet là gì?

Internet là một mạng máy tính toàn cầu mà bất cứ ai cũng có thể kết nối bằng máy tính các nhân của họ. Mỗi máy tính trên Internet được gọi là một Host. Mỗi Host có khả năng như nhau về truyền thông tới máy khác trên Internet. Một số Host được nối tới mạng bằng đường nối Dial - up tạm thời, số khác được nối bằng đường nối mạng thật sự.

Vào mạng Internet, bạn sẽ có thể học tập, tìm thấy nhiều tài liệu để nghiên cứu trao đổi thư từ, nói chuyện với bạn bè, xem phim, nghe nhạc và đặc biệt là muốn đặt hàng và mua bán, ....

Internet server là những server cung cấp các dịch vụ Internet như Web server, Mail server, FPT server, ....

Chương trình máy khách (client): Là chương trình có người dùng thân thiện, chạy trên máy khách, truy cập thông tin hay dịch vụ từ những chương trình khác trên Internet.

Chương trình máy phục vụ (server): Là chương trình trên máy chủ, cung cấp thông tin hay dịch vụ được yêu cầu cho chương trình Client trên Internet. Chương trình server tập trung hóa quá trình và thông tin để chương trình Client xử lý.

Internet Service Provider (ISP): Là nơi cung cấp các dịch vụ trên Internet cho khách hàng. Mỗi ISP có nhiều khách hàng và có thể sở hữu nhiều dịch vụ Internet khác nhau.

* ***Các dịch vụ thường dùng trên Internet***

Dịch vụ World Wide Web (WWW, W3): Gọi tắt là Web. Đây là dịch vụ phổ biến hiện nay trên Internet. Để sử dụng dịch vụ này, các máy Client cần có chương trình duyệt Web gọi là Web browser. Thông qua Internet, các browser (trình duyệt) truy cập được thông tin của Web server. Các trình duyệt đồ họa thông dụng hiện nay là Chrome, Mozilla FireFox, Microsoft Edge, Internet Explorer, Opera, ...

Dịch vụ Electronic Mail: Viết tắt là Email. Đây là dịch vụ cho phép các cá nhân trao đổi thư từ với nhau qua Internet.

Dịch vụ FTP (File Transfer Protocol): Đây là dịch vụ truyền hay nhận tập tin trên Internet. Tên các FTP server thường có dạng: ftp.domainname. Giao thức truyền tập tin trên Internet là: ftp://ftp.domainname.

Để truyền thông với các máy tính khác, mọi máy tính trên Internet phải hỗ trợ giao thức chung TCP/IP. TCP/IP (Transmission Control Protocol/ Intemet Protocol).

Để thực hiện trao đổi thông tin trên mạng, mỗi máy tính trên Internet đều được định danh để phân biệt với các máy khác, thông qua địa chỉ IP (IP address). Mỗi địa chỉ IP gồm bốn số hệ thập phân có giá trị từ 0 đến 255, phân cách bằng dấu chấm.

URL (Uniform Resource Locator): Để truy xuất một dịch vụ thông tin của server, người ta cần phải xác định loại. địa chỉ và cổng của dịch vụ đó.

* 1. World Wide Web

World Wide Web là một dịch vụ phố biến hiện nay trên Internet, cho phép bạn truy xuất thông tin văn bản, âm thanh, video trên toàn thế giới. Thông tin trên Internet hiện nay được thể hiện chủ yếu qua hàng triệu trang Web.

* ***Lợi ích của Web***

Ngày nay một số Website trên tòan thế giới đã đạt tới con số khổng lổ. Thông qua Website các công ty có thể giảm tối đa chỉ phí in ấn và phân phát tài liệu cho khách hàng, đại lí ở nhiều nơi.

Người dùng không cần biết nhiều về máy tính vẫn có thể xem thông tin và mang lại hiệu quả cao nhờ công nghệ Web, giúp thiết lập nhanh các trang Web với giá thành rẻ. Tận dụng được tài nguyên sẵn có trên Internet. Không cần đầu tư lớn cho huấn luyện và đào tạo. Thông tin cập nhật mới nhất, giảm thời gian và giá thành đưa sản phẩm tới thị trường. Thông tin liên lạc nhanh chóng, hiệu quả vượt mọi khoáng cách địa lí.

* ***Trình duyệt Web***

Để sử dụng dịch vụ Web, Client cần có chương trình duyệt Web, kết nối vào Internet thông qua ISP. Nội dung tài liệu được trình bày trong trình duyệt đưới đạng ván bản hoặc đồ họa, phụ thuộc vào loại trình duyệt được sử dụng. Trong ứng dụng Web, trình duyệt Web được dùng để hiển thị nội dung các trang. Ngoài ra nó còn hiển thị kết quả lấy từ cơ sở đữ liệu.

Internet Explorer 5.0 là một trình duyệt chuẩn cho phép trình bày nội dung do Web server cung cấp, cho phép đãng kí tới bất kì Website nào trên Internet, hỗ trợ trình bày trang Web trong chế độ toàn màn hình, cung cấp trang định hướng di chuyển giúp duyệt trang Web thuận tiện hơn, cho biết Web site nào bị thay đối kể từ lần viếng thăm cuối cùng, bảo vệ theo vùng, lưu lại các Website ưa thích.

* ***Web server***

Là ứng dụng cài đặt trên máy hay Internet, cho phép nhiều người dùng có thể truy xuất cùng lúc tới một cơ sở dữ liệu gọi là ứng dụng dạng Client/Server. Khi đó hệ thống ứng dụng phải có chương trình Client xử lí một số nhiệm vụ và chương trình server xử lí nhiệm vụ khác.

Web là một ứng dụng Client/Server. Trong đó, Web server đóng vai trò một chương trình server xử lý các nhiệm vụ xác định, như tìm trang thích hợp, xử lý tổ chức dữ liệu, kiểm tra dữ liệu hợp lệ, sắp xếp dữ liệu và trá kết quả về Client, đảm bảo an toàn đữ liệu. Web cũng là nơi lưu trữ cơ sở dữ liệu. Việc xây dựng một ứng dụng trên Web có thể rất khác so với các loại lập trình khác không giống những ứng dụng Client/Server thông thường.

Trong thực tế, mỗi ứng dụng Web đều tồn tại hai loại: Trang Web tĩnh và trang Web động. Trang Web tĩnh là trang Web thông thường không kết nối cơ sở dữ liệu. Ngược lại, trang Web động là trang Web thường là trang kết nối và truy cập cơ sỡ dữ liệu. Điều này có nghĩa là khi làm mới trang Web động, dữ liệu trình bày trên trang Web được đọc từ cơ sở dữ liệu.

Để xây đựng mô hình Client/Server thông thường, ứng dụng Client duy trì nối kết với thành phần server của nó. Bên cạnh việc yêu cầu gửi và lưu dữ liệu, Client truy vấn server theo định ký hầu đám bảo kết nối vẫn tồn tại. Nếu server bị đóng, Client sẽ nhận biết điều này và thực hiện các thao tác riêng như gửi thông báo đến người dùng.

Khi xây dựng ứng dụng Web, không chỉ quan tâm đến thời gian tồn tại của server trong bộ nhớ mà còn quan tâm đến kết nối mạng Internet. Khi trình duyệt yêu cầu một trang Web, nó được server đang mở ghi nhận và gửi đi. Server sẽ kiểm tra và xác định xem tài liệu trang Web và tất cả các thành phần liên quan đến trang như: hình ảnh, âm thanh.... được chuyển tới trình duyệt hay chưa.

Để server và trình duyệt có thể liên lạc và trao đổi thông tin với nhau, ta phải xây đựng các ứng dụng Web bằng những ngôn ngữ lập trình như JAVA, PHP, ASP.NET, …

1. **NGÔN NGỮ JAVA**
2. Lịch sử ra đời của ngôn ngữ JAVA

Java ban đầu được phát triển bởi James Gosling tại Sun Microsystems (đã được Oracle mua lại) và phát hành vào năm 1995 như là một thành phần cốt lõi của nên tảng Java của Sun Microsystems . Các trình biên dịch Java, máy ảo và thư viện lớp ban đầu và tham chiếu ban đầu được Sun phát hành theo giấy phép độc quyền . Kế từ tháng 5 năm 2007, tuân thủ các thông số kỹ thuật của Quy trình cộng đồng Java. Sun đã cấp phép lại hầu hết các công nghệ Java của mình theo Giấy phép công cộng GNU. Trong khi đó, những người khác đã phát triển các triển khai thay thế cho các công nghệ này, chẳng hạn như Trình biên dịch GNU cho Java (trình biên dịch mã byte), GNU Classpath (thư viện chuẩn) và IcedTea -Web (plugin trình duyệt cho các applet).

Kể từ ngày 20 tháng 3 năm 2018, cả Java 8 và 11 đều được hỗ trợ chính thức. Các phiên bản phát hành chính của Java, cùng với ngày phát hành của chúng:

* JDK 1.0 (ngày 23 tháng 1 năm 1996)
* JDK 1.1 (ngày 19 tháng 2 năm 1997)
* J2SE 1.2 (ngày 8 tháng 12 năm 1998)
* J12E 1.3 (ngày 8 tháng 5 năm 2000)
* J2SE 1.4 (ngày 6 tháng 2 năm 2002)
* J2SE 5.0 (ngày 30 tháng 9 năm 2004)
* Java SE 6 (ngày 11 tháng 12 năm 2006)
* Java SE 7 (ngày 28 tháng 7 năm 2011)
* Java SE 8 (ngảy 18 tháng 3 năm 2014)
* Java SE 9 (ngày 21 tháng 9 năm 2017)
* Java SE 10 (ngày 20 tháng 3 năm 2018)
* Java SE 11 (ngày 25 tháng 9 năm 2018)
* Java SE 12 (ngày 19 tháng 3 năm 2019)

Phiên bản mới nhất là Java SE 12 phát hành tháng 3 năm 2019. Kể từ Java 9 không còn được hỗ trợ, Oracle khuyên người sử dụng sử dụng Java 12. Oracle phát hành bản cập nhật nào cuối cùng cho Java 8 LTS, đó là miễn phí cho mục đích thương mại, vào tháng giêng năm 2019. Java 8 sẽ được hỗ trợ với bản cập nhật nào cho mục đích cá nhân lên đến ít nhất là tháng 12 năm 2020. Oracle khuyên bạn nên gở bỏ phiên bản cũ của Java” vì rủi ro nghiêm trọng do chưa được giải quyết về vấn đề an ninh. Oracle hỗ trợ mở rộng cho Java 6 đã kết thúc vào tháng 12 năm 2018.

1. Java là gì?

Java là một ngôn ngữ lập trình hướng đối tượng (OOP) và dựa trên các lớp. Khác với ngôn ngữ lập trình thông thường, thay vì biên dịch mã nguồn thành mã máy hoặc thông dịch mã nguồn khi chạy, Java được thiết kế để biên địch mã nguồn thành bytecode, bytecode sau đó sẽ được môi trường thực thi chạy.

Điều quan trọng về Java giúp phân biệt nó với nhiều công nghệ khác là nó được thiết kế sao cho mã được viết bằng Java có thể chạy trên bất kỳ hệ thống nào mà máy áo Java (JVM) có thể chạy. Khái niệm “Viết một lần, chạy mọi nơi” được sử dụng như một khẩu hiệu để thúc đẩy các khả năng đa nên tảng của Java.

Bản thân ngôn ngữ lập trình Java là ngôn ngữ hướng đối tượng, về mặt cú pháp tương tự như C++. Không giống như một số ngôn ngữ khác đi trước nó, đã triển khai các lớp nhưng không yêu cầu sử dụng chúng, các chương trình Java luôn được thiết kế với thiết kế hướng đối tượng.

Trong khi ngôn ngữ Java và máy ảo Java chạy mã Java được ghép nối chặt chẽ, hai ngôn ngữ này tách biệt nhau. Mã từ các ngôn ngữ khác được thiết kế riêng cho JVM, như Groovy và Scala, cũng có thể chạy trên máy ảo Java.

Hãy cẩn thận đừng nhầm lẫn Java với JavaScrip. Mặc dù cả hai ngôn ngữ hiện được tìm thấy trong nhiều môi trường. JavaScript được sử dụng phố biến nhất để tăng sức mạnh tương tác bên trong trình duyệt web, là một công cụ hoàn toàn khác. Khác với một phần của cái tên, cả hai không có nhiều điểm chung.

Có 5 mục tiêu chính trong việc xây dựng ngôn ngữ Java:

* + - 1. Đơn giản, hướng đối tượng và quen thuộc.
      2. Mạnh mẽ và an toàn.
      3. Kiến trúc trung lập và di động,
      4. Thực thi với hiệu suất cao.
      5. Dịch ra bytecode, phân luồng và năng động.

1. Java được đùng để làm gì?

Tất cả các loại ứng dụng, từ các trò chơi cực kỳ phổ biến như Minecraft đến môi trường phát triển tích hợp Eclipse được các nhà phát triển sử dụng cho nhiều ngôn ngữ và nền tảng khác nhau, chạy trên Java. Java cũng cấp một số ứng dụng được xây dựng riêng cho web. Mặc dù với những cải tiến về JavaScript và HTML, các Applet Java không còn là tiêu chuẩn thực tế cho các ứng dụng web tương tác, nhiều người vẫn dựa vào Java để cung cấp trải nghiệm tương tác trên trình duyệt.

Ứng dụng JavaFX: JavaFX là một nền tảng phần mềm để tạo và phân phối các ứng dụng máy tính để bàn, cũng như các ứng dụng Internet phong phú (RIA) có thể chạy trên nhiều loại thiết bị. JavaFX dự định thay thế Swing làm thư viện GUI tiêu chuẩn cho JavaSE, nhưng cả hai sẽ được đưa vào trong tương lai gần. JavaFX có hỗ trợ cho máy tính để bàn và trình duyệt web trên Microsoft Windows, Linux và MacOS. JavaFX không hỗ trợ giao diện hệ điều hành gốc.

Một số nên tảng cung cấp hỗ trợ phần cứng trực tiếp cho Java: có các bộ điều khiển vi mô có thể chạy mã byte Java trong phần cứng thay vì máy ảo Java và một số bộ xử lý dựa trên ARM có thể hỗ trợ phần cứng để thực thi mã byte Java thông qua tùy chọn Jazelle của chúng , mặc dù hỗ trợ hầu hết đã bị loại bỏ trong các triển khai hiện tại của ARM.

1. **GIỚI THIỆU VỀ JAVA SERVLET**
2. Java Servlet là gì?

Hiện nay, trong lập trình có một xu hướng rất quan trọng đang được tập trung phát triển ứng dụng, đó là xây dựng các chương trình dịch vụ Java ở phía máy chủ (Server).

Servlet là thành phần chính được sử dụng để phát triển các chương trình dịch vụ Java ở phía máy chủ. Các Servlet là các chương trình Java thực hiện ở các ứng dụng Server (tên gọi “Servlet” cũng gần giống như “Applet” ở phía máy Client) để trả lời cho các yêu cầu của Client. Các Servlet không bị ràng buộc với một giao thức Client-Server cụ thể nào cả, nhưng giao thức thường được sử dụng là HTTP, do vậy, khi nói tới Servlet nghĩa là nói tới HTTP Servlet. Servlet là sự phát triển mở rộng của CGI để đảm bảo Server thực hiện được các chức năng của mình. Ta có thể sử dụng Servlet của Java để tuỳ chỉnh lại một dịch vụ bất kỳ, như Web Server, Mail Server, v.v.

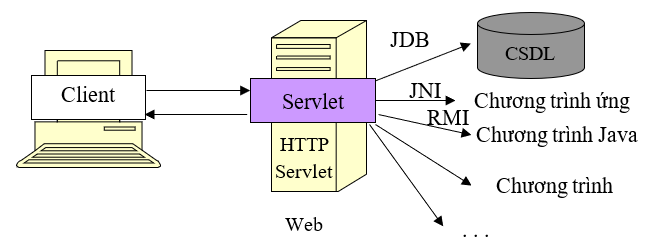
Web Server hiển thị các tư liệu được viết trong HTML và hồi đáp cho yêu cầu của người sử dụng qua HTTP. Các tư liệu HTML chứa các văn bản được đánh dấu (định dạng) để các trình duyệt như IE, Netscape đọc được.

Một trình duyệt chấp nhận đầu vào ở dạng HTML, khi người sử dụng nhấn một nút để yêu cầu một số thông tin nào đó, một Servlet đơn giản được gọi để xử lý các yêu cầu đó. Các công việc chính của Servlet được mô tả khái quát trong hình 1.1, bao gồm:

* Đọc các dữ liệu tường minh được Client gửi đến từ các yêu cầu (dữ liệu theo các khuôn dạng – form data).
* Đọc các dữ liệu không tường minh được Client gửi đến từ các yêu cầu (dữ liệu trong phần đầu của yêu cầu – request headers).
* Xử lý và lưu trữ các dữ liệu được cung cấp dưới dạng HTML.

Gửi trả lời dữ liệu tường minh cho Client (dạng HTML), cung cấp các nội dung động, ví dụ trả lời yêu cầu Client về các câu truy vấn vào các Servlet gộp là loại mô tơ được viết cho nhiều loại Server khác nhau, kể cả Apache, Fast Track Server, Enterprise Server của Netscape, Personal Web Server, v.v. Hiện nay có các loại sau:

* *Apache Tomcat:* Mô tơ này hỗ trợ thêm cho Apache. Nó được sử dụng như là một Server độc lập để kiểm tra các Servlet và các trang JSP, hoặc được tích hợp vào Apache Web Server, <http://java.sun.com.products/servlet/>.
* *Jrun của Live Software:* Jrun là mô tơ cho Servlet và JSP, hỗ trợ đầy đủ Servlet API trong các Web Server phổ biến trên mọi hệ điều hành, <http://www.allaire.com.products/jrun/>.
* *WebSphere Application Server của IBM:* còn được gọi là ServletExpress.
* *ServletExec của New Atlanta:* ServletExec là mô tơ cho Servlet và JSP, hỗ trợ đầy đủ Servlet API trong các Web Server phổ biến trên mọi hệ điều hành, <http://newatlanta.com/>.
* Quản lý các thông tin trạng thái và trả lời dữ liệu không tường minh cho Client (các mã trạng thái và các phần đầu của trả lời).



*Hình 1.1 - Vai trò của Servlet*

Viết một Servlet là tương đối dễ. Ta chỉ cần có Tomcat, nó là tổ hợp của Java Server Pages 1.1 và Servlet 2.2. Tomcat có thể tải miễn phí từ http://java.sun.com/products/jsp/tomcat/, phần cài đặt sẽ được mô tả ở phần sau.

Các Servlet cũng được sử dụng thay cho kịch bản giao diện cổng chung CGI Script. Khi tạo ra một trang Web, ta cũng sẽ tạo ra một ứng dụng Web. Trước khi sử dụng Servlet để tạo ra các ứng dụng Web, chúng ta đi tìm hiểu xem có những khả năng lựa chọn nào khác để phát triển những ứng dụng Web.

* **CGI**: Theo cách thông thường, để bổ sung các chức năng vào cho một Web Server người ta hay sử dụng Common Gateway Interface (CGI), một giao diện độc lập với ngôn ngữ cho phép một Server khởi động một tiến trình ngoại để nhận thông tin được yêu cầu thông qua các biến môi trường. Mỗi yêu cầu được trả lời bởi một tiến trình riêng thông qua một đại diện riêng của một chương trình CGI hoặc bởi một kịch bản CGI (thường được viết bằng ngôn ngữ thông dịch như Perl).
* **Fast CGI**: Open Marked đã phát triển một chuẩn khác thay cho CGI được gọi là Fast CGI. Fast CGI hành động giống như CGI. Nó khác ở chỗ, Fast CGI tạo ra một tiến trình bền vững cho từng chương trình.
* Một số chương trình ứng dụng khác như ASP.NET và JavaScript cũng hỗ trợ để tạo ra các ứng dụng Web. ASP.NET được Microsoft phát triển để tạo ra các nội dung cho các trang Web động. Trong ASP, trang HTML có thể nhúng những phần nhỏ được viết bằng VBScript hoặc JScript. Netscape đưa ra kỹ thuật được gọi là JavaScript, cho phép đưa các phần mã lệnh nhỏ nhúng vào trang HMTL, nhằm tạo ra những nội dung Web động một cách linh họat hơn. Ngoài ra, Netscape còn cung cấp NSAPI, Microsoft đưa ra ISAPI cho các Web Server của họ.
* ***Servlet có một số ưu điểm so với CGI:***
* Một Servlet không làm việc trong một tiến trình riêng. Điều này loại bỏ được việc phải tạo ra quá nhiều tiến trình mới cho mỗi yêu cầu.
* Một Servlet sẽ thường trực trong bộ nhớ giữa các yêu cầu, trong khi các chương trình CGI cần phải tải xuống và được khởi động cho từng yêu cầu CGI.
* Chỉ cần một Servlet trả lời đồng thời cho tất cả các yêu cầu. Điều này cho phép tiết kiệm được bộ nhớ và đảm bảo nó dễ dàng quản lý được dữ liệu một cách thống nhất.
* Một Servlet có thể thực hiện bởi một Servlet Engine trong phạm vi kiểm soát Sandbox để đảm bảo an toàn trong việc sử dụng các Servlet.

Các lớp Servlet của Java có thể được nạp tự động để mở rộng các chức năng của Server. Các Servlet của Java thực hiện bên trong JVM. Chúng được đảm bảo an toàn và chuyển đổi tương thích giữa các hệ điều hành và giữa các Server với nhau. Điều này khác với các Applet, Servlet chỉ thao tác được trong miền của một Server.

Servlet API được phát triển dựa trên những điểm mạnh của Java platform nhằm giải quyết vấn đề của CGI và Server API. Nó là một API đơn giản, hỗ trợ tất cả các Web server và thậm chí cho phép các ứng dụng máy chủ dùng để kiểm tra và quản lý các công việc trên Server. Nó giải quyết vấn đề thực thi bằng việc thực hiện tất cả các yêu cầu như các luồng Thread trong quá trình xử lý, hoặc việc cân bằng tải trên một Server trong các cụm máy tính Cluster. Các Servlet dễ dàng chia sẻ tài nguyên với nhau.

Trong định nghĩa Servlet, vấn đề bảo mật được cải tiến theo nhiều cách. Trước hết, bạn hiếm khi thực thi được các câu lệnh trên Shell với dữ liệu cung cấp bởi người dùng mà Java API đã cung cấp với những khả năng truy cập đến tất cả các hàm thông dụng. Bạn có thể sử dụng Java Mail để đọc và gửi mail, kết nối vào các CSDL (thông qua JDBC), tệp lớp (.class) và những lớp liên quan để truy cập hệ thống tệp, CSDL, RMI, CORBA, Enterprise Java Beans (EJB), …

1. Ưu điểm của Servlet

Servlet được sử dụng để thay thế cho những công nghệ Web động. Việc sử dụng Servlet mang lại những lợi thế:

* *Dễ di chuyển:* Servlet được viết bằng Java nên nó có tính di động cao, thực hiện được trên nhiều hệ điều hành, trên các Web Server khác nhau. Khái niệm “Viết một lần, chạy ở mọi nơi” cũng rất đúng với Servlet.
* *Mạnh mẽ:* Servlet hỗ trợ rất hiệu quả cho việc sử dụng các giao diện lõi API như lập trình mạng, xử lý đa luồng, xử lý ảnh, nén dữ liệu, kết nối các CSDL, bảo mật, xử lý phân tán và triệu gọi từ xa RMI, CORBA, v.v. Nó cũng thích hợp để trao đổi tin, truyền thông giữa Client và Server một cách bình thường.
* *Hiệu quả:* Servlet có tính hiệu quả cao. Một khi được tải về, nó sẽ được lưu lại trong bộ nhớ của máy chủ. Servlet duy trì các trạng thái của nó, do vậy những tài nguyên ngoại như việc kết nối với CSDL cũng sẽ được lưu giữ lại.
* *An toàn:* Bởi vì Servlet được viết bằng Java nên nó kế thừa được tính an toàn của Java. Cơ chế tự động dọn rác và việc không sử dụng con trỏ của Java giúp cho Servlet thoát khỏi nhiều công việc quản lý bộ nhớ. Đồng thời nó xử lý các lỗi rất an toàn theo cơ chế xử lý ngoại lệ của Java.
* *Tính tích hợp:* Các Servlet được tích hợp với các Server. Chúng cộng tác với các Server tốt hơn các chương trình CGI.
* *Tính linh hoạt:* Các Servlet hoàn toàn mềm dẻo. Một HTTP Servlet được sử dụng để tạo ra một trang Web, sau đó ta có thể sử dụng thẻ <Servlet> để đưa nó vào trang Web tĩnh, hoặc sử dụng với các Servlet khác để lọc ra các nội dung cần thiết.

1. Môi trường thực hiện Servlet

Các Servlet thường là sự mở rộng (kế thừa) các lớp chuẩn Java trong gói javax.servlet (chứa các khuôn mẫu cơ bản của Servlet) và javax.servlet.http (mở rộng các khuôn mẫu cơ bản của Servlet và các yêu cầu theo HTTP).

Servlet là một lớp Java và vì thế cần được thực thi trên một máy ảo Java (JVM) và bằng một dịch vụ được gọi là mô tơ Servlet (Servlet Engine). Servlet Engine tải lớp Servlet lần đầu tiên nó được yêu cầu, hoặc ngay khi Servlet Engine được bắt đầu. Servlet ngừng tải để xử lý nhiều yêu cầu khi Servlet Engine bị tắt hoặc nó bị dừng lại.

Như vậy, để dịch và thực hiện các Servlet, việc có các Servlet là chưa đủ, mà cần phải có một mô tơ Servlet để kiểm tra và triển khai chúng. Hiện nay một số mô tơ tương thích với nhiều loại Web Server khác nhau, nhưng nguyên lý hành động tương đối giống nhau. Người ta chia chúng thành ba loại.

* Mô tơ Servlet đơn
* Mô tơ Servlet gộp
* Mô tơ Servlet nhúng

1. **GIỚI THIỆU VỀ JSP**
2. JSP là gì?

JSP là một cách đơn giản hóa hơn cho Servlet, là bước chuyển tiếp tiếp theo của Servlet trong ứng dụng lava. Nếu như đối với Servlet ta phải viết mã Java và biên dịch thủ công trước khi đưa vào sử dụng với trình chủ Web server thì JSP không cần điều này. JSP viết mã Java tương tự Servlet nhưng cho phép trộn lẫn Java với các thẻ định đạng HTML. Trình diễn dịch JSP sẽ chịu trách nhiệm kết hợp mã Java và thẻ HTML để tạo ra Servlet xử lý các yêu cầu mà trình khác gửi đến.

Đối với trang JSP ta chỉ cân biên dịch một lần duy nhất sau đó giữ nguyên mã bytecode ở các lần thực thi kế tiếp. Chính vì lí do này trang JSP được xem là có tốc độ thực thi tương đương với Servlet và ưu điểm hơn hẳn các công nghệ xử lý trang động hiện hành như CGI (Common Gateway Interface) hay ASP (Active Server Papes).

**Chu trình sống của JSP**

Trang JSP có chu trình sống xác định tính từ khi hệ thống đọc biên dịch trang JSP, gọi thực thi và loại trang ra khỏi bộ nhớ. Chu trình sống của JSP trải qua các giai đoạn sau:

* Biên dịch trang JSP
* Nạp trang
* Khởi tạo
* Thực thi
* Dọn dẹp

*Biên dịch trang JSP*

Khi trình duyệt yêu cầu trang JSP, Web server sẽ kiểm tra trang JSP đã được biên dịch hay chưa. Nếu chưa biên dịch hoặc đã biên dịch nhưng trang JSP mới vừa thay đổi trong mã nguồn thì Web server sẽ thực hiện biên dịch trang JSP. Quá trình biên dịch trang JSP thực tế là chuyển trang JSP thành Servlet. File biên dịch .class của trang sẽ được đặt trong thư mục đệm. Như vậy quá trình biên dịch chỉ diễn ra một lần. Nếu trang đã biên dịch mà sau đó không có thay đổi mã nguồn thì quá trình biên dịch lần sau sẽ không xảy ra, do đó tốc độ thực thi trang sẽ nhanh hơn. Sau khi trang đã được biên dịch, mã trang sẽ được nạp vào bộ nhớ đề thực thị.

*Nạp trang*

Từ giai đoạn này, trang JSP đã được biên dịch ra Servlet cho nên việc nạp trang JSP cũng giống như nạp Servlet. Để nạp Servlet, trình chủ Web Server cần phải biết tên của lớp servlet (file.class) cũng là tên mà trình duyệt dùng đề triệu gọi Servlet. Các servlet (file.class) mặc định của Jrun được đặt trong thư mục [JRUN\_HOME]\servlets. Khi có một yêu cầu triệu gọi JSP, trình chủ Web server sẽ xem JSP đã nạp vào bộ nhớ hay chưa, nếu chưa có nó sẽ nạp vào bộ nhớ.

*Khởi tạo*

Khi đã nạp mã thành công, Web server sẽ gọi đến phương thức khởi tạo trang. Mặc dù JSP được dịch ra Servlet nhưng phương thức khởi tạo của JSP lại mang tên jspInit() chứ không phải là init().

*Thực thi*

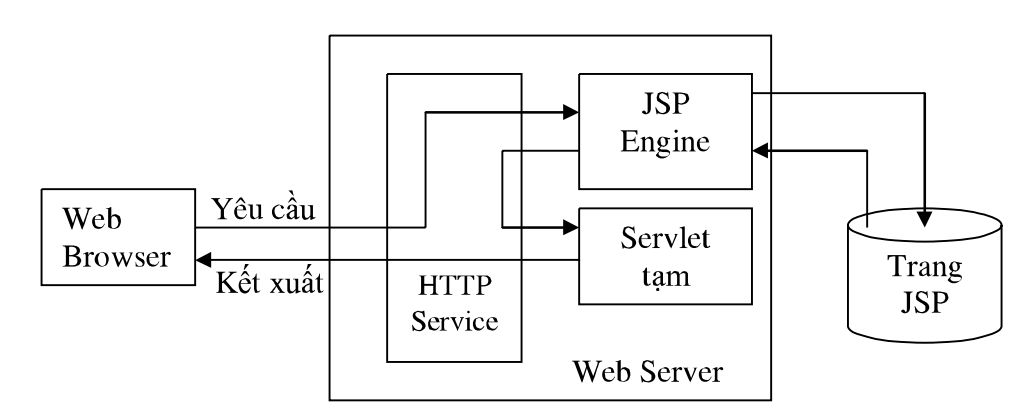
Sau quá trình khởi tạo JSP sẽ gọi đến phương thức jspService(). Phương thức này sẽ cho ta hai lớp đối tượng HttpServletRequest và HttpServletResponse để đọc và ghi kết xuất trả về trình khách.

*Dọn dẹp*

Khi trang JSP đã thực thi xong, trình chủ Web server sẽ gọi phương thức jspDestroy() đeer giải phóng mã trang ra khỏi bộ nhớ.

1. Cơ chế hoạt động của trang JSP

Servlet đưa mã HTML vào lệnh Java trong khi ngược lại JSP đưa lệnh Java vào các mã HTML. Các trang JSP chứa các thẻ đặc biệt quy định gaafn giống thẻ của ngôn ngữ HTML. Khi từ trình duyệt gửi yêu cầu một trang JSP, trình chủ Web Server sẽ đọc trang JSP từ đĩa cứng, bộ diễn dịch JSP Engine (thường được cài đặt trong các trình chủ Web Server) sẽ diễn dịch mã lệnh Java chứa trong trang JSP thành một Servlet. Sau đó trình chủ sẽ triệu gọi Servlet trả kết xuất thuần HTML về cho trình khách. Cơ chế nảy được minh họa như sau:



*Hình 1.2 – Cơ chế triệu gọi trang JSP*

1. Các cú pháp cơ bản của JSP
2. Thẻ bọc mã <% %> hay <jsp:scriptlet>

Ưu điểm của trang JSP là khả năng nhúng mã Java giữa các thẻ định dạng HTML. Khi gặp thẻ <% bộ diễn địch JSP sẽ biết được lệnh cân thực hiện tiếp theo là mã lệnh Java.

<html>

<%

mã Java

%>

</html>

Ví dụ: <html>

<% // Dưới đây là mã lệnh java

out.println ( “Xin chào các bạn.”);

%>

</html>

Cú pháp <% %> vẫn thường được sử dụng hơn

1. Hiễn thị kết xuất bằng cú pháp <%= %>

Thay vì sử dụng cú pháp <% %> đẻ diễn đạt một khối gồm nhiều lệnh ta có thể sử dụng cú pháp <%= %> chỉ đẻ hiển thị kết xuất của một giá trị biến hay một biêu thức, hàm nào đó.

Ví dụ: <html>

Xin chào <%=name%>

Bạn có <%= getMessage() %> hôm nay.

</html>

Trong đó name là biến, getMessage() là một hàm trả về số kiểu int. JSP không dùng dấu (;) ở cuối các biến hoặc biểu thức gọi hàm trong cú pháp <%= %>, bởi vì nội dung của cú pháp này sẽ được chuyển thành lệnh out.println() tương đương như sau: out.println(“Xin chào ”+ name);

1. Chèn chú thích vào mã trang JSP -

Cũng như Java, JSP cho phép ta dùng cú pháp // để chú thích một dòng mã lệnh và cú pháp /\* \*/ được áp dụng cho nhiều dòng. Các dòng chú thích sẽ được bỏ qua khi trình chủ diễn dịch trang JSP.

// Chú thích một dòng trong JSP

/\* Chú thích nhiều

dòng trong JSP

\*/

Ngoài ra JSP còn cung cấp thêm cho ta cú pháp chú thích <%-- --%>. Tất cá khối lệnh Java và HTML nằm giữa hai dấu chú thích này sẽ được trình biên dịch bỏ qua.

1. Khai báo phương thức và biến bằng <%! %>

Một cú pháp sau cùng mà JSP cung cấp là <%! %>. Cú pháp này cho phép ta khai báo một hoặc nhiều phương thức và biến. Phương thức và biến sau đó có thể được triệu gọi bất kỳ đâu trong trang JSP.

Ví dụ: <html>

<%! String name = “John Wick”; // Khai báo biến name %>

<%! int age = 35; %>

</html>

1. Các thẻ xử lý nhúng và chuyển hướng giữa các trang

Trong quá trình xử lý và chuyển hướng dữ liệu cho form ta có thể nhúng và triệu gọi những trang JSP khác với trang hiện hành. Hay khi nhận dữ liệu submit từ trình khách, trang JSP nhận được dữ liệu có thể chuyển hướng hoặc triệu gọi đến trang JSP khác. JSP cung cấp cho ta các thẻ sau:

1. Thẻ nhúng mã nguồn <%@ include file %>

Thẻ này thường dùng để đem nội dung file.html hay file.jsp bên ngoài vào trang hiện hành. Cú pháp như sau:

<%@ include file = “filename” %>

Ví dụ: Trong ứng dụng ta thường sử dụng các hàm thư viện chứa trong file common.jsp. Thư viện này được đưa vào trang JSP hiện hành như sau:

<%@ include file = “common.jsp” %>

1. Thẻ <jsp:include>

Chỉ thị <%@ include %> chỉ dùng để nhúng các mã nguồn tĩnh. Nếu muốn nhúng kết quả kết xuất từ các trang jsp, servlet hay .html khác vào trang hiện hành JSP cung cấp cho ta thẻ <jsp:include> với cú pháp sau:

<jsp:include page = “filename” flush = “true” />

Ví dụ: Trong ứng dụng có sử dụng thẻ <jsp:include> để nhúng kết quả của trang cart.jsp vảo các trang khác như sau:

<jsp: include page = “cart.jsp” flush = “true” />

1. Chuyển tham số bằng thẻ <jsp:param>

Khi nhúng trang bằng thẻ <jsp:include> ta có thể chuyển tham số cho trang được nhúng để nhận được kết quả kết xuất linh động hơn bằng cách sử dụng thẻ con <jsp:param>.

Ví dụ: Ta có hai trang test1.jsp và trang test2.jsp và truyền tham số “Xin chào.” cho trang test2.jsp như sau:

Trang test1.jsp

<jsp:include page = “test2.jsp” flush = “true” />

<jsp:param name = “say” value = “Xin chào.” />

<jsp:include>

<%

out.println (“Trang testl.jsp<br>”);

%>

Trang test2.jsp

<%

String param = request.getPararmeter(“say”);

out.println(“Trang test2.jsp <br>”);

out.println(“param : “ + param);

%>

1. Thẻ chỉ dẫn biên dịch trang <%@ page %>

Thẻ <%@ page %> chỉ dẫn một số tính chất biên dịch áp dụng cho trang jsp. Ta có thể sử dụng ther này đề khai báo các thư viện import của Java, chỉ định tùy chọn trang jsp có cần giữ trên cache bộ nhớ của trình chủ để tăng tốc hay không...

<%@ page attribute = “value” … %>

Ví dụ: Để khai báo sử dụng các thư viện java.

<html>

<%@ page import = “java.sql.\*” %>

</html>

1. Thẻ chuyển hướng đến trang khác <jsp:forward>

Thẻ này giúp ta triệu gọi và chuyển hướng trang web sang địa chỉ khác hoặc chuyển dữ liệu cho trang jsp khác xử lý.

Cú pháp: <jsp:forward page="filename” />

Ví dụ: Khi xử lý trang dữ liệu đăng nhập (Login page) ta kiểm tra mật khẩu, nếu hợp lệ ta chuyển người dùng đến trang tài nguyên cho phép truy cập. Nếu không hợp lệ, chuyển người dùng đến trang thông báo lỗi.

<%

if (request.getParameter(“password”) == "123456”) {

%>

<jsp:forward page = “errorLogin.jsp” >

</jsp:forward>

<%

} else {

%>

<jsp:forward page = “successLogin.jsp” >

</jsp:forward>

<%

)

%>

1. Chuyển hướng sang trang mới với sendRedirect()

Phương thức sendRedirect() cũng có chức năng giống như thẻ <jsp:forward>. Nhưng nó khác nhau ở chỗ là sendRedirect() chỉ thực hiện triệu gọi trang mà không chuyển tham số của trang hiện hành cho trang chuyển hướng.

Ví dụ:

<% response.sendRedirect(“shop.jsp”); %>

1. Thẻ sử dụng thành phần Bean <jsp: useBean>

Thẻ <jsp: useBean> được dùng để khai báo phạm vi và định danh id (identify) nhận dạng Bean.

Thẻ này khá linh động, ngữ nghĩa chính xác phụ thuộc vào giá trị đưa ra. Cú pháp của <jsp: useBean> như sau:

<jsp: useBean id = “name” scope = “page | request | session | application” class = “packagename.classname”>

Ví dụ: <jsp: useBean id = “cart” scope = “session” class = “gioHang”/>

1. Thẻ đặt thuộc tính cho Bean <jsp:setProperty>

Thẻ nảy dùng để gán giá trị vào thuộc tính Bean. Thuộc tính tên của Bean chỉ định cho đối tượng phải được định nghĩa và nằm trong phạm vi cho phép. Cú pháp của <jsp:setProperty> như sau:

<jsp:setProperty name= “beanName” prop\_expr />

Trong đó prop\_expr có thể khai báo như sau:

property = “\*” |

property = “propertyName” |

property = “propertyName” param = “parameterName” |

property = “propertyName” value = “propertyValue” |

1. Thẻ lấy thuộc tính cho Bean <jsp:getProperty>

Thẻ này dùng để lấy giá trị của thuộc tính Bean và chuyển giá trị thành kiểu chuỗi. Cú pháp cho thẻ <jsp:getProperty> như sau:

<jsp: getProperty name = “name” property = “propertyName`” />

name: thuộc tính trình bày tên của bean.

property: Thuộc tính của bean mà ta cần lấy giá trị.

1. Sử dụng các đối tượng trong trang JSP

Trình diễn dịch JSP cho phép ta sử dụng một số đối tượng đã khai báo trước. Điều này giúp ta viết mã lệnh trong trang JSP nhanh hơn Servlet.

* Đối tượng **out**: xuất phát từ lớp PrintWriter. Đỗi tượng này được dùng đẻ định dạng kết xuất gởi về máy khách.

Cú pháp: *out.Phươngthức(các biến...)*

Một số phương thức của đối tượng: clear(); clearBuffer(); flush();...

* Đối tượng **request**: xuất phát tử lớp HttpServletRequest. Đối tượng này dùng để lấy về các tham số hay dữ liệu do trình khách chuyển lên.

Cú pháp: *request. Phươngthức(các biến...)*

Một số phương thức của đối tượng: getParameter(String name), getAtribute(String name); getMethod();...

* Đối tượng **response**: Xuất phát từ lớp javax.servlet.HttpServletResponse, tương tự đối tượng **out**, đối tượng **response** dùng để đưa kết xuất trả về trình khách. Tuy nhiên đối tượng **out** được đùng thường xuyên hơn do được hỗ trợ thêm luồng đệm để tăng tốc kết xuất.

Cú pháp: *response.Phươngthức(các biến...)*

Một số phương thức của đối tượng: addCookie(Cookie cookie); sendError(int sc); sendRedirect();

* Đối tượng **session**: Xuất phát tử lớp javax.servlet.http.HttpSession, đối tượng này dùng để theo dõi kết nối và lưu vết một phiên làm việc giữa trình khách và trình chủ.

Cú pháp: *session.Phươngthức(các biến...)*

Một số phương thức của đối tượng: getAttribute(String name); getAtributeNames(); getId();

* Đối tượng **application**: Xuất phát từ lớp javax.servlet.ServletContext trong java. Có thể dùng đối tượng này để chia sẻ thông tin giữa tất cả người sử dụng, lưu các thông tin mang tính bao quát lên ứng dụng JSP. **Application** cũng được dùng để đếm số lần truy cập của người sử dụng.

Cú pháp: *application.Phươngthức(các biến...)*

Một số phương thức của đối tượng: getAttribute(String name); getServerInfo(); getAttributeNames();

1. **GIỚI THIỆU HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MYSQL**
2. Giới thiệu MySQL

MySQL là hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh với tốc độ và tính bảo mật cao.

MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên Internet. MySQL miễn phí hoàn toàn cho nên bạn có thẻ tải MySQL từ trang chủ. Nó có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32, Win64 cho các hệ điều hành dòng Windows, Linux, MacOS, SunOS, Unix, FreeBSD, ...

MySQL là một trong những ví dụ rất cơ bản về Hệ quản trị cơ sở dữ liệu quan hệ sử dụng Ngôn ngữ truy vấn có cầu trúc (SQL).

MySQL được sử dụng cho việc bổ trợ JAVA, PHP, Perl và nhiều ngôn ngữ khác. Nó làm nơi lưu trữ những thông tin trên các trang web viết bằng JAVA, PHP hay Perl, ...

1. Mục đích sử dụng cơ sở dữ liệu

Mục đích sử dụng cơ sở dữ liệu bao gồm các chức năng như: lưu trữ (storage), truy cập (accessibility), tổ chức (organization) và xử lí (manipulation).

* *Lưu trữ:* Lưu trữ trên đĩa và có thể chuyển đổi dữ liệu từ cơ sở dữ liệu này sang cơ sở dữ liệu khác, nếu bạn sử dụng cho quy mô nhỏ, bạn có thể chọn cơ sở đữ liệu nhỏ như: Microsoft Excel, Microsoft Access, MySQL, Microsoft Visual FoxPro, ... Nếu ứng dụng có quy mô lớn, bạn có thể chọn cơ sở dữ liệu có quy mô lớn như: Oracle, SQL Server, ...
* *Truy cập:* Truy cập dữ liệu phụ thuộc vào mục đích và yêu cầu của người sử dụng, ở mức độ mang tính cục bộ, truy cập cơ sở dữ liệu ngay trong cơ sở dữ liệu với nhau, nhằm trao đổi hay xử lí dữ liệu ngay bên trong chính nó. Nhưng do mục đích và yêu cầu người dùng vượt ra ngoài cơ sở dữ liệu, nên bạn cần có các phương thức truy cập dữ liệu giữa các cơ sở dữ liệu với nhau như: Microsoft Access với SQL Server, hay SQL Server và cơ sở dữ liệu Oracle, ...
* *Tổ chức:* Tổ chức cơ sở dữ liệu phụ thuộc vào mô hình cơ sở dữ liệu, phân tích và thiết kế cơ sở dữ liệu tức là tổ chức cơ sở dữ liệu phụ thuộc vào đặc điểm riêng của từng ứng dụng. Tuy nhiên khi tổ chức cơ sở dữ liệu cần phải tuân theo một số tiêu chuẩn của hệ thống cơ sở dữ liệu nhằm tăng tính tối ưu khi truy cập và xử lí.
* *Xử lí:* Tùy vào nhu cầu tính toán và truy vấn cơ sở dữ liệu với các mục đích khác nhau, cần phải sử dụng các phát biểu truy vấn cùng các phép toán, phát biểu của cơ sở dữ liệu để xuất ra kết quả như yêu cầu. Để thao tác hay xử lí dữ liệu bên trong chính cơ sở dữ liệu ta sử dụng các ngôn ngữ lập trình như: JAVA, PHP, C++, Visual Basic, ...

# CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

# CHƯƠNG 3. CÀI ĐẶT

# KẾT LUẬN