



BUỔI 2 - JSP

CYBERSOFT.EDU.VN



❑ JSP

- ✓ Tìm hiểu JSP
- ✓ So sánh JSP và Servlet
- ✓ Vòng đời JSP
- ✓ JSP Form
- ✓ Sử dụng Expression Language
- ✓ Sử dụng thư viện JSTL

❑ Dự án CRM

- ✓ Tạo cấu trúc dự án
- ✓ Xây dựng giao diện trang chủ
- ✓ Xây dựng giao diện trang quản lý thành viên.
- ✓ Xây dựng giao diện trang quản lý công việc.

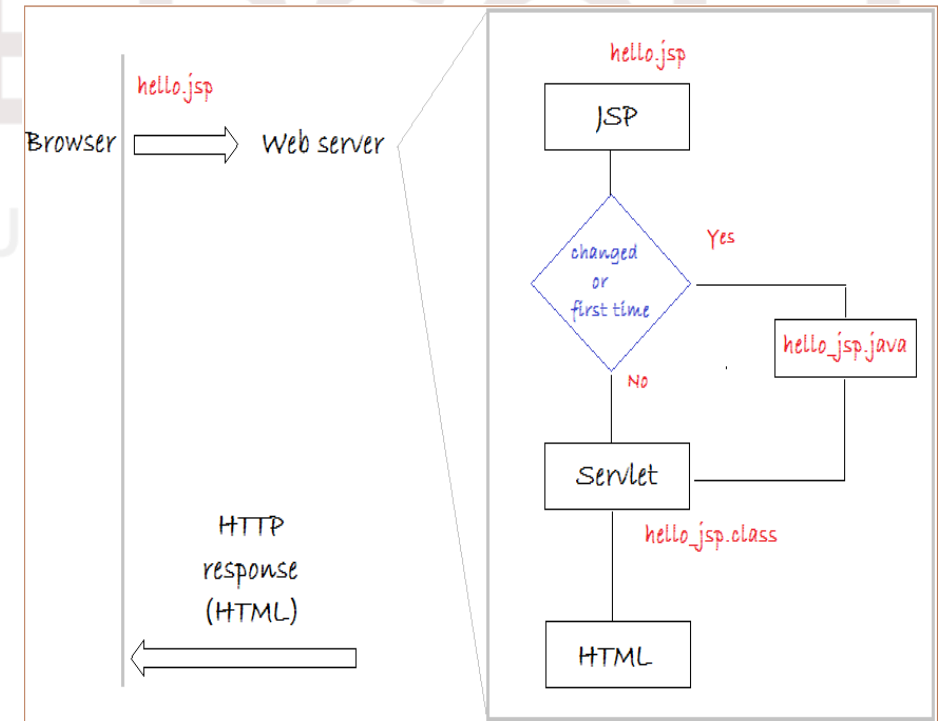
JSP là gì?

- ❑ **JSP** (JavaServer Pages) là công nghệ được sử dụng để tạo ra các ứng dụng web động.
- ❑ **JSP** giúp chúng ta dễ dàng viết các mã code Html, Css và Javascript hơn.
- ❑ Với **JSP** lập trình viên có thể thêm code Java trong code Html.
- ❑ Nhúng code Java vào Html trong JSP bằng cách sử dụng thẻ **Scriptlet**(<% %>).
- ❑ File **JSP** có phần mở rộng là **.jsp**.
- ❑ Khi chạy, **JSP** được chuyển đổi thành **Servlet** rồi sau đó mới được thực thi.

- ☐ Dễ dàng viết các mã code Html.
- ☐ Hiệu suất cao và có khả năng mở rộng.
- ☐ Các trang JSP có thể truy cập trực tiếp mà không cần cấu hình trong file web.xml hoặc thông qua Annotation @WebServlet như Servlet.
- ☐ Bất kỳ sự thay đổi trong mã JSP được xử lý bởi Container Web nên chúng ta không cần khởi động lại Tomcat mà chỉ cần F5 lại trang thì chương trình sẽ được biên dịch.

Mối quan hệ JSP - Servlet

- ❑ Trong lần chạy đầu tiên **Web Server** chuyển file **hello.jsp** thành file **hello_jsp.java** và dịch nó thành file **hello_java.class** (Servlet), nó sẽ tạo ra Html trả về phía người dùng.
- ❑ Trong lần chạy thứ 2 trở đi, **Web Server** sẽ kiểm tra file **hello.jsp** nếu không có gì thay đổi thì gọi tới Servlet **hello_java.class** và trả Html về phía người dùng. Nếu có thay đổi **Web Server** sẽ tạo lại file **hello_jsp.java** và biên dịch thành file **hello_java.class**.



JSP – Biên dịch

```
<html>
  <head>
    <title>My First JSP Page</title>
  </head>
  <%
    int count = 0;
  %>
  <body>
    Page Count is:
    <% out.println(++count); %>
  </body>
</html>
```

ĐÀO T



```
public class hello_jsp extends HttpServlet
{
    public void _jspService(HttpServletRequest request, HttpServletResponse response)
        throws IOException, ServletException
    {
        PrintWriter out = response.getWriter();
        response.setContentType("text/html");
        out.write("<html><body>");
        int count=0;
        out.write("Page count is:");
        out.print(++count);
        out.write("</body></html>");

    }
}
```

JSP TAG

❑ JSP Comment

❖ Cú pháp:

`<%-- Nội dung --%>`

- Sử dụng để **ghi chú**, chương trình sẽ bỏ qua các nội dung được viết bên trong cặp thẻ này.

❖ Ví dụ:

```
<body>

    <%!
        int count = 0;
    %>

    <%-- Nội dung cần ghi chú ở đây --%>

    <h2>
        Số cần hiển thị là:
    <%
        out.println(++count);
    %>

</h2>

</body>
```

❑ Scriptlet Tag

❖ Cú pháp:

`<% Code Java %>`

- Sử dụng để viết code Java trong file JSP.

❖ Ví dụ:

```
<body>

    <%!
        int count = 10;
    %>

    <%-- Nội dung cần ghi chú ở đây --%>

    <ul>

        <%
            for (int i = 0; i < count; i++) {
        %>

        <li><%= i + 1 %></li>

        <%
            }
        %>

    </ul>

</body>
```

JSP TAG

❑ Declaration Tag

❖ Cú pháp:

`<%! Code Java %>`

➤ Được sử dụng để **khai báo biến** và **phương thức** trong file JSP.

❖ Ví dụ:

```
<body>

<%!int count = 0;
    int getCount() {
        System.out.println("In getCount() method");
        return count;
    }
%>

<h1>
    <%
        out.println(getCount());
    %>
</h1>

</body>
```

❑ Directive Tag

❖ Cú pháp:

`<%@ Code %>`

➤ Định nghĩa một số phụ thuộc trang.

❖ Ví dụ:

```
<%@page import="java.util.ArrayList"%>
<%@page import="models.Task"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
```

```
<body>

    <%@ include file = "header.jsp" %>

    <h1>Nội dung body</h1>

    <%@ include file = "footer.jsp" %>

</body>
```


Implicit Objects JSP

IMPLICIT OBJECT	DESCRIPTION
request	Đối tượng HttpServletRequest được sử dụng liên quan đến việc request.
response	Đối tượng HttpServletResponse được sử dụng liên kết với các phản ứng đó response lại cho trình duyệt.
out	Đối tượng JspWriter được sử dụng liên quan đến dòng đầu ra của phản ứng.
session	Đối tượng HttpSession được sử dụng liên kết với các phiên làm việc cho người sử dụng được request.
application	Đối tượng ServletContext được sử dụng cho các ứng dụng web.
config	Đối tượng ServletConfig liên kết với các servlet cho trang JSP hiện tại.
pageContext	Đối tượng pageContext mà gói gọn với môi trường của một yêu cầu duy nhất cho trang JSP hiện tại
page	Được sử dụng để gọi các phương thức được định nghĩa bởi lớp Servlet đã được biên dịch
exception	Đối tượng Exception cho phép dữ liệu exception để được truy cập bởi JSP đã chỉ rõ

Jsp Form

login.jsp

```
<div style="width:500px; margin:auto;">
  <h2>ĐĂNG NHẬP</h2>
  <% String path = request.getContextPath(); %>

  <form action="<%= path %>/login" method="post">

    <label>Tên đăng nhập</label>
    <input type="text" name="username">

    <label>Tên đăng nhập</label>
    <input type="password" name="password">

    <button type="submit">Đăng nhập</button>
  </form>
</div>
```

info.jsp

```
<body>
  <h2>THÔNG TIN ĐĂNG NHẬP</h2>
  <p>Tên đăng nhập
    <%= request.getAttribute("name") %>
  </p>
  <p>Mật khẩu
    <%= request.getAttribute("pass") %>
  </p>
</body>
```

LoginServlet

```
@WebServlet(urlPatterns = "/login")
public class LoginServlet extends HttpServlet{

    protected void doGet(HttpServletRequest req, HttpServletResponse resp) {}

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {

        String tenDN = req.getParameter("username");
        String matKhai = req.getParameter("password");

        req.setAttribute("name", tenDN);
        req.setAttribute("pass", matKhai);
        req.getRequestDispatcher("info.jsp").forward(req, resp);
    }
}
```

Bài tập 2:

Sử dụng Servlet tạo trang đăng nhập và trang chào mừng.

- ✓ Thiết kế form cho trang đăng nhập cho phép người dùng nhập thông tin username và password.
- ✓ Submit form bằng phương thức POST.
 - + Nếu username = 'admin' và password = '123456' thì chuyển qua trang welcome xuất thông báo 'Chào mừng ...'
 - + Nếu sai thì xuất thông báo lỗi 'Sai username hoặc mật khẩu'.

- ❖ **Bước 1:** Tạo dự án có tên **BaiTapLogin**.
- ❖ **Bước 2:** Tạo **Servlet Login** và **Welcome**.
- ❖ **Bước 3:** Tạo trang **login.jsp** và **welcome.jsp**.
- ❖ **Bước 4:** Cấu hình **Web Container**.
- ❖ **Bước 5:** Trong **Servlet Login**, override lại phương thức **getPost** → Sử dụng **request.getParameter** để lấy dữ liệu người dùng gửi lên → forward dữ liệu qua **Servlet Welcome**.
- ❖ **Bước 6:** Trang **welcome.jsp** hiển thị thẻ **h1** có nội dung “Xin chào ” + **username** và **password** do người dùng gửi lên.

Cấu trúc thư mục



- ▼ Bai01
 - > Deployment Descriptor: Bai01
 - > JAX-WS Web Services
 - ▼ Java Resources
 - ▼ src
 - ▼ com.myclass.servlet
 - > LoginServlet.java
 - > Libraries
 - > JavaScript Resources
 - > build
 - ▼ WebContent
 - > META-INF
 - > WEB-INF
 - login.jsp
 - welcome.jsp

SOFT
GIA LẬP TRÌNH

LoginServlet

```
@WebServlet(urlPatterns = "/login")
public class LoginServlet extends HttpServlet{
    private static final long serialVersionUID = 1L;

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {

        req.getRequestDispatcher("login.jsp").forward(req, resp);
    }

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {
        resp.setCharacterEncoding("UTF-8");
        String username = req.getParameter("username");
        String password = req.getParameter("password");

        if(username.equals("admin") && password.equals("123456")) {
            req.setAttribute("name", username);
            req.getRequestDispatcher("welcome.jsp").forward(req, resp);
        }else {
            req.setAttribute("message", "Sai tên username hoặc password");
            req.getRequestDispatcher("login.jsp").forward(req, resp);
        }
    }
}
```

Login.jsp

```
<div class="container mt-5">
  <div class="row">
    <div class="col-md-6 m-auto">
      <h3 class="text-center">FORM LOGIN</h3>
      <p class="text-center text-danger">${ message }</p>
      <form action="<%=request.getContextPath()%>/login" method="post">
        <div class="form-group">
          <label>Username</label> <input type="text" name="username"
            class="form-control">
        </div>
        <div class="form-group">
          <label>Password</label> <input type="password" name="password"
            class="form-control">
        </div>
        <button class="btn btn-success">Login</button>
      </form>
    </div>
  </div>
</div>
```

Welcome.jsp

```
<body>
  <h2>Xin chào ${ name }</h2>
</body>
```

❑ Expression Language

`${ Code Java }`

- ✓ **JSP EL** là một cách **truy xuất vào dữ liệu của đối tượng** trong file JSP.
- ✓ **EL** cho phép thực hiện **tính toán, so sánh và kiểm tra**.
- ✓ Mục đích **tạo cách viết ngắn gọn, rõ ràng và nhanh chóng** thay cho việc dùng **`<%= %>`** để in ra giá trị của code Java.

`<%= fullname %>`



`${ fullname }`

Truy xuất thuộc tính Bean

HomeServlet

```
@Override
protected void doGet(HttpServletRequest req, HttpServletResponse resp)
    throws ServletException, IOException {

    Product product = new Product("Điện thoại Nokia", 20, 1500000);
    req.setAttribute("mobile", product);
    req.getRequestDispatcher("home.jsp").forward(req, resp);
}
```

Java Bean

```
public class Product {
    private String name;
    private int quantity;
    private int price;

    // get/set
}
```



home.jsp

```
<ul>
    <li>Tên SP:    ${ mobile.name }    </li>
    <li>Số lượng:  ${ mobile.quantity } </li>
    <li>Giá bán:   ${ mobile.price }    </li>
</ul>
```

Truy xuất mảng/tập hợp

HomeServlet

```
@Override
protected void doGet(HttpServletRequest req, HttpServletResponse resp)
    throws ServletException, IOException {

    List<Product> products = new ArrayList<Product>();
    products.add(new Product("Điện thoại Nokia", 20, 1500000));
    products.add(new Product("Điện thoại Samsung", 50, 5500000));

    req.setAttribute("products", products);

    req.getRequestDispatcher("home.jsp").forward(req, resp);
}
```



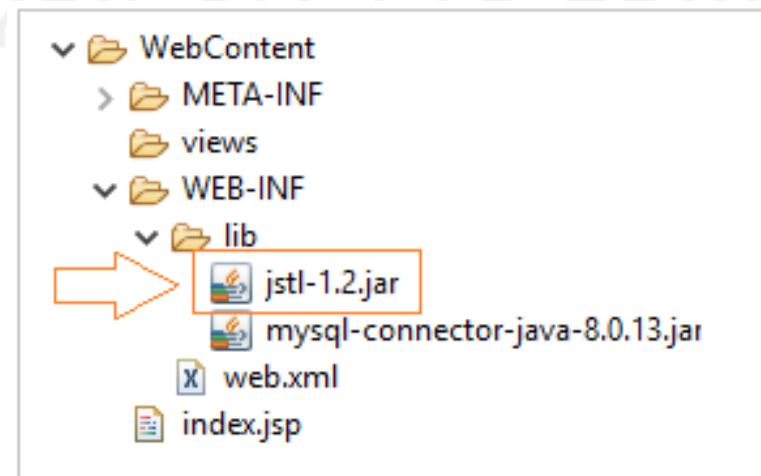
home.jsp

```
<ul>
    <li>Tên SP:    ${ products[0].name }    </li>
    <li>Số lượng:  ${ products[0].quantity } </li>
    <li>Giá bán:   ${ products[0].price }    </li>
</ul>
```

Thư viện JSTL

- **JSTL** là **thư viện hỗ trợ các thẻ** (tag) trong file JSP **giúp tính toán** và **hiển thị logic thay cho code Java** với mục đích tối ưu lập trình giao diện Jsp.
- **JSTL** giúp code **dễ đọc, rõ ràng, dễ hiểu** và **giúp giảm thiểu code Java** trong JSP.
- Để sử dụng **JSTL** phải **download thư viện** và import vào project.
Download <https://mvnrepository.com/artifact/javax.servlet/jstl/1.2>

License	CDDL GPL 2.0
Categories	Java Specifications Tải về
Date	(Jun 23, 2011)
Files	pom (356 bytes) jar (404 KB) View All
Repositories	Central Redhat GA WSO2 Dist
Used By	1,530 artifacts



Thư viện JSTL

❑ Trong JSTL có rất nhiều bộ thẻ để xử lý các vấn đề khác nhau

❖ **Core:** chứa các thẻ lệnh điều khiển cơ bản.

`<%@ taglib uri="http://java.sun.com/jstl/core" prefix="c" %>`

❖ **Format:** chứa các thẻ định dạng và đa ngôn ngữ.

`<%@ taglib uri="http://java.sun.com/jstl/fmt" prefix="fmt" %>`

❖ **Function:** cung cấp các hàm hỗ trợ cho EL.

`<%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>`

❖ **Sql:** chứa các thẻ làm việc với CSDL.

❖ **Xml:** chứa các thẻ xử lý xml.

- ❑ **Core** chứa các **thẻ thay thế các lệnh cơ bản trong Java** để phù hợp với lập trình giao diện theo cú pháp thẻ.

<c:if>:

Tương tự **lệnh if** trong java.

<c:choose>:

Tương tự **if...else if...else** trong java.

<c:forEach>:

Tương tự **for-each** trong java.

<c:url>:

Set giá trị cho biến bằng **url tương đối**.

Thẻ <c:if >

❑ Cú pháp

<c:if test="{\$<điều kiện>}">Nội dung</c:if>

❑ Ví dụ:

```
<ul>
  <li>Họ và tên: ${student.name}</li>
  <li>Điểm TB: ${student.mark}</li>
  <li>Chuyên ngành: ${student.major}</li>
  <c:if test="${student.mark > 9}">
    <li><strong>Đanh hiệu ong vàng</strong></li>
  </c:if>
</ul>
```



- Họ và tên: Phương
- Điểm TB: 10.0
- Chuyên ngành: APP
- **Danh hiệu ong vàng**

Thẻ <c:choose >

❑ Cú pháp

<c:choose>

<c:when test="\$<ĐK 1>">Nội dung 1</c:when>

<c:otherwise>Nội dung 2</c:otherwise>

</c:choose>

❑ Ví dụ

```
<li>Xếp loại:  
  <c:choose>  
    <c:when test="$<student.mark < 5>">Yếu/kém</c:when>  
    <c:when test="$<student.mark > 9>">Giỏi</c:when>  
    <c:when test="$<student.mark > 6.5>">Khá</c:when>  
    <c:otherwise>Trung bình</c:otherwise>  
  </c:choose>  
</li>
```

TRÌNH

Thẻ <c:forEach>

❑ Cú pháp

<c:forEach

var=*“biến chứa phần tử hiện tại”*

items=*“tập hợp các phần tử”*

begin=*“vị trí của phần tử bắt đầu mặc định là 0”*

end=*“vị trí của phần tử cuối cùng mặc định là vị trí phần tử cuối cùng”*

varStatus=*“biến trạng thái”*>

Nội dung

</c:forEach>

Thẻ <c:forEach>

```
<table border="1" style="width:100%">
<tr>
  <th>Họ và tên</th>
  <th>Điểm</th>
  <th>Chuyên ngành</th>
  <th>Xếp loại</th>
</tr>
<c:forEach var="student" items="${students}">
  <tr>
    <td>${student.name}</td>
    <td>${student.mark}</td>
    <td>${student.major}</td>
    <td>
      <c:choose>
        <c:when test="${student.mark < 5}">Yếu/kém</c:when>
        <c:when test="${student.mark > 9}">Giỏi</c:when>
        <c:when test="${student.mark > 6.5}">Khá</c:when>
        <c:otherwise>Trung bình</c:otherwise>
      </c:choose>
    </td>
  </tr>
</c:forEach>
</table>
```



Họ và tên	Điểm	Chuyên ngành	Xếp loại
Phương	10.0	BIZ	Giỏi
Cường	8.5	APP	Khá
Hạnh	3.5	MOB	Yếu/kém
Hương	5.5	MUL	Trung bình

Bài tập 1: Tạo dự án có tên **BaiTapJSP** với 2 Servlet có tên **Home** và **Product**.

- ✓ Tương ứng với mỗi Servlet là 2 file **home.jsp** và **product.jsp**.
- ✓ Tạo java bean **Product** với các thuộc tính: name, quantity, price.
- ✓ Trang **home.jsp**:
 - ✓ Thiết kế link để khi click vào sẽ chuyển sang trang sản phẩm.
- ✓ Trong trang **product.jsp**:
 - ✓ Thiết kế form thêm mới sản phẩm gồm các thông tin như trong Product bean.
 - ✓ Tạo giao diện load thông tin danh sách sản phẩm.
- ✓ Trong **ProductServlet**:
 - ✓ Phương thức doGet sử dụng để load danh sách sản phẩm.
 - ✓ Phương thức doPost sử dụng để thêm mới sản phẩm.

QUẢN LÝ SẢN PHẨM

Tên sản phẩm

Số lượng:

Giá bán:

Lưu lại

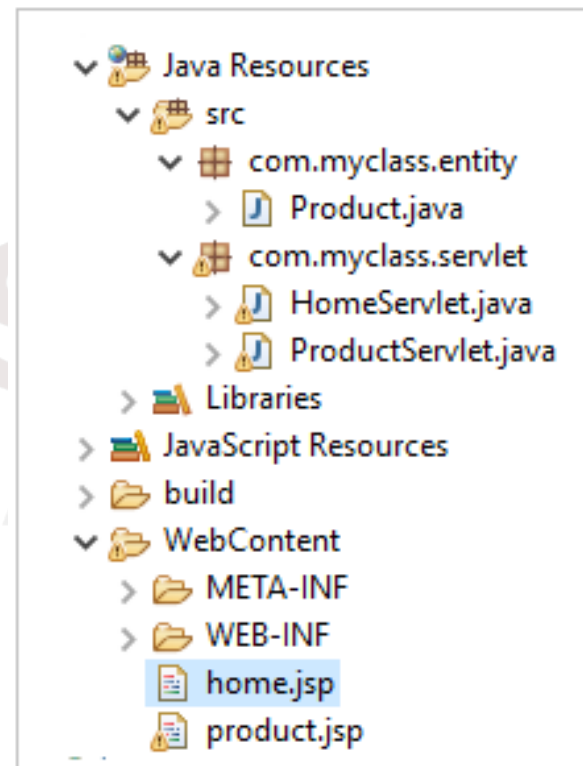
STT	Tên Sản Phẩm	Số Lượng	Giá Bán
1	Điện thoại Nokia	20	1500000
2	Điện thoại Samsung	50	5500000

Bài tập

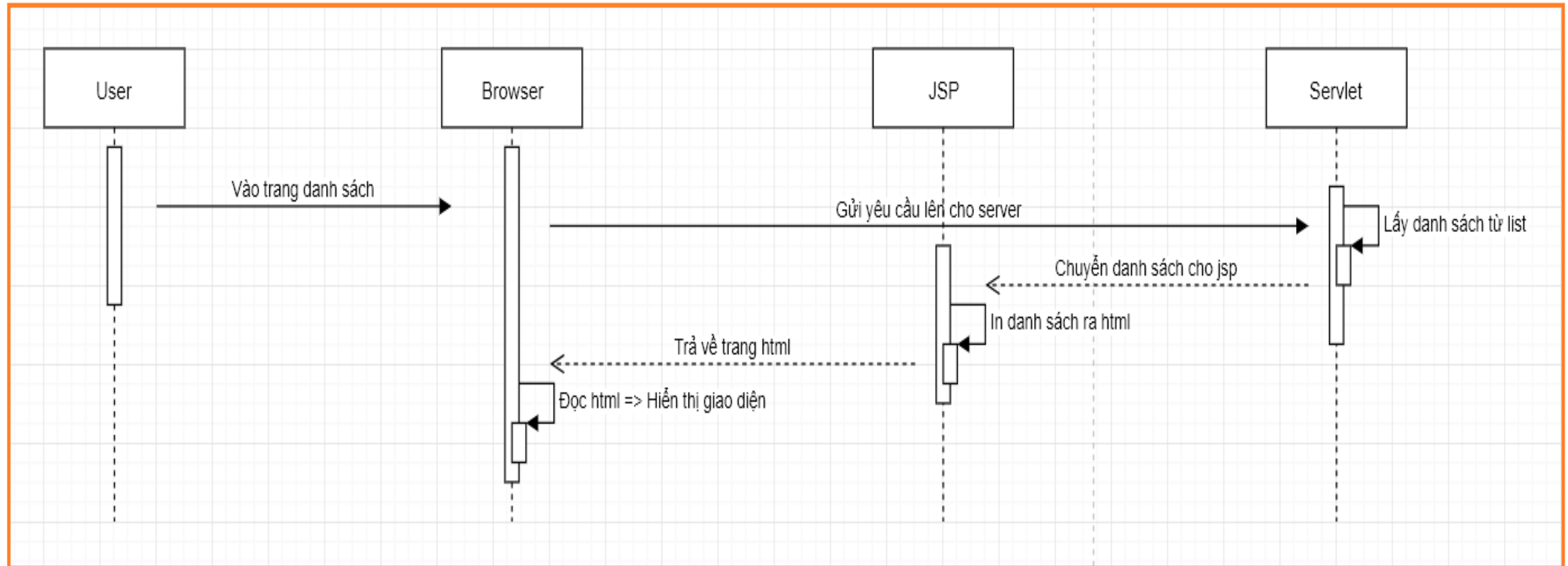
❑ Các bước thực hiện

- ✓ **Bước 1:** Tạo dự án
- ✓ **Bước 2:** Tạo HomeServlet và ProductServlet.
- ✓ **Bước 3:** Tạo home.jsp và product.jsp
- ✓ **Bước 4:** Cấu hình container.
- ✓ **Bước 5:** Tạo Product bean.
- ✓ **Bước 6:** Tạo danh sách Product, hiển thị ra giao diện.
- ✓ **Bước 7:** Tạo form thêm mới sản phẩm.
- ✓ **Bước 8:** Viết chức năng thêm mới sản phẩm.

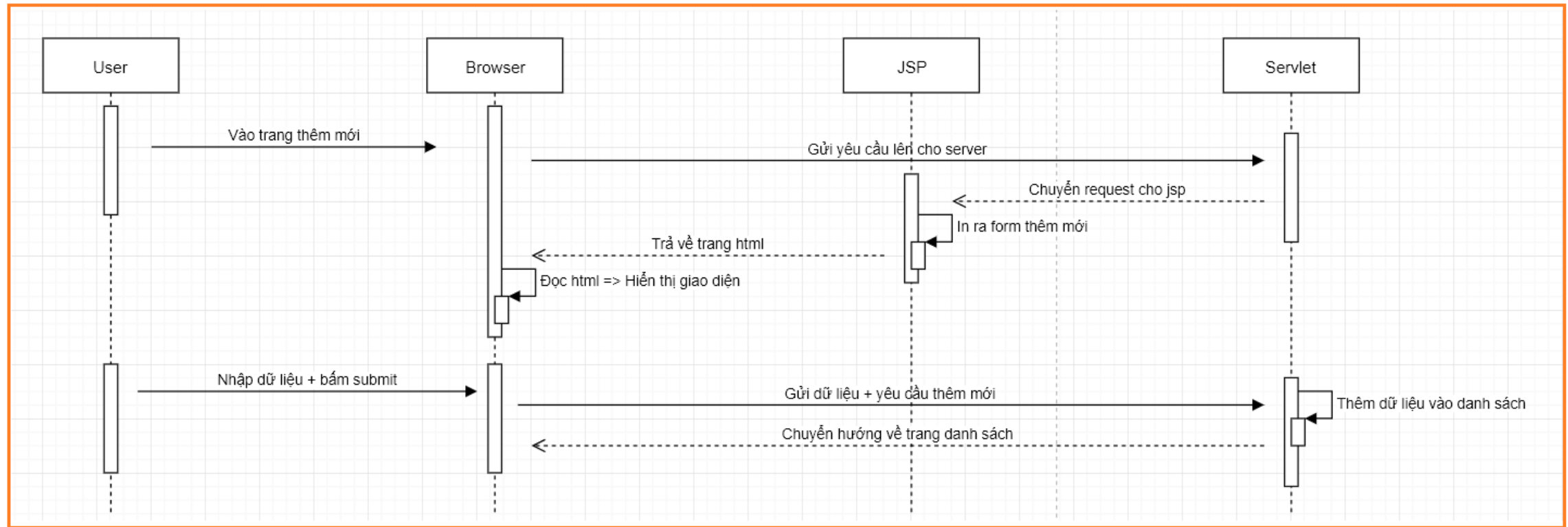
Cấu trúc thư mục



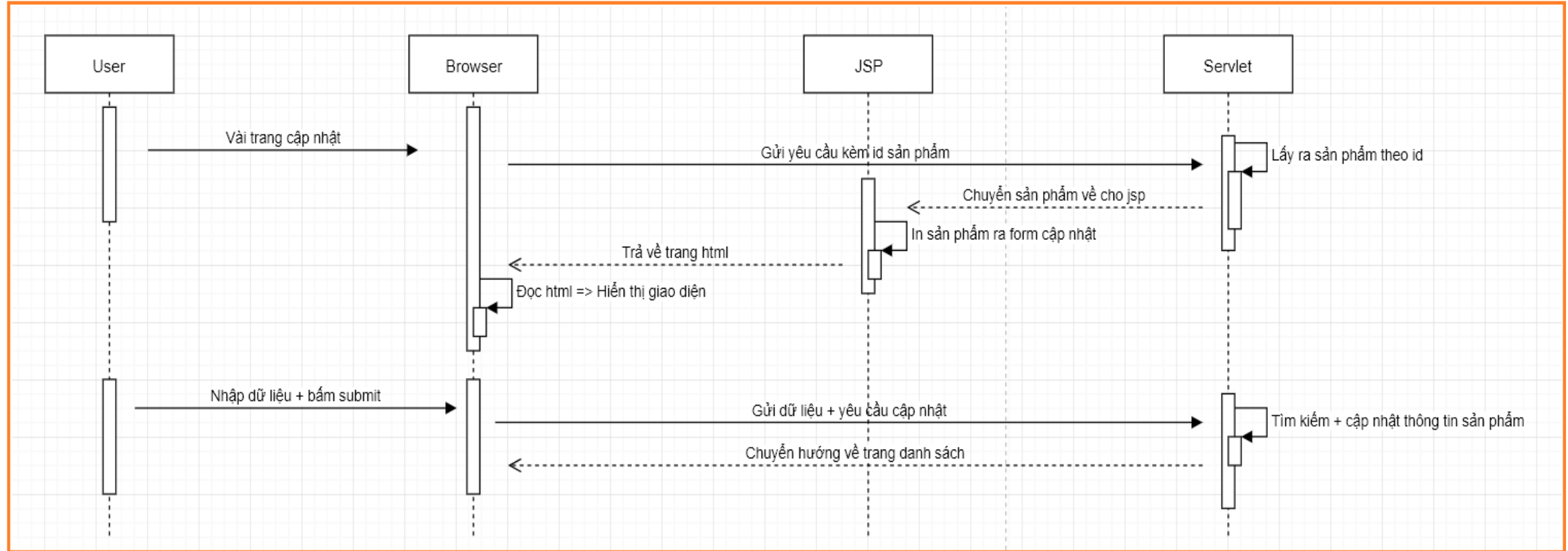
Lấy danh sách



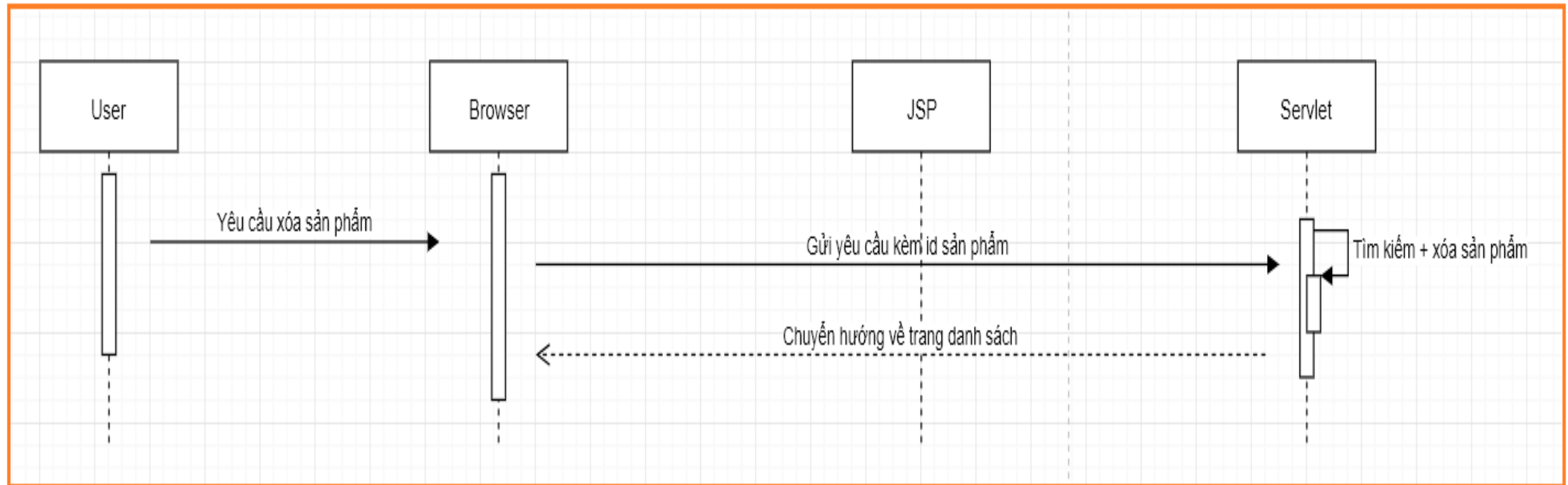
Thêm sản phẩm



Cập nhật sản phẩm



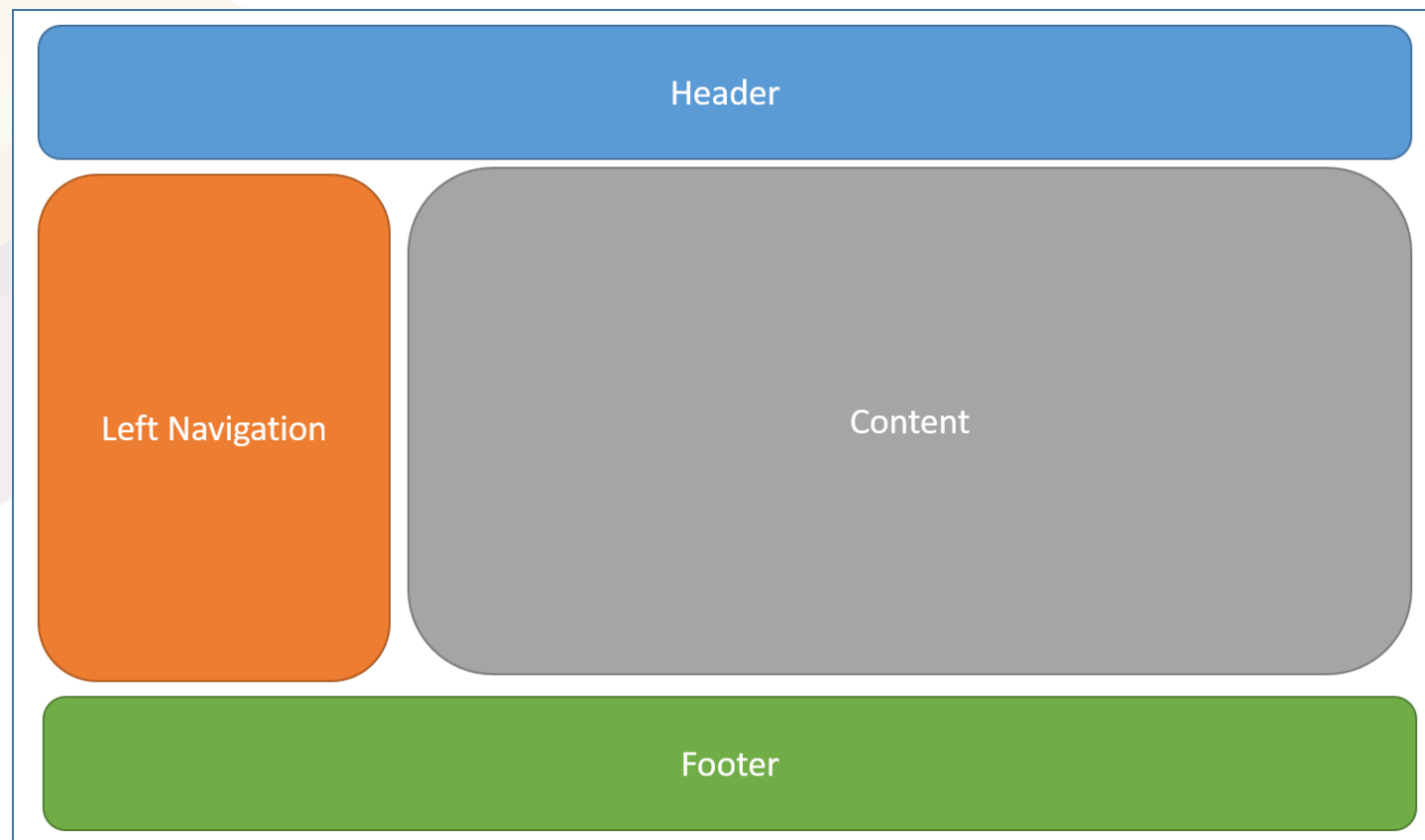
Xóa sản phẩm



Thư viện Sitemesh

❑ Sitemesh là gì?

- Sitemesh là thư viện giúp **tách các thành phần dùng chung** trong trang web thành các file riêng và **nối chúng lại với nhau khi cần đến**.




Thư viện Sitemesh

❑ Download thư viện

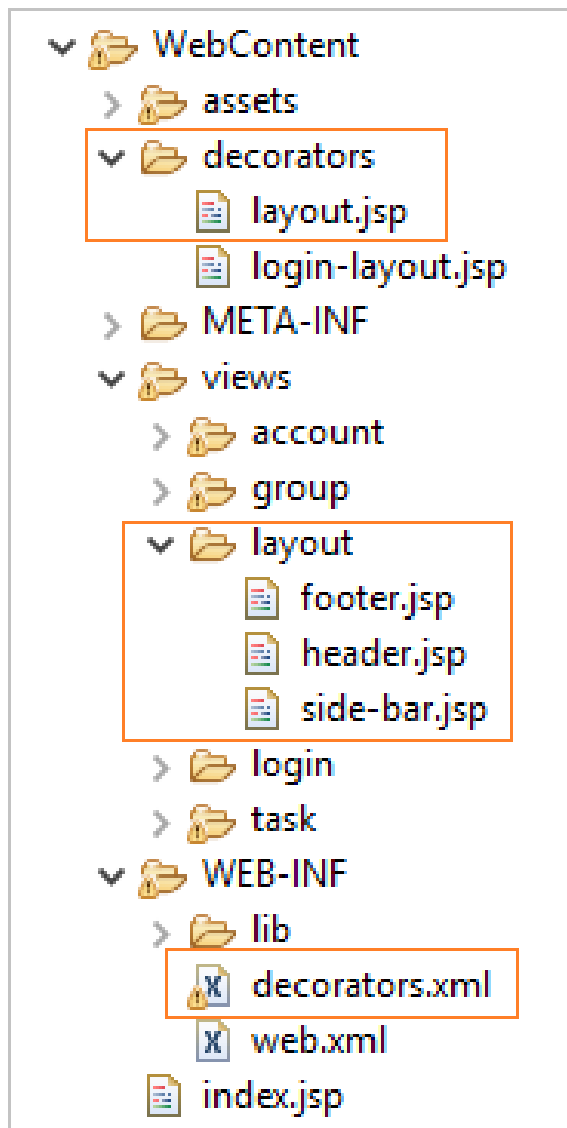
➤ Truy cập vào link

<https://mvnrepository.com/artifact/org.opensymphony/sitemesh/2.4.2>

License	Apache 1.1
Organization	Opensymphony
HomePage	http://www.opensymphony.com/sitemesh
Date	(Apr 25, 2009)
Files	pom (2 KB) jar (177 KB) View All
Repositories	Central
Used By	104 artifacts

 Tải về

Thư viện Sitemesh



- ❖ **Bước 1:** Trong thư mục **WebContent** tạo thư mục **decorators** và file **layout.jsp** để chứa layout cho website.
- ❖ **Bước 2:** Trong thư mục **views** tạo **thư mục layout** và **các file jsp** chứa thành phần dùng chung.
- ❖ **Bước 3:** Trong thư mục **WEB-INF** tạo file **decorators.xml** để cấu hình **Sitemesh**.
- ❖ **Bước 4:** Cấu hình Sitemesh trong file **web.xml**.

❖ WebContent --> decorators --> layout.jsp

Head

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@taglib uri="http://www.opensymphony.com/sitemesh/decorator"
    prefix="dec"%>
<%@taglib uri="http://java.sun.com/jstl/core" prefix="c"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title><dec:title /></title>
<meta content="width=device-width, initial-scale=1.0" name="viewport" />
<meta content="" name="description" />
<meta content="" name="author" />
<link href="assets/plugins/bootstrap/css/bootstrap.min.css"
    rel="stylesheet" />
<link href="assets/css/custom.css" rel="stylesheet" />
<dec:head />
</head>
<body class="fixed-top">
</body>
</html>
```

Body & Footer

```
<body class="fixed-top">
<jsp:include page="/views/layout/header.jsp" />
<!-- BEGIN CONTAINER -->
<div id="container" class="row-fluid">
    <!-- BEGIN SIDEBAR -->
    <jsp:include page="/views/layout/side-bar.jsp" />
    <!-- END SIDEBAR -->
    <!-- BEGIN PAGE -->
    <div id="main-content">
        <!-- BEGIN PAGE CONTAINER-->
        <div class="container-fluid">
            <dec:body />
        </div>
        <!-- END ADVANCED TABLE widget-->
    </div>
    <!-- END PAGE CONTAINER-->

    <!-- BEGIN FOOTER -->
    <jsp:include page="/views/layout/footer.jsp" />
    <!-- END FOOTER -->

    <!-- BEGIN JAVASCRIPTS -->
    <script src="assets/js/jquery-1.8.3.min.js"></script>
    <script src="assets/js/custom.js"></script>
    <!-- END JAVASCRIPTS -->
    <dec:getProperty property="page.scripts"></dec:getProperty>
</body>
</html>
```

Thư viện Sitemesh

❖ WEB-INF --> decorators.xml

Những Url bắt đầu bằng /auth sẽ load login-layout.jsp.

```
<?xml version="1.0" encoding="UTF-8"?>
<decorators defaultdir="/decorators">
  <decorator name="auth" page="login-layout.jsp">
    <pattern>/auth*</pattern>
  </decorator>
  <decorator name="web" page="layout.jsp">
    <pattern>/*</pattern>
  </decorator>
</decorators>
```

Những Url bắt đầu bằng / sẽ load trang layout.jsp.

Thư viện Sitemesh

❖ Cấu hình web.xml

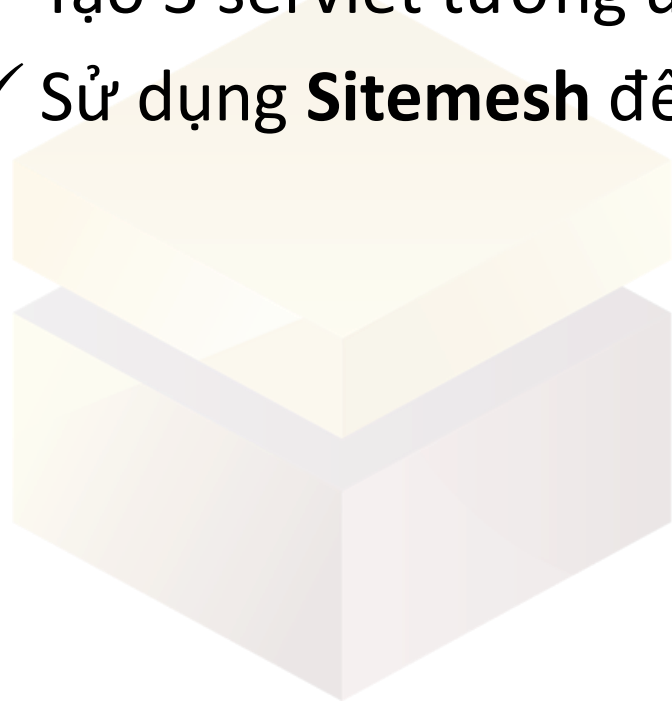
```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/
  id="WebApp_ID" version="3.0">
  <display-name>SERVLET_CRM</display-name>
  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>

  <!-- Cấu hình filter SiteMesh -->
  <filter>
    <filter-name>sitemesh</filter-name>
    <filter-class>com.opensymphony.sitemesh.webapp.SiteMeshFilter</filter-class>
  </filter>
  <filter-mapping>
    <filter-name>sitemesh</filter-name>
    <url-pattern>*</url-pattern>
  </filter-mapping>

</web-app>
```

Bài tập

- ❑ **Bài tập 3:** Tạo dự án có tên **SiteMeshDemo**
- ✓ Tạo 3 servlet tương ứng **HomeServlet**, **UserServlet**, **RoleServlet**.
- ✓ Sử dụng **Sitemesh** để tạo layout dùng chung cho cả 3 trang.



CYBERSOFT

ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

Mục tiêu

ADMIN PAGE

DASHBOARD

Quản lý thành viên

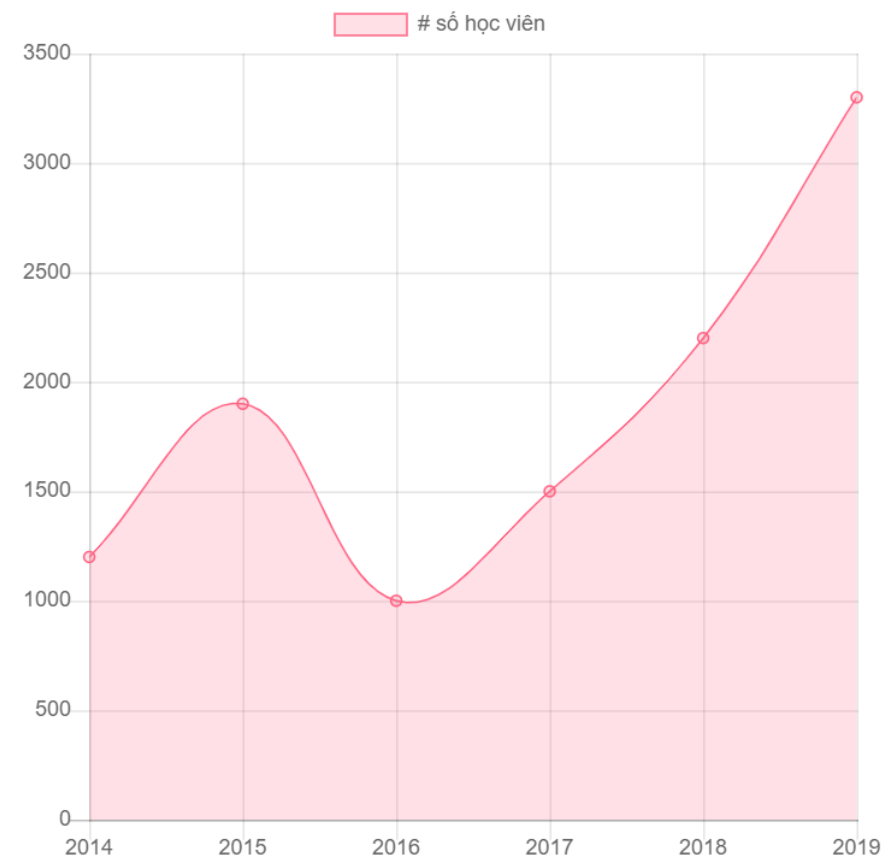
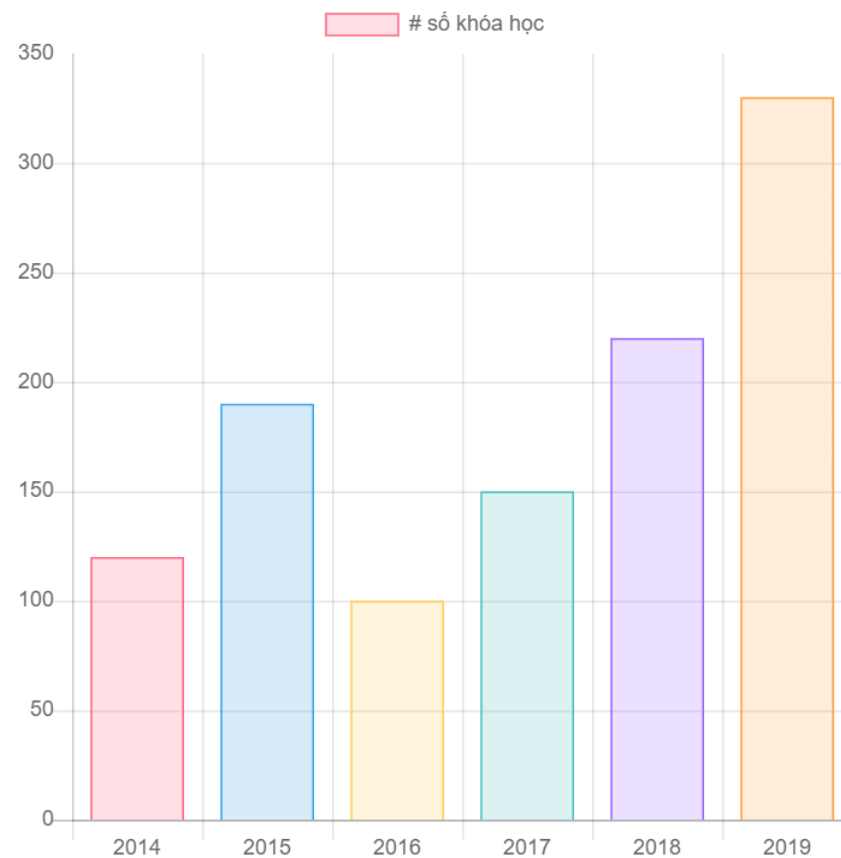
Quản lý quyền

Cấu hình hệ thống

Thông tin khác



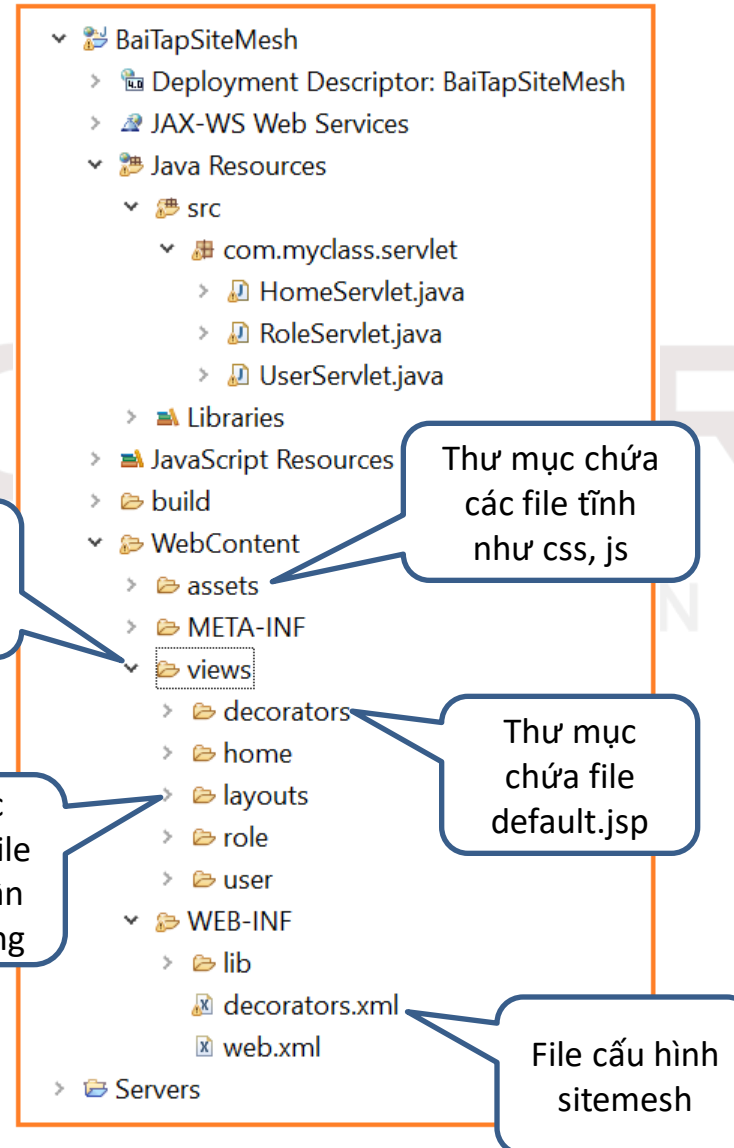
Cybersoft ▾



❑ Các bước thực hiện

- ✓ Bước 1: Tạo dự án.
- ✓ Bước 2: Tạo Servlet(HomeServlet, RoleServlet, UserServlet).
- ✓ Bước 3: Tạo các trang jsp cho dự án.
- ✓ Bước 4: Chia layout (Các file đặt trong thư mục views/layouts).
- ✓ Bước 5: Tạo file default.jsp (đặt trong thư mục decorators).
- ❖ Cấu hình Sitemesh.
 - ✓ Bước 6: Tạo file decorators.xml (đặt trong thư mục WEB-INF).
 - ✓ Bước 7: Thêm filter sitemesh vào file web.xml

Cấu trúc dự án



Servlet

❑ HomeServlet

```
@WebServlet(name = "HomeServlet", urlPatterns = "/home")
public class HomeServlet extends HttpServlet{

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {

        req.getRequestDispatcher("/views/home/index.jsp").forward(req, resp);
    }
}
```

❑ RoleServlet

```
@WebServlet(name = "RoleServlet", urlPatterns = "/role")
public class RoleServlet extends HttpServlet{

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {

        req.getRequestDispatcher("/views/role/index.jsp").forward(req, resp);
    }
}
```

header.jsp - footer.jsp

header.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<link rel="stylesheet"
    href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">
<!-- Bootstrap CSS -->
<link rel="stylesheet" href='<c:url value="/assets/css/bootstrap.min.css" />'>
<link rel="stylesheet" href='<c:url value="/assets/css/style.css" />'>
```

footer.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<script src="<c:url value="/assets/js/jquery.slim.min.js" />"></script>
<script src="<c:url value="/assets/js/popper.min.js" />"></script>
<script src="<c:url value="/assets/js/bootstrap.min.js" />"></script>
```

Navbar.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<nav class="navbar navbar-expand-sm navbar-light bg-light w-100">
  <a class="navbar-brand" href="#"><i class="fa fa-align-justify"></i></a>
  <button class="navbar-toggler d-lg-none" type="button"
    data-toggle="collapse" data-target="#collapsibleNavId"
    aria-controls="collapsibleNavId" aria-expanded="false"
    aria-label="Toggle navigation"></button>
  <div class="collapse navbar-collapse" id="collapsibleNavId">
    <ul class="navbar-nav ml-auto mt-2 mt-lg-0">
      <li class="nav-item dropdown"><a
        class="nav-link dropdown-toggle" href="#" id="dropdownId"
        data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
        Cybersoft </a>
        <div class="dropdown-menu dropdown-menu-right"
          aria-labelledby="dropdownId">
          <a class="dropdown-item" href="#">Thông tin cá nhân</a> <a
            class="dropdown-item" href="#">Cài đặt</a> <a
            class="dropdown-item" href="#">Thoát</a>
        </div></li>
    </ul>
  </div>
</nav>
```

Sidebar.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<div id="side-bar">
    <div class="logo">ADMIN PAGE</div>
    <ul class="list-group rounded-0">
        <li class="dashboard">DASHBOARD</li>
        <li><a href="user-list.html"> <i class="fa fa-user mr-2"></i>
            Quản lý thành viên
        </a></li>
        <li><a href="role-list.html"> <i class="fa fa-book mr-2"></i>
            Quản lý quyền
        </a></li>
        <li><a href="#"> <i class="fa fa-cogs mr-2"></i> Cấu hình hệ
            thống
        </a></li>
        <li><a href="#"> <i class="fa fa-slack mr-2"></i> Thông tin
            khác
        </a></li>
    </ul>
</div>
```

default.jsp

Khai báo thư viện thẻ của SiteMesh để sử dụng.

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib uri="http://www.opensymphony.com/sitemesh/decorator" prefix="dec" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>
        <dec:title />
    </title>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <jsp:include page="/views/layouts/header.jsp" />
    <dec:head />
</head>
<body>
    <div class="d-flex justify-content-between">
        <!-- SIDE BAR -->
        <jsp:include page="/views/layouts/sidebar.jsp" />

        <div id="admin-wrapper">
            <!-- NAVBAR -->
            <jsp:include page="/views/layouts/navbar.jsp" />

            <!-- CONTENT -->
            <section id="admin-content" class="p-3">
                <dec:body />
            </section>
        </div>
    </div>
    <jsp:include page="/views/layouts/footer.jsp" />
    <dec:getProperty property="page.scripts" />
</body>
</html>
```

Tiêu đề của mỗi trang sẽ được thay thế vào đây khi trang được gọi

Include các file chứa thành phần dùng chung

Nội dung của mỗi trang sẽ được thay thế vào đây khi trang được gọi

File home/index.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<head>
    <title>Trang chủ</title>
</head>

<div class="row">
    <div class="col-md-6">
        <canvas id="myChart" width="400" height="400"></canvas>
    </div>
    <div class="col-md-6">
        <canvas id="myChart2" width="400" height="400"></canvas>
    </div>
    <div class="col-md-1"></div>
</div>

<content tag="scripts" >
    <script src='<c:url value="/assets/js/chart.min.js" />'></script>
    <script src='<c:url value="/assets/js/custom-chart.js" />'></script>
</content>
```


decorator.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<decorators defaultdir="/views/decorators">
  <decorator name="default" page="default.jsp">
    <pattern>/*</pattern>
  </decorator>
</decorators>
```

Những url bắt đầu bằng / sẽ sử dụng file default.jsp làm layout dùng chung.

Khai báo đường dẫn thư mục chứa file layout dùng chung.

Khai báo file layout dùng chung.

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xmlns-compat" xsi:schemaLocation="http://xmlns.jcp.org/xmlns-compat http://xmlns.jcp.org/xmlns-compat.xsd">
  <display-name>BaiTapSiteMesh</display-name>
  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>

  <filter>
    <filter-name>sitemesh</filter-name>
    <filter-class>com.opensymphony.sitemesh.webapp.SiteMeshFilter</filter-class>
  </filter>

  <filter-mapping>
    <filter-name>sitemesh</filter-name>
    <url-pattern>*</url-pattern>
  </filter-mapping>
</web-app>
```

Mọi url bắt đầu bằng / đều phải đi qua SiteMeshFilter

Đây là một Filter, mọi request đều phải đi qua cái này

Dự án CRM

- Phân tích dự án
- Tạo dự án CRM
- Dàn layout bằng Servlet – JSP
- Xây dựng giao diện trang quản lý thành viên
- Xây dựng giao diện trang quản lý nhóm việc
- Xây dựng giao diện trang quản lý công việc
- Xây dựng giao diện trang quản lý thông tin cá nhân

DỰ ÁN CRM



C-R-M

6

5

7

Dashboard

Thành viên

Nhóm việc

THÀNH VIÊN

+ Thêm mới

Danh sách thành viên

10 records per page

Search:

Họ tên	Email	Địa chỉ	Điện thoại	Action
Cybersoft	cybersoft@gmail.com	459 - LẦU 2- SƯ VẠN HẠNH, QUẬN 10, HCM	0961051014	<div>✓✎🗑</div>
Lê Sỹ Mạnh	manh_dep_zai@gmail.com	02 - Phường Khắc Khoan - Đa Kao - Q.1 - tp.HCM	0366058489	<div>✓✎🗑</div>
Nguyễn Thái Hòa	nguyenthaihoa@gmail.com	39 - Cộng Hòa - Tân Bình - Tp.HCM	0356420241	<div>✓✎🗑</div>
Nguyễn Văn A	nguyenvana@gmail.com	Nguyễn Thái Sơn - Gò Vấp - Hồ Chí Minh	0123456788	<div>✓✎🗑</div>
Nguyễn Văn Tèo 123	teo@gmail.com	WAY ST 55	0123456788	<div>✓✎🗑</div>
Trần Văn Tâm	vantam2019@gmail.com	Quảng Phát - Quảng Tiến - Trảng Bom - Đồng Nai	0366127899	<div>✓✎🗑</div>

Hiện thị 1 đến 6 của 6 bản ghi

← Prev

1

Next →

DỰ ÁN CRM



C-R-M

6

5

7

Cybersoft

Dashboard

Thành viên

Nhóm việc

Nhóm việc 001

Thêm đầu việc

Thống kê công việc

Danh sách đầu việc

Trần Văn Tâm

Chưa thực hiện

Đầu việc 006

21/02/2019

Trần Văn Tâm

Đầu việc 007

21/02/2019

Trần Văn Tâm

Đầu việc 008

20/02/2019

Trần Văn Tâm

Đang thực hiện

kkkak

27/02/2019

Trần Văn Tâm

Đầu việc 005

19/02/2019

Trần Văn Tâm

Đầu việc 010

20/02/2019

Trần Văn Tâm

Hoàn thành

Đầu việc 004

23/02/2019

Trần Văn Tâm

2019 © Cybersoft CRM.

Tổng kết

☐ Nội dung đã học

- ✓ Tổng quan về JSP.
- ✓ Vòng đời và cách hoạt động của JSP.
- ✓ Sử dụng dụng EL trong JSP.
- ✓ Tích hợp và sử dụng thư viện JSTL.
- ✓ Thực hành chia layout bằng thư viện Sitemesh.