

**I. MỤC TIÊU:**

- Xây dựng ứng dụng web theo mô hình MVC sử dụng thư viện JSTL.

**II. TÓM TẮT LÝ THUYẾT:****1. Thư viện JSTL:**

- JSP Standard Tag Library (JSTL) là thư viện thẻ chuẩn cung cấp các thẻ để kiểm soát hành vi trang, lặp đi lặp lại và các lệnh điều khiển, các thẻ quốc tế hóa, và các thẻ SQL.

**a. Tổng quan các chức năng trong JSTL:**

Chức năng	Mô tả / Khai báo
Các thẻ cơ bản (Core Tags)	Các thẻ cơ bản cung cấp hỗ trợ cho bộ lặp (iteration), các điều kiện logic, bắt ngoại lệ, url, chuyển tiếp (forward) hoặc chuyển hướng (redirect),.... Để sử dụng JSTL core tags, bạn cần đưa nó vào trang JSP giống dưới đây:
	<ul style="list-style-type: none"><li>• <code>&lt;%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %&gt;</code></li></ul>
Các thẻ định dạng và địa phương hóa. (Formatting and Localization Tags)	Những thẻ này cung cấp định dạng cho các con số, ngày tháng và hỗ trợ i18n thông qua miền địa phương (locales) và bó tài nguyên (resource bundles). Bạn cần đưa nó vào trang JSP với cú pháp dưới đây.
	<ul style="list-style-type: none"><li>• <code>&lt;%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %&gt;</code></li></ul>
Các thẻ SQL (SQL Tags)	Các thẻ <b>JSTL SQL</b> cung cấp các hỗ trợ cho việc tương tác với cơ sở dữ liệu quan hệ như <b>Oracle</b> , <b>MySQL</b> ..v. Sử dụng thẻ SQL mà bạn có thể chạy các truy vấn cơ sở dữ liệu, bạn đưa nó vào trong JSP với cú pháp dưới đây:
	<ul style="list-style-type: none"><li>• <code>&lt;%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql" %&gt;</code></li></ul>
Các thẻ XML (XML Tags)	Thẻ XML được sử dụng để làm việc với các tài liệu XML như phân tích cú pháp XML, chuyển đổi dữ liệu <b>XML</b> và <b>XPath</b> đánh giá biểu thức. Cú pháp để có thể sử dụng các thẻ XML trong JSP trang là:
	<ul style="list-style-type: none"><li>• <code>&lt;%@ taglib uri="http://java.sun.com/jsp/jstl/xml" prefix="x" %&gt;</code></li></ul>
Các thẻ hàm JSTL (JSTL Functions Tags)	Thẻ JSTL cung cấp một số chức năng mà chúng ta có thể sử dụng để thực hiện các toán tử thông dùng, hầu hết trong số đó là cho String như nối chuỗi, phân tách chuỗi, ...Cú pháp:
	<ul style="list-style-type: none"><li>• <code>&lt;%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %&gt;</code></li></ul>

**b. Các thẻ JSTL cơ bản (JSTL core tags)**

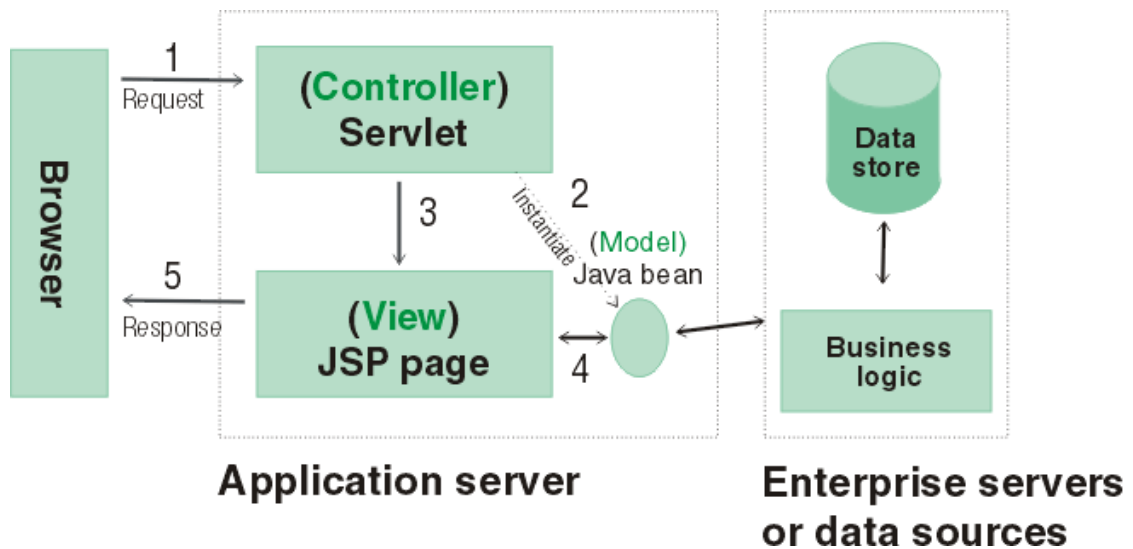
Thẻ	Mô tả
<c:out>	Để viết một cái gì đó trong trang <b>JSP</b> , bạn có thể sử dụng <b>EL</b> cũng có thể này
<c:import>	Giống với <jsp:include> hoặc chỉ thị include (include directive)
<c:redirect>	Chuyển hướng (redirect) yêu cầu tới một nguồn dữ liệu khác.
<c:set>	Sét đặt giá trị biến cho bởi phạm vi.
<c:remove>	Loại bỏ biến ra khỏi phạm vi đã cho.
<c:catch>	Bắt ngoại lệ và gói vào một đối tượng.
<c:if>	Điều kiện logic đơn giản, sử dụng với <b>EL</b> và bạn có thể sử dụng nó để xử lý các trường hợp ngoại lệ từ <c:catch>
<c:choose>	Tag có điều kiện đơn giản mà thiết lập một bối cảnh cho các trường hợp loại trừ lẫn nhau có điều kiện, đánh dấu bằng <c:when> và <c:otherwise>
<c:when>	Thẻ con của <c:choose>, khi một điều kiện tại when là đúng.
<c:otherwise>	Thẻ con của <c:choose>, khi tất cả các điều kiện <c:when> là sai.
<c:forEach>	Dùng để lặp trên một tập hợp.
<c:forTokens>	dùng để lặp trên chuỗi (tokens) được phân cách bởi một dấu phân cách.
<c:param>	Được sử dụng với <c:import> để truyền các tham số.
<c:url>	Để tạo một URL với các tham số tùy chọn (optional query string parameters).

**c. Các toán tử EL:**

Bằng (equals)	==	eq
Không bằng (Not equals)	!=	ne
Nhỏ hơn (Less than)	<	lt
Lớn hơn (Greater than)	>	gt
Nhỏ hơn hoặc bằng (Less than or equals)	<=	le
Lớn hơn hoặc bằng (Greater than or equals)	>=	ge

**2. Mô hình MVC:**

- MVC chia ứng dụng phần mềm ra thành 3 phần có tương tác với nhau là **Model** (dữ liệu), **View** (giao diện), **Controller** (tương tác với Model và View, giúp 2 phần này giao tiếp với nhau)



Trong mô hình ứng dụng web thì:

- **Model** sẽ là JavaBean, Enterprise JavaBean hay Web Service.
- Các trang html, servlet, jsp,... là những thành phần đại diện cho **View**.
- Cuối cùng, servlet đóng vai trò là **Controller**.

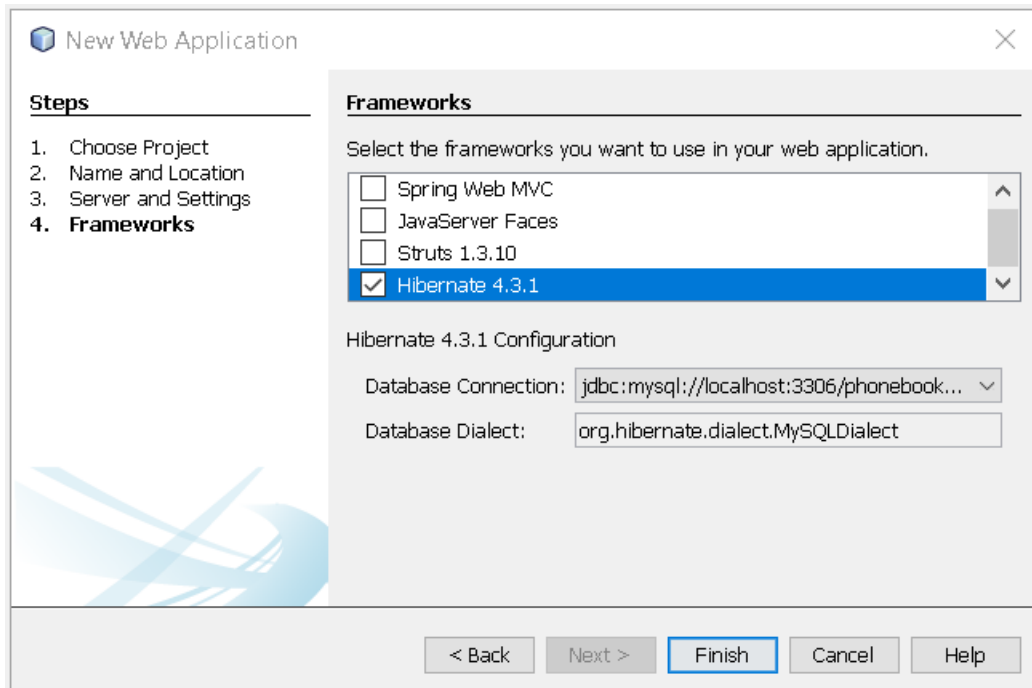
**III. NỘI DUNG THỰC HÀNH:**

Trong Lab này sinh viên sẽ được hướng dẫn các bước xây dựng ứng dụng web theo mô hình MVC. Ứng dụng quản lý danh bạ điện thoại.

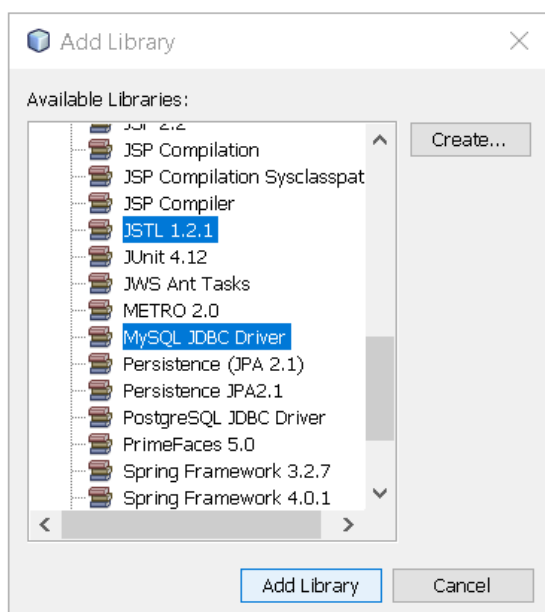
**Lab7\_1: Cấu hình ứng dụng web dùng hibernate.**

Sinh viên tạo database: phonebook sau đó import file phonebook.sql.

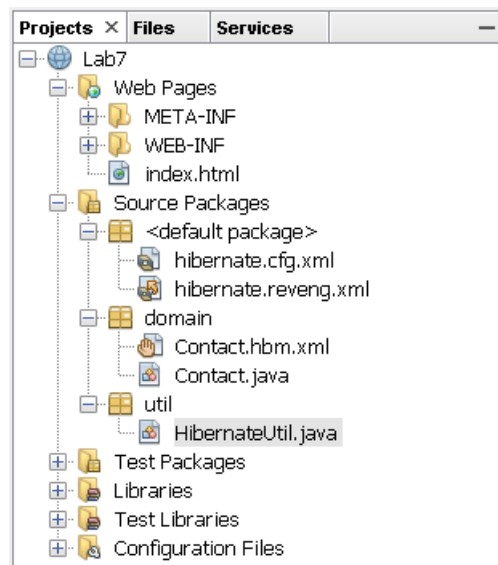
- Tạo project web, đặt tên Lab7, ở Step 4, chọn Framework Hibernate 4.3.1



- Thêm thư viện JSTL và Mysql driver



- Tạo các đối tượng từ database (xem lại Lab5)



- Tạo các dịch vụ CRUD cho đối tượng Contact.

```
public class ContactService {
    private Session ses;
    public ContactService() {
        ses = HibernateUtil.openSession();
    }
    public List<Contact> getContacts() {
        Transaction tx = ses.beginTransaction();
        List<Contact> ds = ses.createQuery("from Contact").list();
        tx.commit();
        return ds;
    }
    public Contact get(int id) {
        Transaction tx = ses.beginTransaction();
        Contact c = (Contact) ses.get(Contact.class, id);
        tx.commit();
        return c;
    }
    public int create(Contact c) {
        Transaction tx = ses.beginTransaction();
        ses.save(c);
        tx.commit();
        return c.getId();
    }
    public void save(Contact c) {
        Transaction tx = ses.beginTransaction();
        ses.update(c);
        tx.commit();
    }
    public void delete(Contact c) {
        Transaction tx = ses.beginTransaction();
        ses.delete(c);
        tx.commit();
    }
}
```

### Lab7\_2: Hiển thị danh sách contact

Trong bài lab này, chúng ta sẽ thực hiện các trang:

- Trang **servlet ListContacts.java**: có vai trò là model, đóng gói dữ liệu chuyển cho trang ShowContact.jsp.
- Trang **ShowContacts.jsp**: có vai trò là view, hiển thị dữ liệu từ model chuyển qua.
- Trang **servlet Controller.java**: có vai trò là controller, điều hướng view và model tùy theo yêu cầu.

#### Servlet ListContacts.java:

- Tạo mới servlet, đặt tên **ListContacts**, tích vào checkbox, nhớ tên URL Pattern(s)

**New Servlet**

**Steps**

1. Choose File Type
2. Name and Location
3. **Configure Servlet Deployment**

**Configure Servlet Deployment**

Register the Servlet with the application by giving the Servlet an internal name (Servlet Name). Then specify patterns that identify the URLs that invoke the Servlet. Separate multiple patterns with commas.

☒ Add information to deployment descriptor (web.xml)

Class Name:

Servlet Name:

URL Pattern(s):

Initialization Parameters:

Name	Value
------	-------

New Edit... Delete

< Back Next > **Finish** Cancel Help

- Code xử lý cho trang servlet **ListContacts**

```
package servlet;
import ...10 lines
public class ListContacts extends HttpServlet {
    private ContactService cs;

    @Override
    public void init() throws ServletException {
        super.init();
        cs = new ContactService();
    }
    protected void processRequest(
        HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        String url = "/WEB-INF/view/ShowContacts.jsp";
        List<Contact> contacts = cs.getContacts();
        request.setAttribute("contacts", contacts);
        RequestDispatcher rd = request.getRequestDispatcher(url);
        rd.forward(request, response);
    }

    HttpServlet methods. Click on the + sign on the left to edit the code.
}
```

**Trang JSP ShowContacts.jsp**

- Tạo thư mục trong WEB-INF đặt tên là view.
- Tạo trang JSP trong thư mục view đặt tên là ShowContacts.jsp.
- Code xử lý cho trang **ShowContacts.jsp**

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type"
              content="text/html; charset=UTF-8">
        <title>Show contacts</title>
    </head>
    <body>
        <h2>PHONE BOOK</h2>
        <table border="1" cellspacing="0">
            <tr>
                <th>id</th>
                <th>name</th>
                <th>address</th>
                <th>phone</th>
                <th>email</th>
            </tr>
            <c:forEach var="contact" items="${contacts}">
                <tr>
                    <td>${contact.id}</td>
                    <td>${contact.name}</td>
                    <td>${contact.address}</td>
                    <td>${contact.phone}</td>
                    <td>${contact.email}</td>
                </tr>
            </c:forEach>
        </table>
    </body>
</html>
```



### Trang servlet Controller:

- Tạo mới trang servlet đặt tên là **Controller**
- Code xử lý cho trang servlet **Controller**:

```
package servlet;
import ...6 lines
public class Controller extends HttpServlet {
    protected void processRequest(
        HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        String act = request.getParameter("act");
        String url = "";
        switch(act) {
            case "view":
                url = "/ListContacts";
                break;
        }
        RequestDispatcher rd = request.getRequestDispatcher(url);
        rd.forward(request, response);
    }
}
```

HttpServlet methods. Click on the + sign on the left to edit the code.

- Chỉnh sửa lại trang **index.html** như sau:

```
<html>
<head>
    <title>Lab 7</title>
</head>
<body>
    <h2>Lab 7</h2>
    <a href="./Controller?act=view">View contacts</a><p>
    <a href="./Controller?act=add">Add contacts</a>
</body>
</html>
```

- Chạy dự án, kích vào link View contacts.

**Lab 7**[View contacts](#)[Add contacts](#)

- Kết quả như hình:

**PHONE BOOK**

id	name	address	phone	email
1	Kha Ho	Tan Binh	09097777777	kho.ho@hotmail.com
2	Duc Le	Quan 10	01454554455	ducle@yahoo.com
3	Bach Ngo	Quan 7	78422122122	bach.ngo@gmail.com

[Lab 7 Home](#)

**Lab7\_3: Thêm một contact mới**

Trong bài lab này, chúng ta sẽ thực hiện các trang:

- Trang **AddContactForm.jsp**: hiển thị form nhập thông tin contact (view).
- Chỉnh sửa **servlet Controller.java** (controller)
- Trang **servlet AddContact.java**: thêm contact mới (model).

**Trang AddContactForm.jsp:**

- Tạo trang JSP trong thư mục view đặt tên là **AddContactForm.jsp**.
- Code xử lý cho trang **AddContactForm.jsp**

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<html>
  <head>
    <meta http-equiv="Content-Type"
      content="text/html; charset=UTF-8">
    <title>Add a contact</title>
  </head>
  <body>
    <h2>Add a contact</h2>
    <form method="post" action="./AddContact">
      <table border="0" cellspacing="0">
        <tr>
          <td>Name:</td><td><input name="name"></td>
        </tr>
        <tr>
          <td>Address:</td><td><input name="address"></td>
        </tr>
        <tr>
          <td>Phone:</td><td><input name="phone"></td>
        </tr>
        <tr>
          <td>Email:</td><td><input name="email"></td>
        </tr>
        <tr>
          <td colspan="2"><input type="submit" value=" Add ">
          <h3><a href="./">Lab 7 Home</a></h3></td>
        </tr>
      </table>
      <div>${mess == null ? "" : mess}</div>
    </form>
  </body>
</html>
```

## LAB 7: XÂY DỰNG ỨNG DỤNG WEB THEO MÔ HÌNH MVC

- Chỉnh sửa code trong trang **servlet Controller.java** như sau:

```
package servlet;
import ...6 lines
public class Controller extends HttpServlet {
    protected void processRequest(
        HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        String act = request.getParameter("act");
        String url = "";
        switch(act) {
            case "view":
                url = "/ListContacts";
                break;
            case "add":
                url = "/WEB-INF/view/AddContactForm.jsp";
                break;
        }
        RequestDispatcher rd = request.getRequestDispatcher(url);
        rd.forward(request, response);
    }
}
```

HttpServletRequest methods. Click on the + sign on the left to edit the code.

- Chạy ứng dụng, kích link [Add a contact](#)
- Kết quả như hình:

### Add a contact

Name:

Address:

Phone:

Email:

[Lab 7 Home](#)

### - Trang servlet AddContact.java

```
package servlet;
import ...8 lines
public class AddContact extends HttpServlet {
    private ContactService cs;
    @Override
    public void init() throws ServletException {
        super.init();
        cs = new ContactService();
    }
    protected void processRequest(
        HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        Contact c = new Contact();
        c.setName(request.getParameter("name"));
        c.setAddress(request.getParameter("address"));
        c.setPhone(request.getParameter("phone"));
        c.setEmail(request.getParameter("email"));
        int id = cs.create(c);
        String url = "/WEB-INF/view/AddContactForm.jsp";
        String mess;
        if (id>0)
            mess = "<font color='green'><h3>Success!</h3></font>";
        else
            mess = "<font color='red'><h3>Fail!</h3></font>";
        request.setAttribute("mess", mess);
        RequestDispatcher rd = request.getRequestDispatcher(url);
        rd.forward(request, response);
    }
}
HttpServlet methods. Click on the + sign on the left to edit the code.
```

### - Điền thông tin contac, kích nút **Add**, kết quả như hình:

**Add a contact**  
  
Name:   
Address:   
Phone:   
Email:   
  
  
[Lab 7 Home](#)  
  
**Success!**

**IV. BÀI TẬP LÀM THÊM:**

- 1) Viết thêm chức năng Xóa, Sửa Contact.
- 2) Thiết kế ứng dụng web theo mô hình MVC, cho phép chứng thực người dùng.