

Lab 02

Mô hình kiến trúc MVC trong phát triển ứng dụng web

Mục tiêu

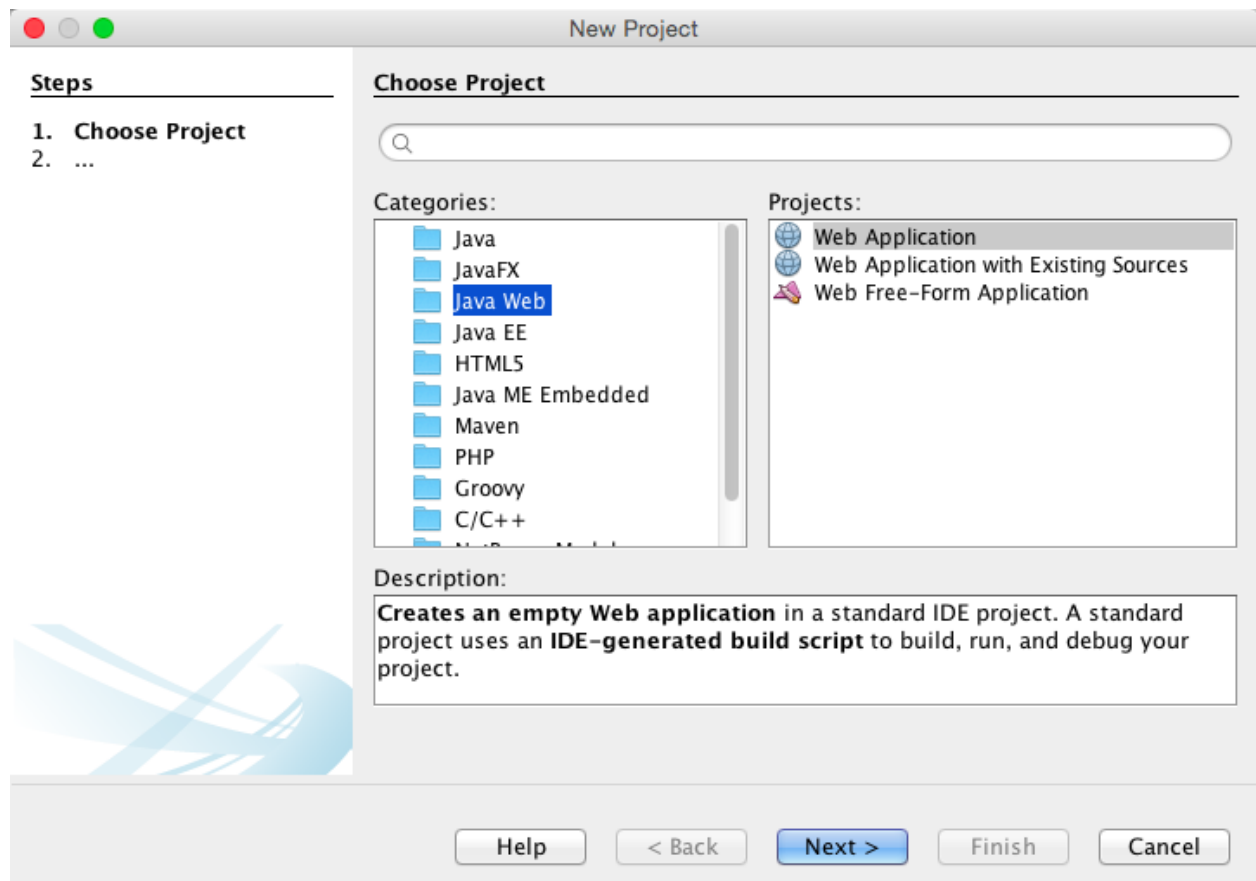
- Tạo ứng dụng web đơn giản dùng MVC design pattern
- Biết các thành phần nào trong web application tương ứng với thành phần Model, View, Control trong kiến trúc MVC

Phần I Bài tập step by step

Bài 1.1:

Tạo một web application đơn giản theo kiến trúc MVC. Ứng dụng này cho phép chúng ta tạo trang login đơn giản dùng JSP – Servlet.

Step 1: Open Netbeans -> Projects -> New Project -> Web Application



New Web Application

Steps

1. Choose Project
2. **Name and Location**
3. Server and Settings
4. Frameworks

Name and Location

Project Name:

Project Location:

Project Folder:

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

New Web Application

Steps

1. Choose Project
2. Name and Location
3. **Server and Settings**
4. Frameworks

Server and Settings

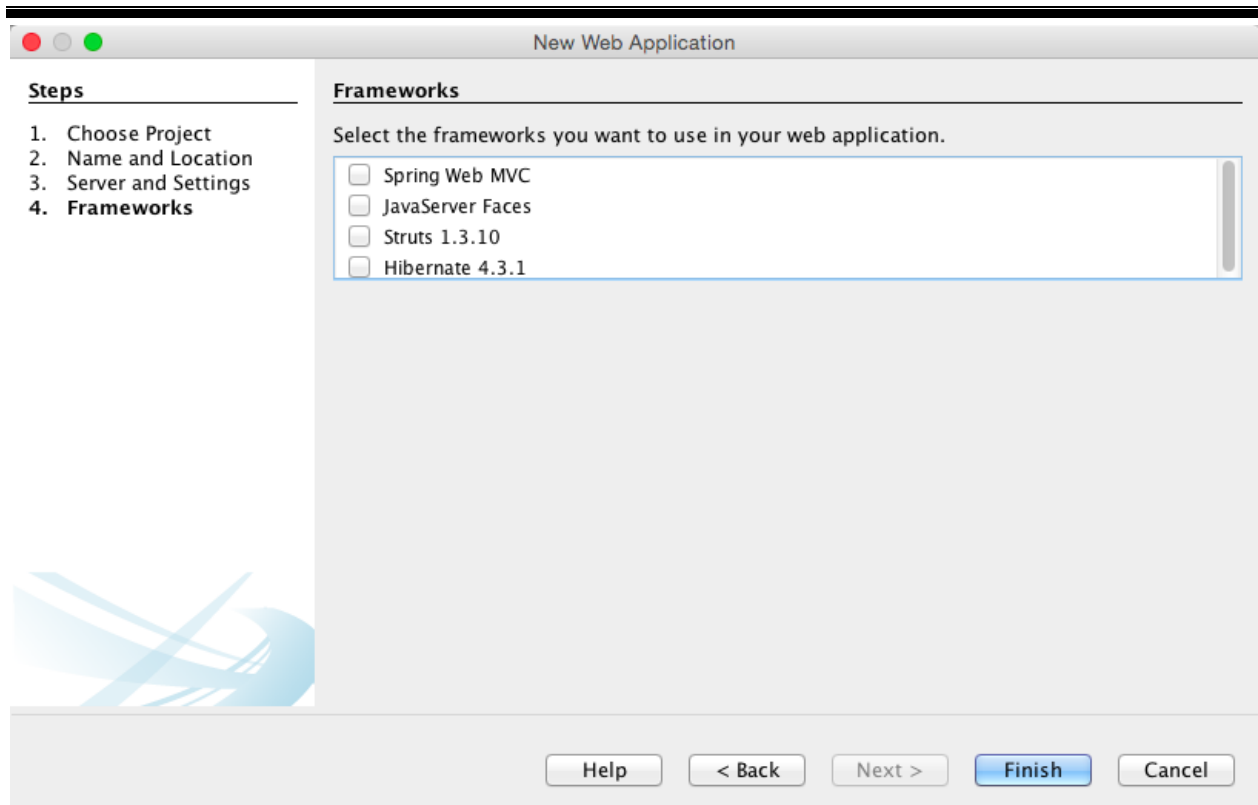
Add to Enterprise Application:

Server:

Java EE Version:

Note: Source Level 7 will be set for Java EE 7 project.

Context Path:



Step 2: Tạo file `UserBean.java` đại diện cho thành phần **Model**, biểu diễn các thông tin về user. Mỗi user có 2 thuộc tính là name và password.

Đặc trưng của thành phần Model là để lưu trữ thông tin về dữ liệu, do vậy mỗi thuộc tính name và password nói trên cần có các phương thức getter và setter tương ứng.

Class `UserBean` như sau:

```
package model;

public class UserBean {

    private String name, password;

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }
}
```

Tạo thêm phương thức `validate()` thực hiện verify nếu user nhập vào là admin hay không:

```
public boolean validate() {
    if (password.equals("admin")) {
        return true;
    } else {
        return false;
    }
}
```

Step 3: Tạo file `LoginServlet.java` đại diện cho **Controller**, thực hiện kiểm soát việc login của user.

New Servlet

Steps

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

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

⚠ Warning: It is highly recommended that you do not place Java classes in the default package.

Help < Back **Next >** Finish Cancel

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

Help < Back Next > Finish Cancel

Khi chọn Add information to deployment descriptor (web.xml), các thông tin về servlet và servlet mapping sẽ được tự động thêm vào file config web.xml.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<web-app version="3.1"
```

```
    xmlns="http://xmlns.jcp.org/xml/ns/javaee"
```

```
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd">
```

```
    <servlet>
```

```
        <servlet-name>LoginServlet</servlet-name>
```

```
        <servlet-class>LoginServlet</servlet-class>
```

```
    </servlet>
```

```
    <servlet-mapping>
```

```
        <servlet-name>LoginServlet</servlet-name>
```

```
        <url-pattern>/LoginServlet</url-pattern>
```

```

</servlet-mapping>

<session-config>

    <session-timeout>

        30

    </session-timeout>

</session-config>

</web-app>

```

Code của file LoginServlet.java như sau:

```

protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html");
    PrintWriter out=response.getWriter();

    String name=request.getParameter("name");
    String password=request.getParameter("password");

    UserBean bean=new UserBean();
    bean.setName(name);
    bean.setPassword(password);
    request.setAttribute("bean",bean);

    boolean status=bean.validate();

    if(status){
        RequestDispatcher rd=request.getRequestDispatcher("login-success.jsp");
        rd.forward(request, response);
    }
    else{
        RequestDispatcher rd=request.getRequestDispatcher("login-error.jsp");
        rd.forward(request, response);
    }
}

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

```

Step 4: Tạo các file view: index.jsp để lấy thông tin từ người dùng, login-success.jsp, login-error.jsp. Các file này đóng vai trò thành phần View trong kiến trúc MVC

File index.jsp chứa form login

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

    <head>

        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">

        <title>Login</title>

    </head>

    <body>

        <h1>Enter information of user:</h1>

        <form action="LoginServlet" method="post">

            Name:<input type="text" name="name"><br>

            Password:<input type="password" name="password"><br>

            <input type="submit" value="login">

        </form>

    </body>

</html>
```

File login-success.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@page import="model.UserBean" %>

<!DOCTYPE html>

<html>

    <head>
```

```
<meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">

<title>Login success</title>

</head>

<body>

<p>You are successfully logged in!</p>

<%

    UserBean bean=(UserBean)request.getAttribute("bean");

    out.print("Welcome, "+bean.getName());

%>

</body>

</html>
```

File login-error.jsp:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

    <head>

        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">

        <title>Login failed</title>

    </head>

    <body>

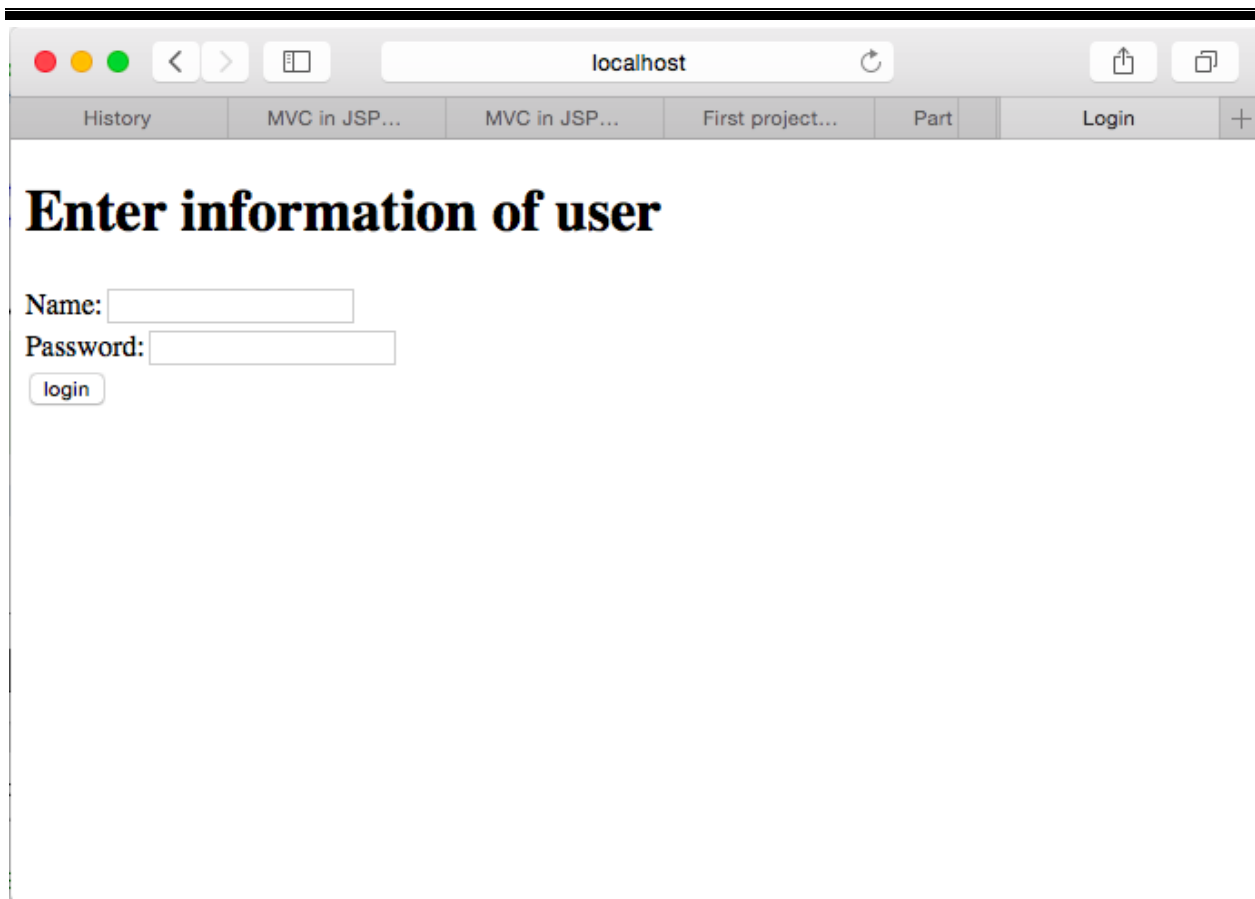
        <p>Sorry! username or password error</p>

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

    </body>

</html>
```

Step 5: Build và Run ứng dụng

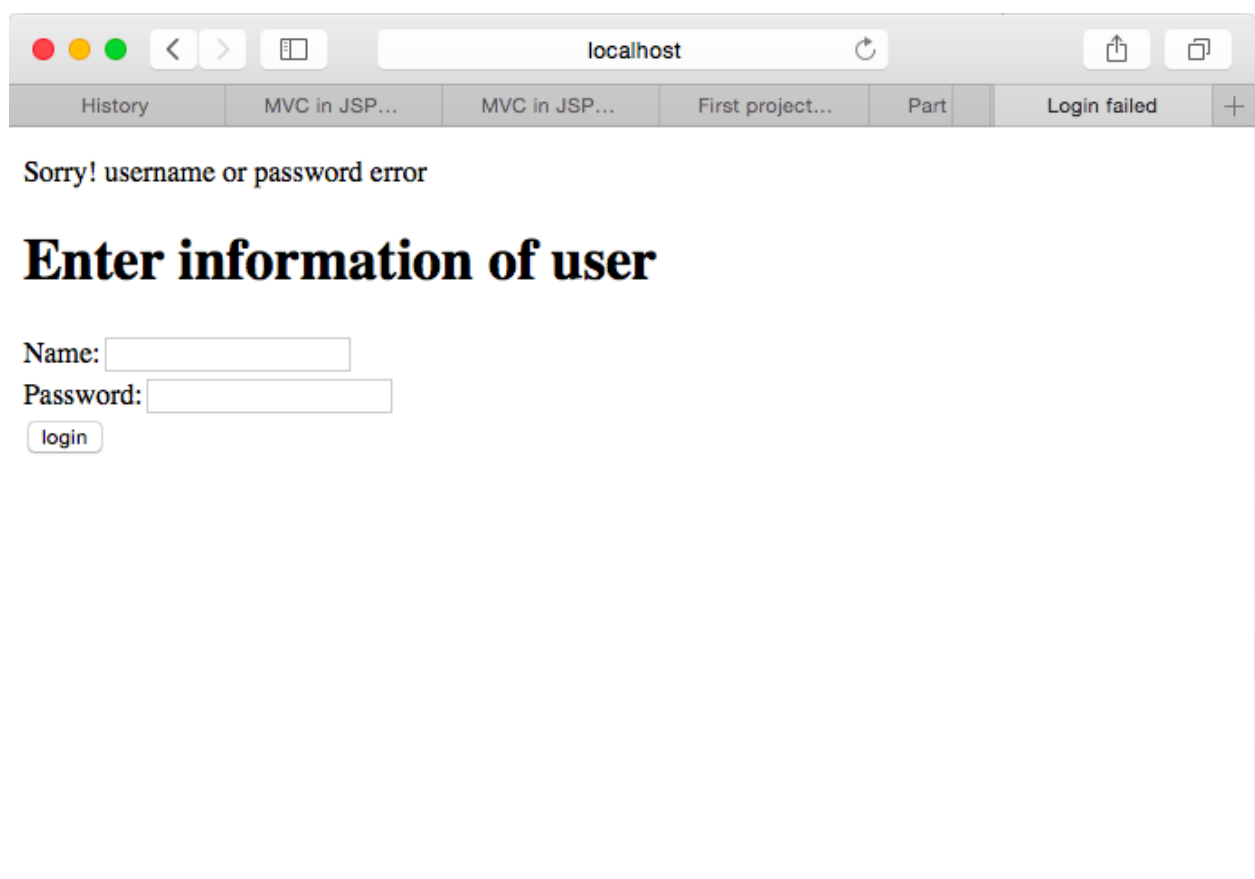


A screenshot of a web browser window. The address bar shows 'localhost'. The browser's history bar includes 'History', 'MVC in JSP...', 'MVC in JSP...', 'First project...', 'Part', and 'Login'. The main content area displays a login form with the title 'Enter information of user'. The form contains two input fields: 'Name:' and 'Password:'. Below the 'Password:' field is a 'login' button.

Enter information of user

Name:

Password:



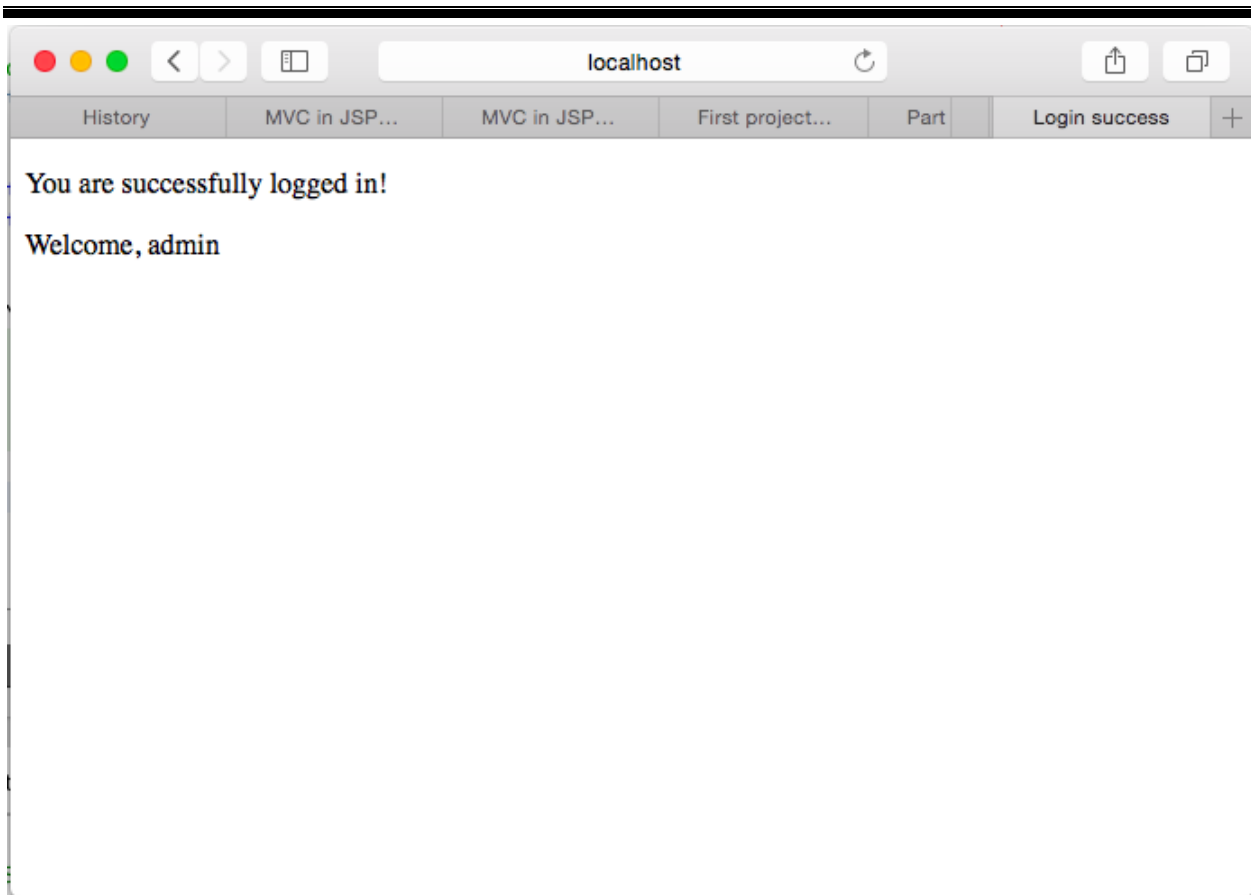
A screenshot of a web browser window. The address bar shows 'localhost'. The browser's history bar includes 'History', 'MVC in JSP...', 'MVC in JSP...', 'First project...', 'Part', and 'Login failed'. The main content area displays a login form with the title 'Enter information of user'. Above the form, there is an error message: 'Sorry! username or password error'. The form contains two input fields: 'Name:' and 'Password:'. Below the 'Password:' field is a 'login' button.

Sorry! username or password error

Enter information of user

Name:

Password:



Câu hỏi:

- Hãy tìm hiểu sự khác nhau giữa hai phương thức: `doGet` và `doPost`.
- Hãy tìm hiểu về sơ đồ sequence diagram của UML và vẽ lại quá trình gửi nhận và xử lý message giữa 3 thành phần model-view-control trong bài tập nói trên.

Phần II: Bài tập tự làm

Tạo một ứng dụng web application theo mô hình MVC có chức năng sau: 1 trang hiển thị form add profile của user (đơn giản first name + last name + username + pass), sau khi add thì redirect đến trang hiển thị: Welcome to new user, [Hiển thị tên đầy đủ của người dùng thông qua thuộc tính Firstname và Lastname]. Sau đó thông báo Username: [...], Password: [...]. Gợi ý: Tạo một user bean mới có 4 thông tin nói trên, 2 pages jsp cho view và 1 servlet.

HẾT