UNIVERSITI MALAYSIA TERENGGANU

CSM3023 WEB-BASED APPLICATION DEVELOPMENT (K1)

BACHELOR OF COMPUTER SCIENCE (MOBILE COMPUTING) WITH HONORS

LAB 5

SEMESTER II 2023/2024

Prepared for:

DR. MOHAMAD NOR HASSAN

Prepared by:

NUR FADHILAH BINTI MOHD RAHMAT
(S67172)

Week 5

JSP: JavaBeans & Java Standard Tag Library (JSTL)

Web Programming 2

Name:

Matric #:

Semester: Lab:

Demonstrator:

Lecturers

PUSAT PENGAJIAN INFORMATIK DAN MATEMATIK GUNAAN (PPIMG), UNIVERSITI MALAYSIA TERENGGANU (UMT)

Revision History

Revision Date	Previous Revision Date	Summary of Changes	Changes Marked
		First Issue	Mohamad Nor Hassan
		Second Issue	Dr Rabiei Mamat Dr Faizah Aplop Dr Fouad Ts Dr Rosmayati Mohemad Fakhrul Adli Mohd Zaki
21/02/2019		Addition of Revision History, Table of Contents, Formatting Cover Page	Fakhrul Adli Mohd Zaki

Table of Contents

Arahan:

Manual makmal ini adalah untuk kegunaan pelajar-pelajar Pusat Pengajian Informatik dan Matematik Gunaan (PPIMG), Universiti Malaysia Terengganu (UMT) sahaja. Tidak dibenarkan mencetak dan mengedar manual ini tanpa kebenaran rasmi daripada penulis.

Sila ikuti langkah demi langkah sebagaimana yang dinyatakan di dalam manual. Tandakan (√) setiap langkah yang telah selesai dibuat dan tulis kesimpulan bagi setiap aktiviti yang telah selesai dijalankan.

Instruction:

This laboratory manual is for use by the students of the School of Informatics and Applied Mathematics (PPIMG), Universiti Malaysia Terengganu (UMT) only. It is not permissible to print and distribute this manual without the official authorisation of the author.

Please follow step by step as described in the manual. Tick (√) each step completed and write the conclusions for each completed activity.

LINK GITHUB:
https://github.com/ilahsyugaa/Lab5_S67172.git

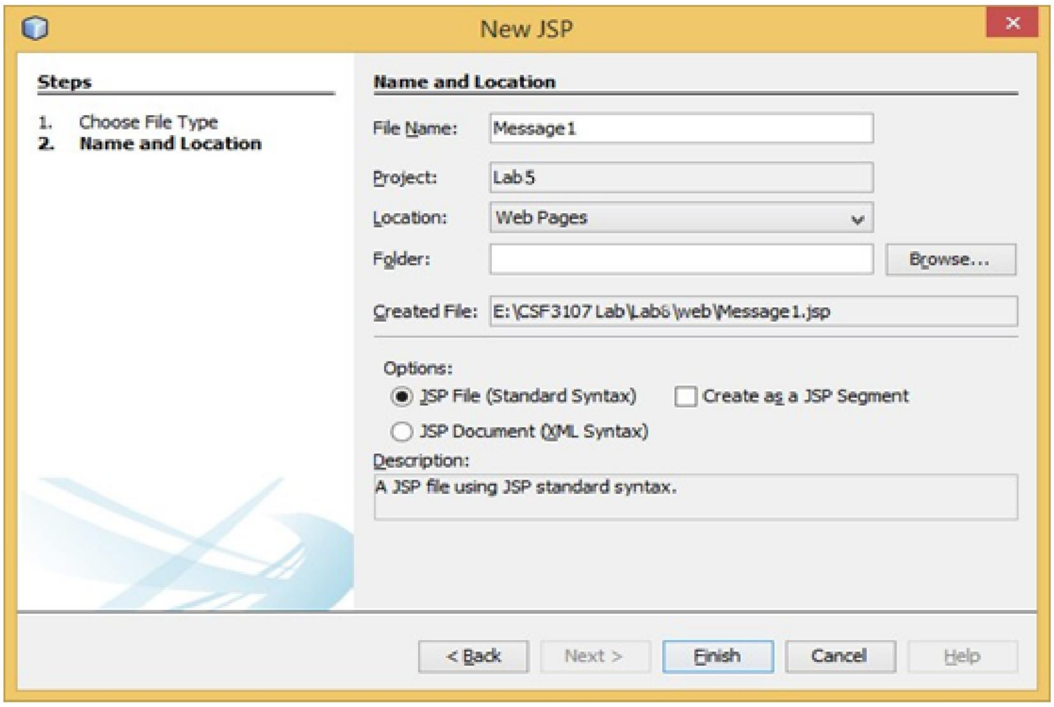
Task 1: Using Scriplet to Access a Simple JavaBeans

Objective: Use Java Scriptlet to access JavaBeans

Problem Description: Write a JavaBeans that can display message "Welcome to CSF3107 courses….!".

Estimated time: 20 minutes

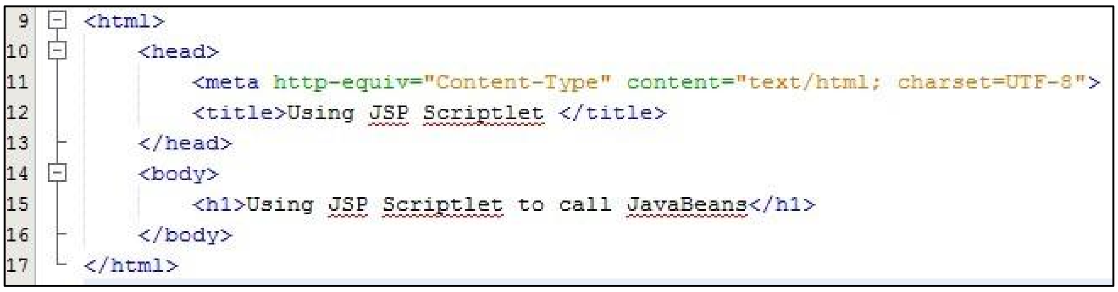
1. Go to *Lab5*'s project.
2. Create a new JSP's file.
3. Type file name as *Message1*.



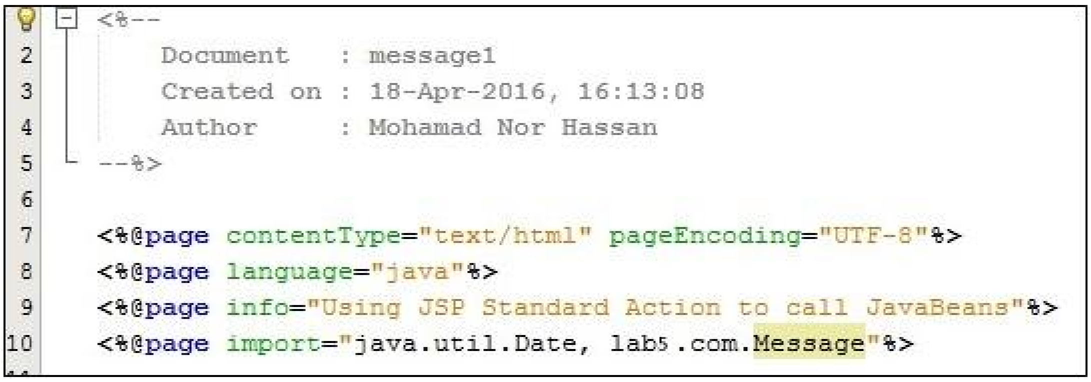
Name and Location

File Name: Message1
Project: Lab5
Location: Web Pages
Folder:
Created File: E:\CSF3107 Lab\Lab6\web\Message1.jsp
Options:
(●) JSP File (Standard Syntax) [ ] Create as a JSP Segment
( ) JSP Document (XML Syntax)
Description:
A JSP file using JSP standard syntax.

4. Rename title as *Using JSP Scriptlet*.
5. Rename <h1> as *Using JSP Standard Action to call JavaBeans.*

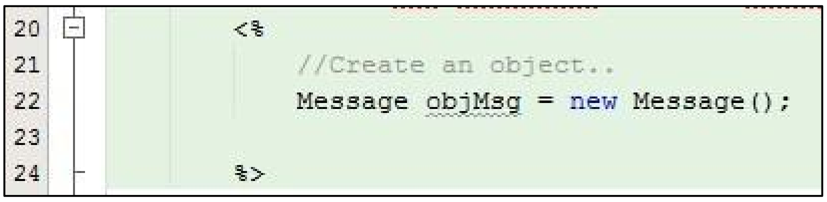

```
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Using JSP Scriptlet </title>
    </head>
    <body>
        <h1>Using JSP Scriptlet to call JavaBeans</h1>
    </body>
</html>
```

6. Use JSP page directive to include the information such as content type,

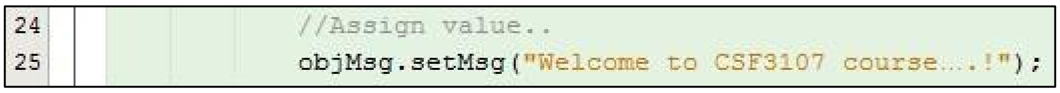

```
<%--
    Document   : message1
    Created on : 18-Apr-2016, 16:13:08
    Author     : Mohamad Nor Hassan
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page language="java"%>
<%@page info="Using JSP Standard Action to call JavaBeans"%>
<%@page import="java.util.Date, lab5.com.Message"%>
```

page info, language, lab8.com package and use java util API.

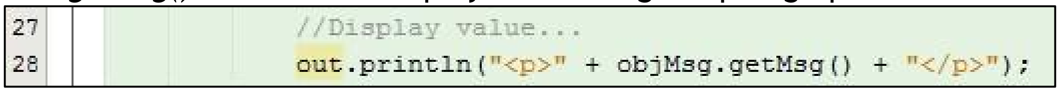
7. Use a Java scriptlet to create an object for *Message* class.


```
        <%
            //Create an object..
            Message objMsg = new Message();

        %>
```

8. Then, assign the value as *"Welcome to CSF3107 course….!* via setter method *setMsg(String msg)*.

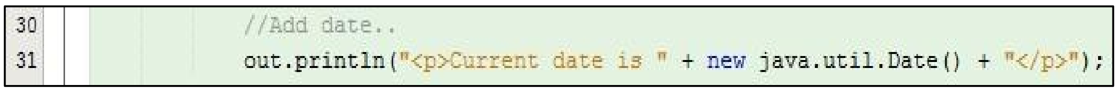

```
            //Assign value..
            objMsg.setMsg("Welcome to CSF3107 course...!");
```

9. Use *getMsg()*'s method to display the message at paragraph.



```
            //Display value...
            out.println("<p>" + objMsg.getMsg() + "</p>");
```

10. Add current date to the next paragraph.

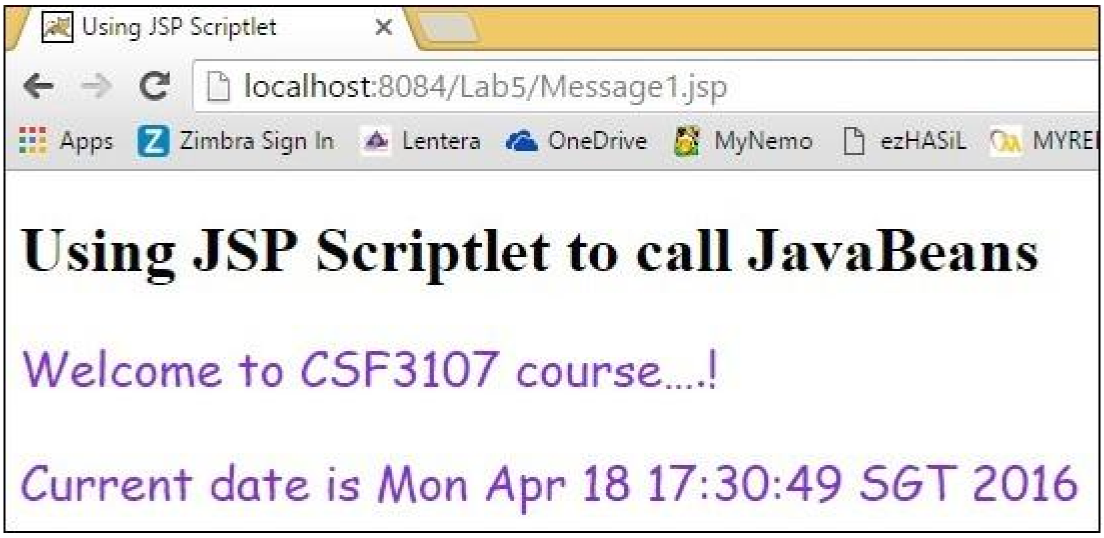


```
        //Add date..
        out.println("<p>Current date is " + new java.util.Date() + "</p>");
```

11. Save *Message1.jsp*

12. Compile and run *Message1.jsp*.

13. You should get the following output.



Using JSP Scriptlet to call JavaBeans

Welcome to CSF3107 course....!

Current date is Mon Apr 18 17:30:49 SGT 2016

Reflection

1. What you have learnt from this exercise?

-JavaBeans encapsulate data and provide getter and setter methods to access and modify this data, promoting data hiding and a clean interface.

2. Explain the differences when calling JavaBeans using JSP Standard Action and Java Scriptlet.

-**JSP Standard Actions:**

-Standard actions handle instantiation and property management declaratively, often without needing to explicitly import classes in the JSP.

-**Java Scriptlets:**

-Scriptlets require explicit instantiation and method calls, which can lead to verbose and cluttered JSP files. Additionally, necessary imports must be declared.

Task 2: Problem Solving using JavaBeans

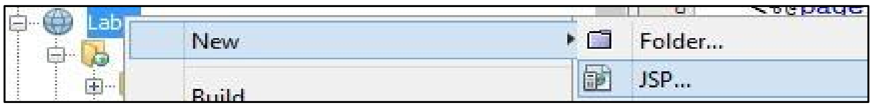
Objective: Use JavaBeans in JavaScriptlet

Problem Description: Create sample web form to register IT's training based on the these fees;
C++ training (values as "1")= RM3000 per pax
Java for beginner (values as "2")= RM3000 per pax
HTML5 (values as "3")= RM2800 per pax
Java EEE (values as "4")= RM5500 per pax
Android Programming (values as "5")= RM3200 per pax
Student = Yes (values as "1") and No (Values as "0")
Student will entitle 10% discount
You must cater all front-end validation.

Estimated time: 60 minutes

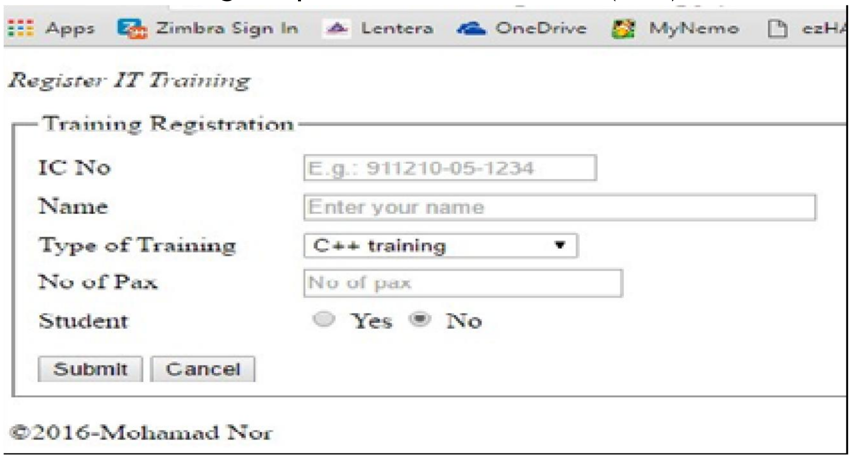
1. Choose Project *Lab5*.

2. Create a new JSP's file.



3. Type file name as *registerTraining*.

4. Prepare the following Graphical User Interface (GUI).



Register IT Training

Training Registration

IC No	E.g.: 911210-05-1234
Name	Enter your name
Type of Training	C++ training
No of Pax	No of pax
Student	Yes [✓ No]

Submit Cancel

©2016-Mohamad Nor

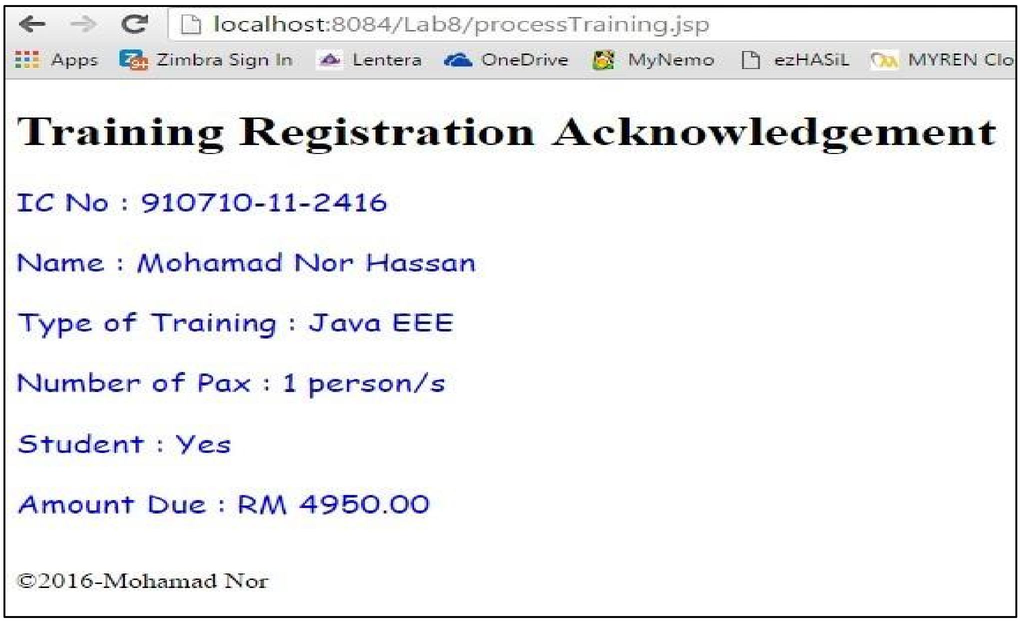
5. Create a *Register* JavaBeans to cater the business requirements and store it into package *lab8.com*.

6. Create a new file name known as *processTraining.jsp*.

7. Attached the *pocessTraining.jsp*'s form to *registerTraining.jsp*.

8. Open *pocessTraining.jsp*'s and use Java Scriptlet to invoke *Register* JavaBeans.

9. Output will appear in web browser.



```
Training Registration Acknowledgement

IC No : 910710-11-2416

Name : Mohamad Nor Hassan

Type of Training : Java EEE

Number of Pax : 1 person/s

Student : Yes

Amount Due : RM 4950.00

©2016-Mohamad Nor
```

Reflection

1. What you have learnt from this exercise?

-**Separation of Concerns:** Using JavaBeans in JSP helps in separating business logic from presentation logic, leading to cleaner and more maintainable code.

-

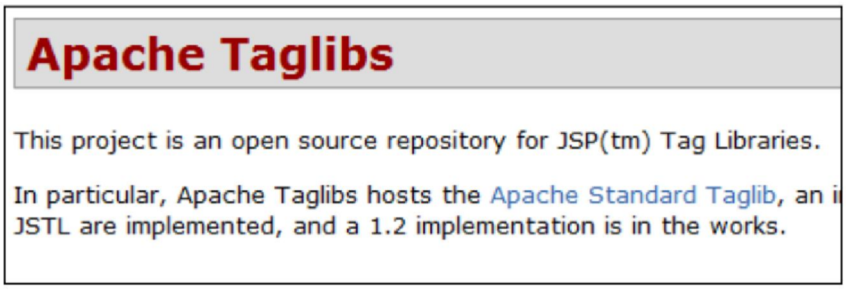
2. Describe the steps how you construct *Register* JavaBeans?

Step 1: Define the Package
Step 2: Declare the Class and Private Properties
Step 3: Create Public Getters and Setters
Step 4: Add Business Logic Methods

Task 3: Installing JSTL Taglibs

Objective:	Installation of Apache Taglibs
Problem Description:	To install JSTL Library
Estimated time:	25 minutes

1. Go to the browser and type URL http://tomcat.apache.org/taglibs/index.html

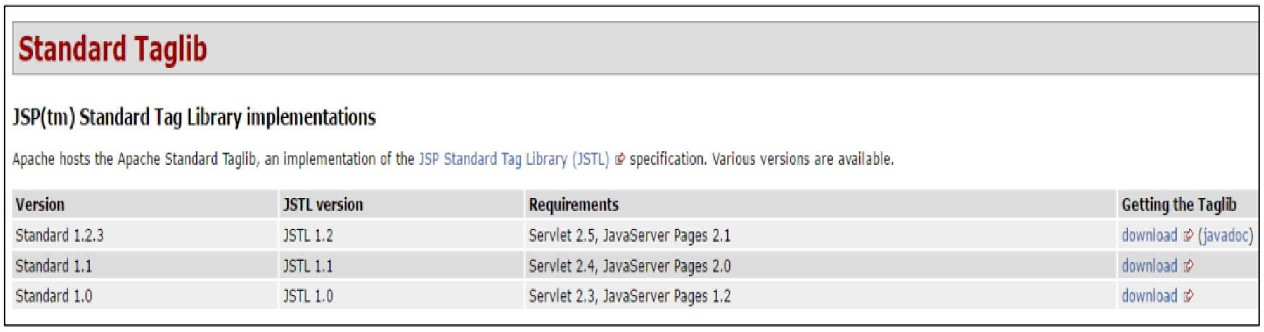


Apache Taglibs

This project is an open source repository for JSP(tm) Tag Libraries.

In particular, Apache Taglibs hosts the Apache Standard Taglib, an i
JSTL are implemented, and a 1.2 implementation is in the works.

2. Click on *Apache Standard Taglib*.



Standard Taglib

JSP(tm) Standard Tag Library implementations

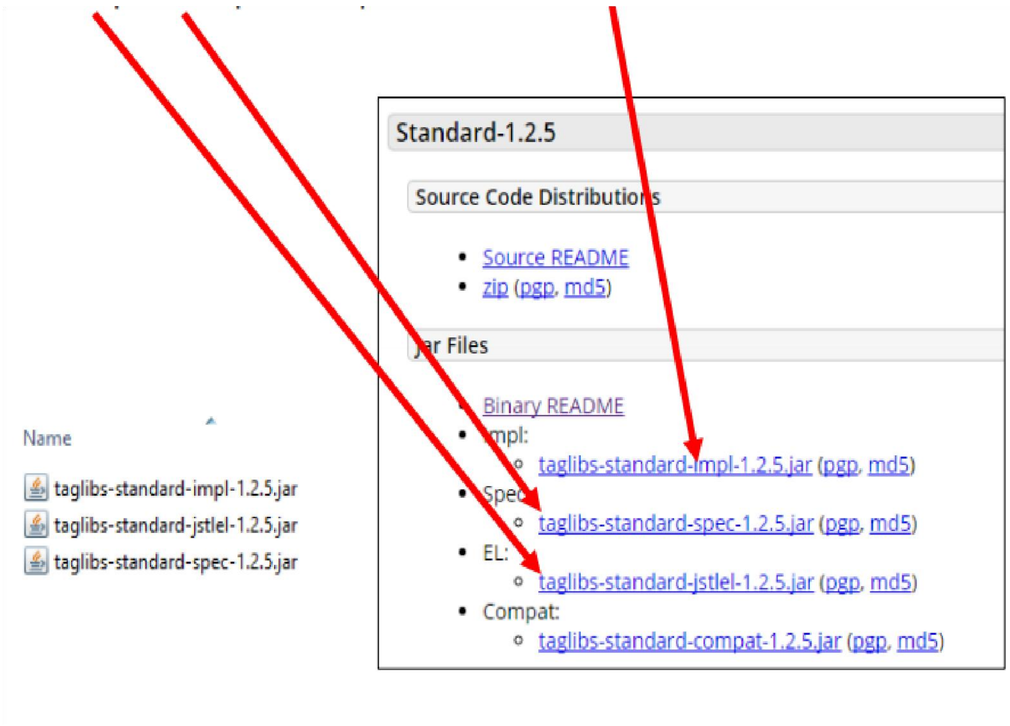
Apache hosts the Apache Standard Taglib, an implementation of the JSP Standard Tag Library (JSTL) specification. Various versions are available.

Version	JSTL version	Requirements	Getting the Taglib
Standard 1.2.3	JSTL 1.2	Servlet 2.5, JavaServer Pages 2.1	download (javadoc)
Standard 1.1	JSTL 1.1	Servlet 2.4, JavaServer Pages 2.0	download
Standard 1.0	JSTL 1.0	Servlet 2.3, JavaServer Pages 1.2	download

3. Choose Version 1.2 and click download link.

4. Click to jar files for Impl and save the link.

5. Repeat step 3 for Spec and EL.


Name

taglibs-standard-impl-1.2.5.jar
taglibs-standard-jstlel-1.2.5.jar
taglibs-standard-spec-1.2.5.jar

Standard-1.2.5

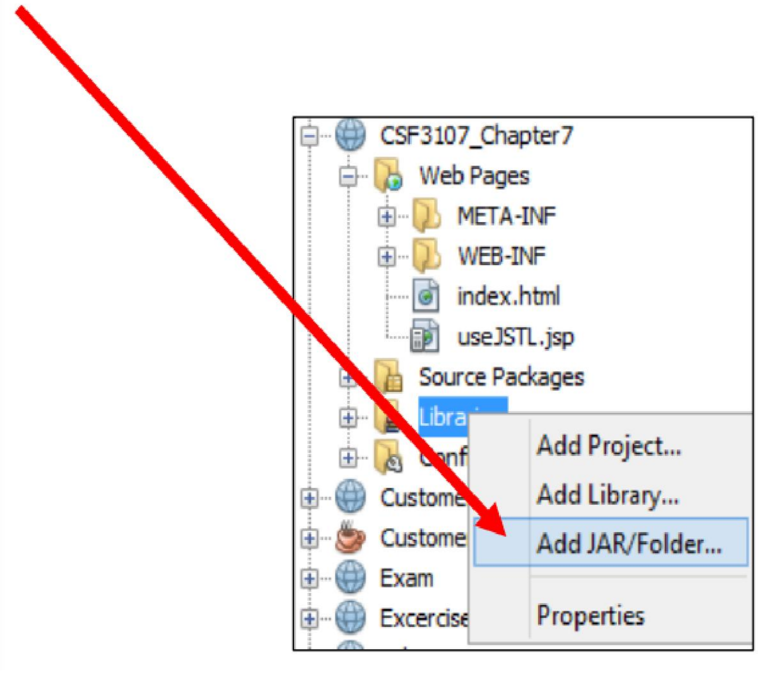
Source Code Distributions

- Source README
- zip (pgp, md5)

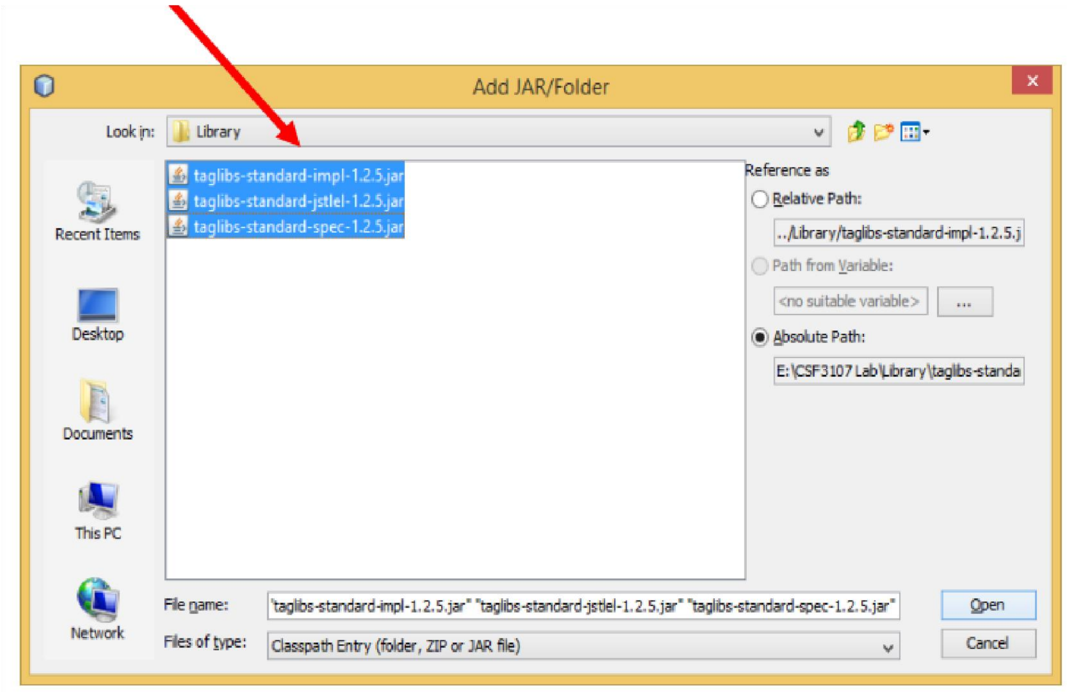
Jar Files

- Binary README
- Impl:
  - taglibs-standard-impl-1.2.5.jar (pgp, md5)
- Spec:
  - taglibs-standard-spec-1.2.5.jar (pgp, md5)
- EL:
  - taglibs-standard-jstlel-1.2.5.jar (pgp, md5)
- Compat:
  - taglibs-standard-compat-1.2.5.jar (pgp, md5)

6. Go to your project -> Libraries.
7. Right click your mouse and click to Add JAR/Folder.



8. Choose taglibs-standard-impl-1.2.5.jar, taglibs-standard-jstlel-1.2.5.jar and taglibs-standard-spec-1.2.5.jar and click Open button.

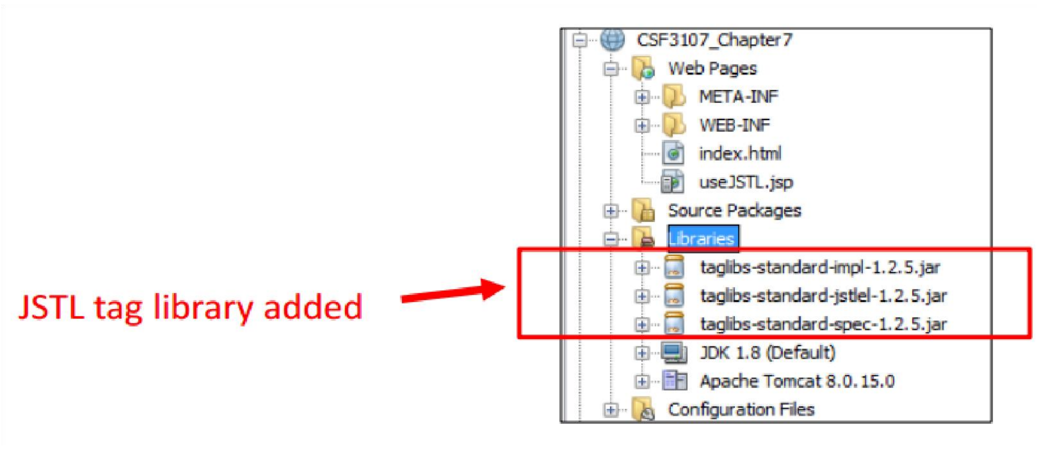


taglibs-standard-impl-1.2.5.jar
taglibs-standard-jstlel-1.2.5.jar
taglibs-standard-spec-1.2.5.jar

Relative Path: ../Library/taglibs-standard-impl-1.2.5.j
Path from Variable: <no suitable variable>
Absolute Path: E:\CSF3107 Lab\Library\taglibs-standa

File name: "taglibs-standard-impl-1.2.5.jar" "taglibs-standard-jstlel-1.2.5.jar" "taglibs-standard-spec-1.2.5.jar"
Files of type: Classpath Entry (folder, ZIP or JAR file)

9. You will see all jars files; taglibs-standard-impl-1.2.5.jar, taglibsstandardjstlel-1.2.5.jar and taglibs-standard-spec-1.2.5.jar successfully added in Libraries's folder in NetBeans.



```
CSF3107_Chapter7
  Web Pages
    META-INF
    WEB-INF
    index.html
    useJSTL.jsp
  Source Packages
  Libraries
    taglibs-standard-impl-1.2.5.jar
    taglibs-standard-jstlel-1.2.5.jar
    taglibs-standard-spec-1.2.5.jar
    JDK 1.8 (Default)
    Apache Tomcat 8.0.15.0
  Configuration Files
```

Reflection

What you have learnt from this exercise?

- JSP tag libraries (taglibs) extend the capabilities of JSP by allowing the use of custom tags. This helps to reduce the amount of Java code within JSP pages, making the pages easier to read and maintain.

Task 4: Using Java Standard Tag Library (JSTL)

Objective: Using JSTL core tags directive to retrieve information from one page and display it in another JSP page and format number, currency.

Problem Description:
1. To set the value “Welcome to CSF3107 – Web Programming courses..!” and display the value using JSTL core.
2. To passing information from one page to another using JSTL core
   i. Create userRegistration.html to key-in information.
   ii. Create userHandler.jsp to receive information key-in from userRegistration.html.
3. Using JSTL’s fmt to format number and currency.

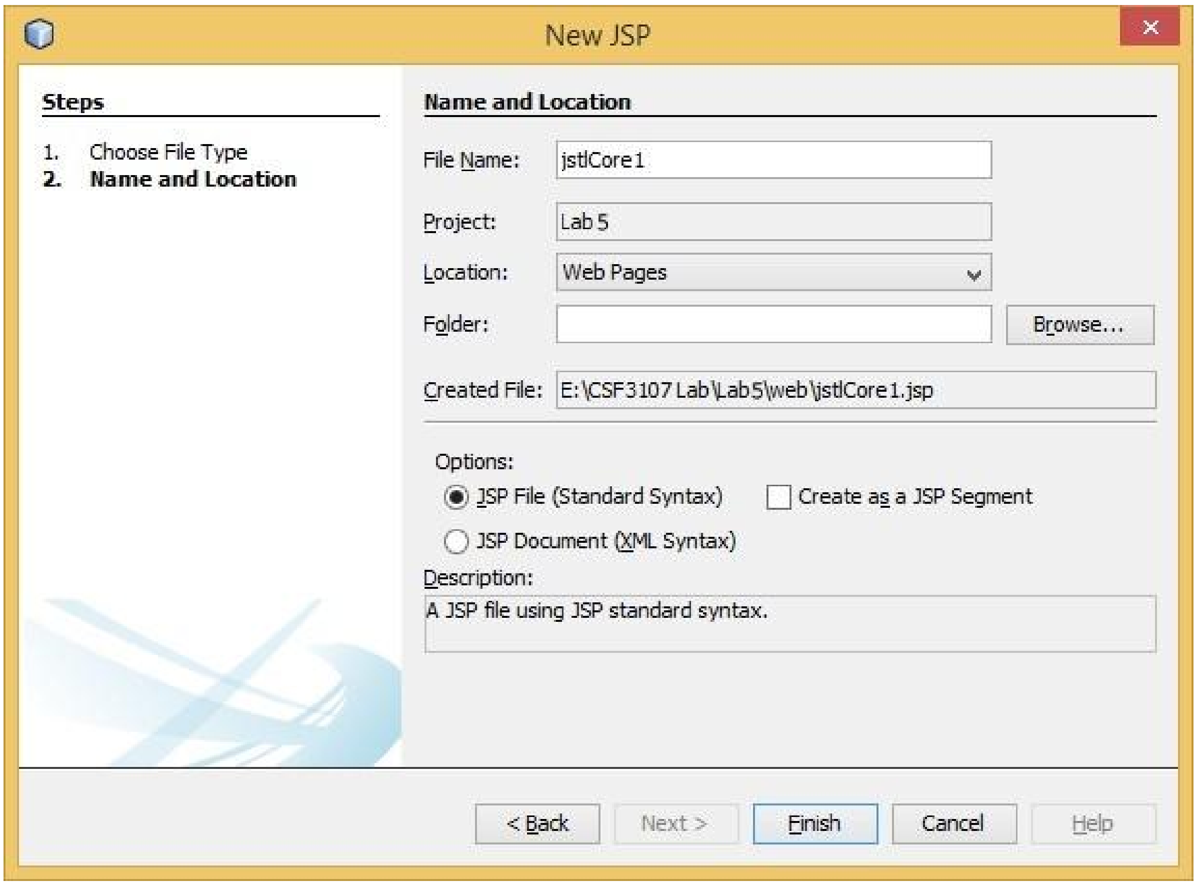
Estimated 30 minutes **time:**

Prerequisite: Must attached JSTL’s library to current Project’s directory (Refer to Chapter 7’s notes).

<u>Task 1 Assign value “ W elcome to CS F3107-Web Programming 2 courses” and display the value inside J SP page</u>

1. Go to project *Lab5*.

2. Create a new JSP’s file as *jstlCore1*.



New JSP

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

File Name: jstlCore1

Project: Lab 5

Location: Web Pages

Folder:

Created File: E:\CSF3107 Lab\Lab5\web\jstlCore1.jsp

Options:

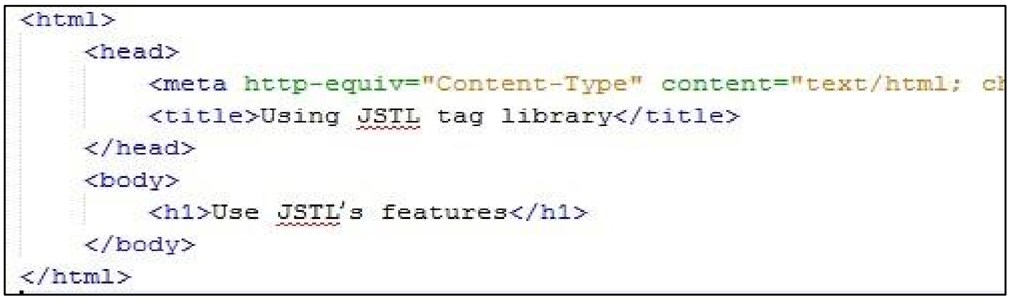
(●) JSP File (Standard Syntax) ☐ Create as a JSP Segment

( ) JSP Document (XML Syntax)

Description:

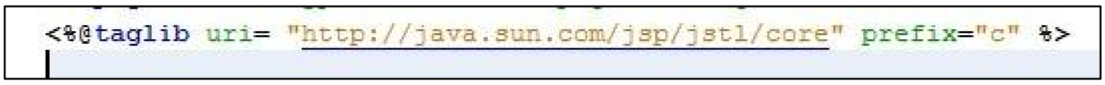
A JSP file using JSP standard syntax.

3. Rename the title as *Using JSTL tag library* and *<h1>* as *Use JSTL's features*.



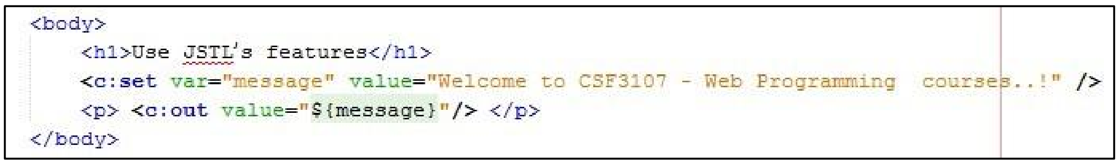
```
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; ch
        <title>Using JSTL tag library</title>
    </head>
    <body>
        <h1>Use JSTL's features</h1>
    </body>
</html>
```

4. Before using JSTL's tag, we need to declare JSTL's library tag using Taglib directive.



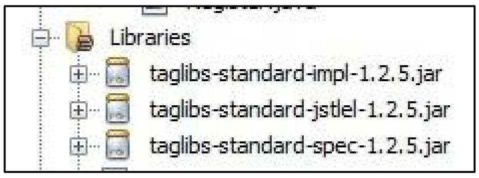
```
<%@taglib uri= "http://java.sun.com/jsp/jstl/core" prefix="c" %>
```

5. Using JSTL *core* tag, assign the value "*Welcome to CSF3107 – Web Programming courses..!*" to specific variable and finally, display the message.



```
<body>
    <h1>Use JSTL's features</h1>
    <c:set var="message" value="Welcome to CSF3107 - Web Programming  courses..!" />
    <p> <c:out value="${message}"/> </p>
</body>
```

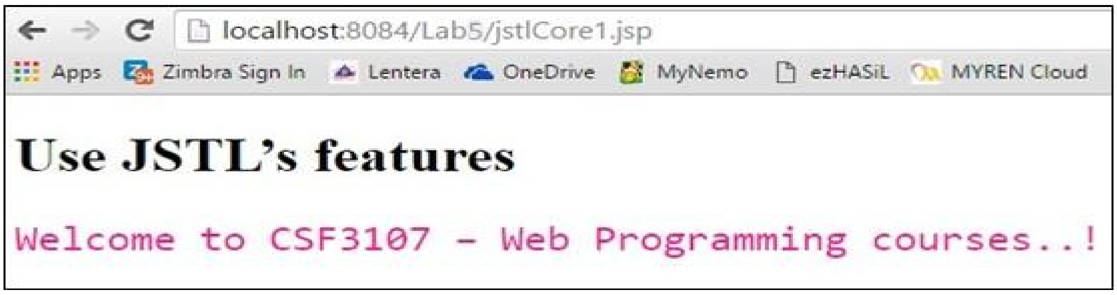
6. Before you compile *jstlCore1.jsp*, please ensure you already attach jar file for JSTL tag library.



```
Libraries
  taglibs-standard-impl-1.2.5.jar
  taglibs-standard-jstlel-1.2.5.jar
  taglibs-standard-spec-1.2.5.jar
```

7. Compile and run *jstlCore1.jsp* page.

8. You will get the following output.



localhost:8084/Lab5/jstlCore1.jsp

Use JSTL's features

Welcome to CSF3107 - Web Programming courses..!

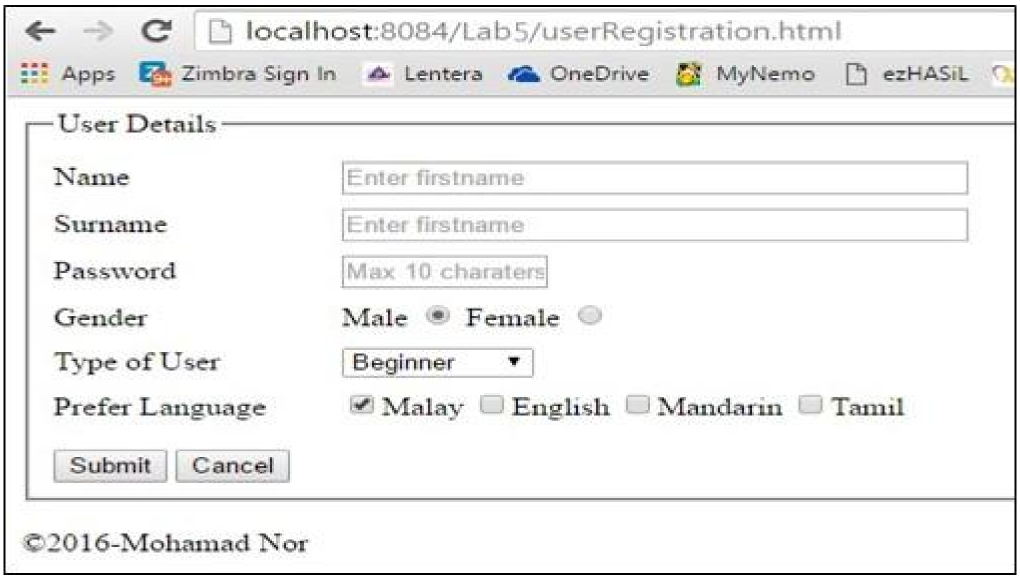
<u>Task 2-Using JSTL's core to request information from one page and display the information</u>

<u>Step 1</u>

1. Go to Lab8's project.

2. Create HTML's file and rename as userRegistration.

3. Produce the following HTML's form.



localhost:8084/Lab5/userRegistration.html

User Details

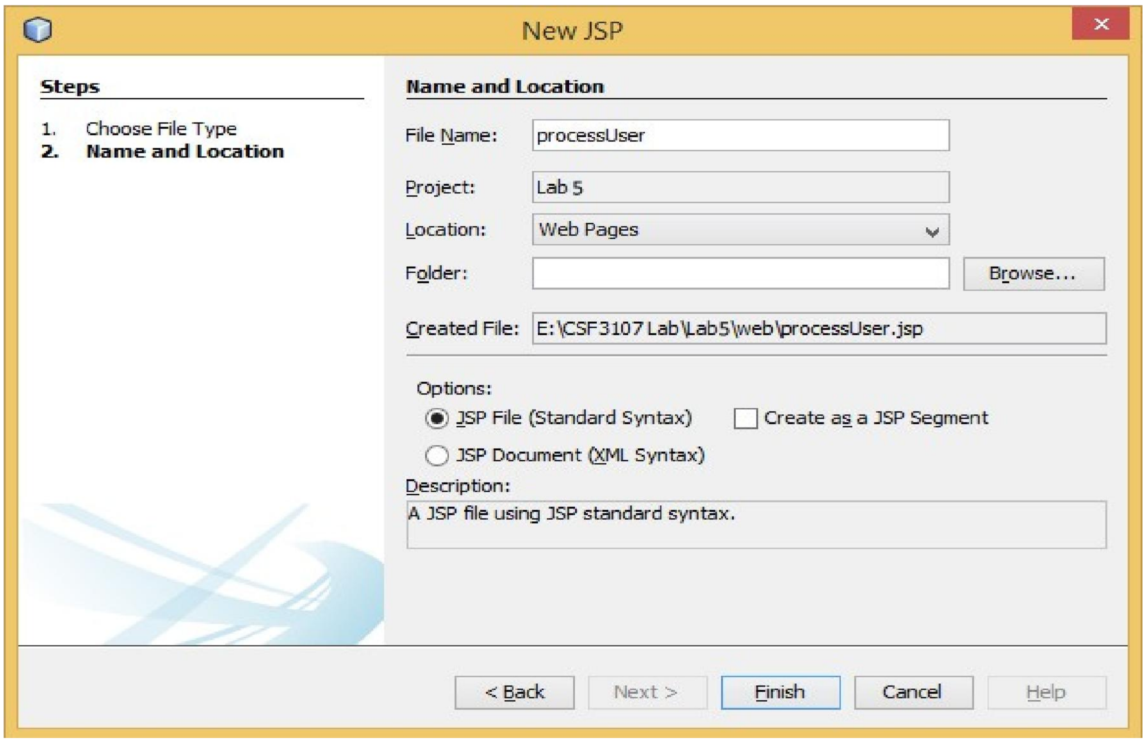
Name	Enter firstname
Surname	Enter firstname
Password	Max 10 charaters
Gender	Male ● Female ○
Type of User	Beginner
Prefer Language	✓ Malay ☐ English ☐ Mandarin ☐ Tamil

©2016-Mohamad Nor

Note: Types of user is *Beginner, Intermediate* and *Advanced*

Step 2

1. Create JSP's file and rename as *processUser*.



New JSP

Steps
1. Choose File Type
2. **Name and Location**

Name and Location

File Name:	processUser
Project:	Lab 5
Location:	Web Pages
Folder:	
Created File:	E:\CSF3107 Lab\Lab5\web\processUser.jsp

Options:
● JSP File (Standard Syntax) ☐ Create as a JSP Segment
○ JSP Document (XML Syntax)

Description:
A JSP file using JSP standard syntax.

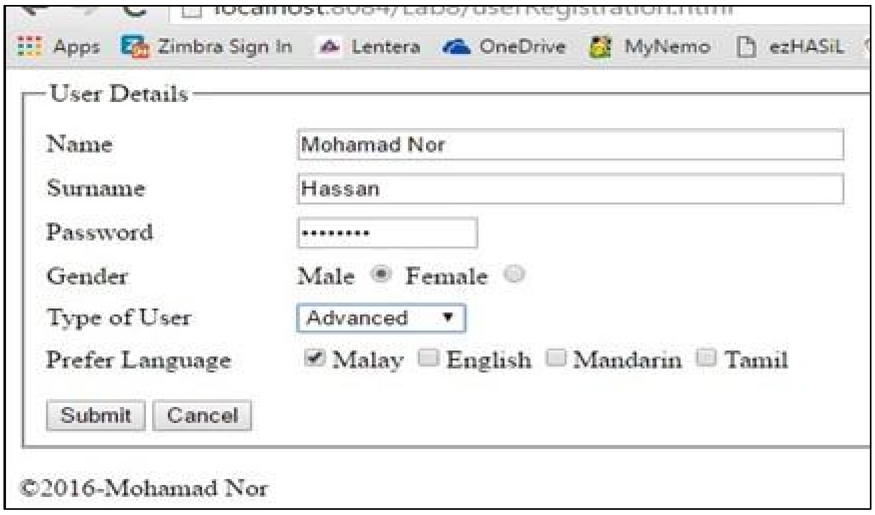
2. Add JSTL taglib directive into the *processUser.jsp's* page.

3. Retrieve the information by using **c:param** tag and display the

   information by using **c:out** tag.

4. Compile *userHandler.jsp* page.

5. Run *userRegistration.html* page and key-in the information.


localhost:8084/Lab8/userRegistration.html

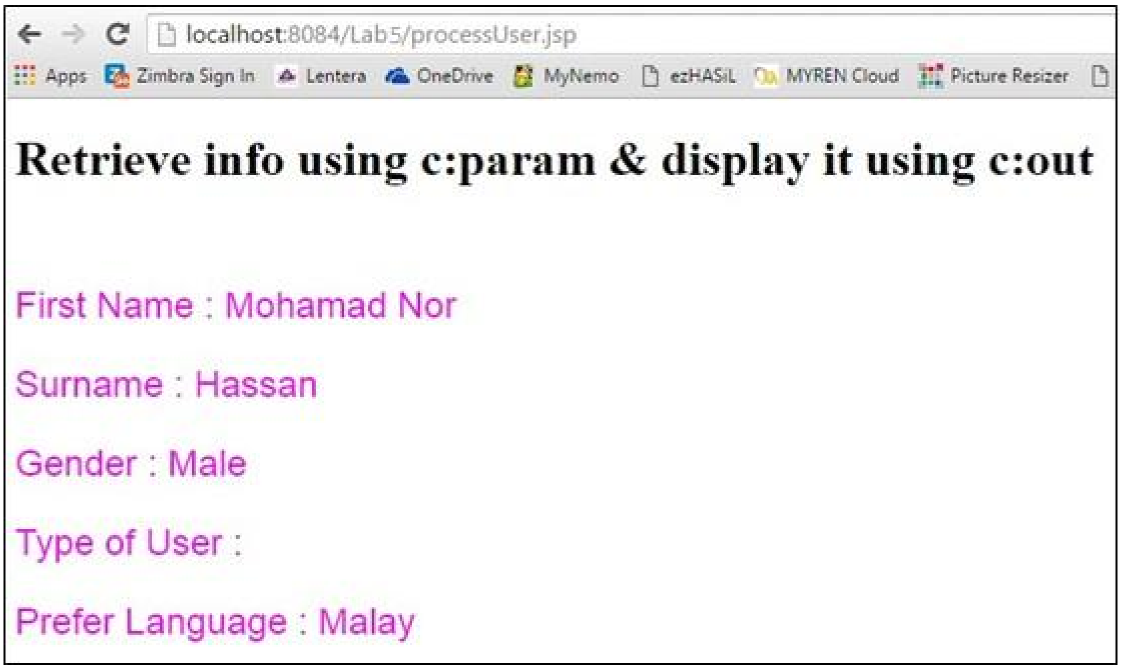
User Details

Name	Mohamad Nor
Surname	Hassan
Password	••••••••
Gender	Male ● Female ○
Type of User	Advanced
Prefer Language	[✓ Malay] English Mandarin Tamil

Submit Cancel

©2016-Mohamad Nor

6. Click *Submit* button.
7. You will get the following output.


localhost:8084/Lab5/processUser.jsp

Retrieve info using c:param & display it using c:out

First Name : Mohamad Nor

Surname : Hassan

Gender : Male

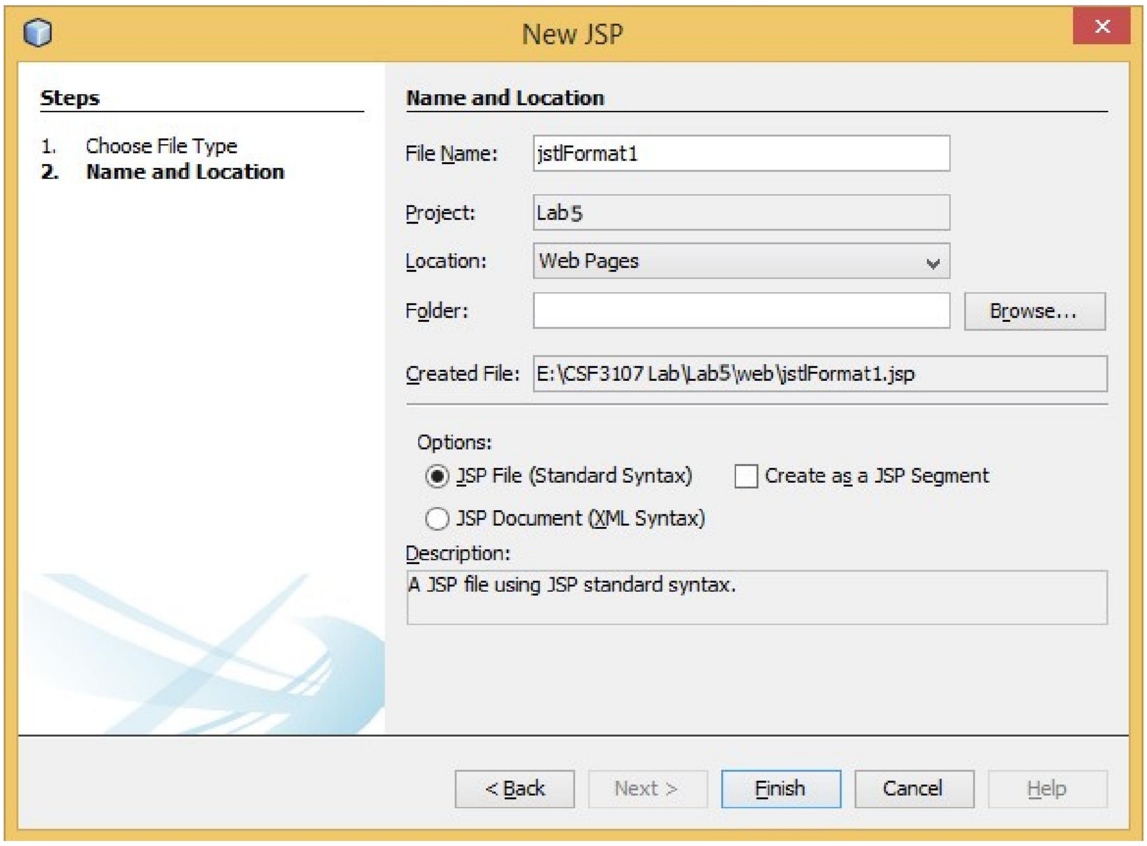
Type of User :

Prefer Language : Malay

Task 3 - Using JSTL's fmt to format number and currency

Step 1

1. Create JSP's file and rename as *jstlFormat1*.


New JSP

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

File Name: jstlFormat1

Project: Lab5

Location: Web Pages

Folder:

Created File: E:\CSF3107 Lab\Lab5\web\jstlFormat1.jsp

Options:

[✓ JSP File (Standard Syntax)] Create as a JSP Segment

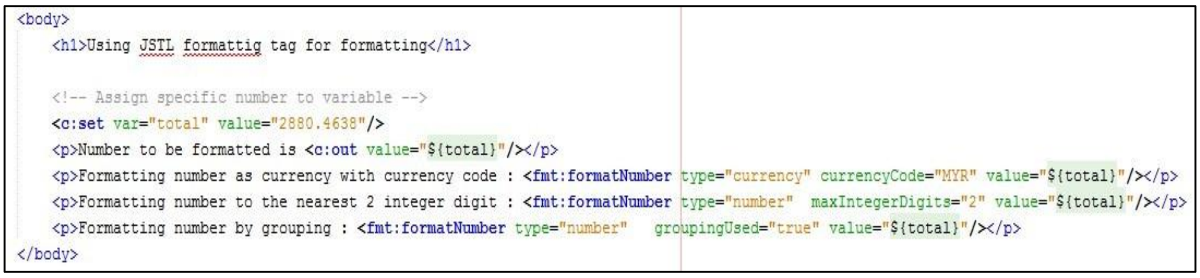
JSP Document (XML Syntax)

Description:

A JSP file using JSP standard syntax.

2. Add JSTL taglib directive for JSTL *core* and JSTL *formatting* into the current code.

3. Add the following code in your page.


```
<body>
    <h1>Using JSTL formattig tag for formatting</h1>

    <!-- Assign specific number to variable -->
    <c:set var="total" value="2880.4638"/>
    <p>Number to be formatted is <c:out value="${total}"/></p>
    <p>Formatting number as currency with currency code : <fmt:formatNumber type="currency" currencyCode="MYR" value="${total}"/></p>
    <p>Formatting number to the nearest 2 integer digit : <fmt:formatNumber type="number"  maxIntegerDigits="2" value="${total}"/></p>
    <p>Formatting number by grouping : <fmt:formatNumber type="number"   groupingUsed="true" value="${total}"/></p>
</body>
```

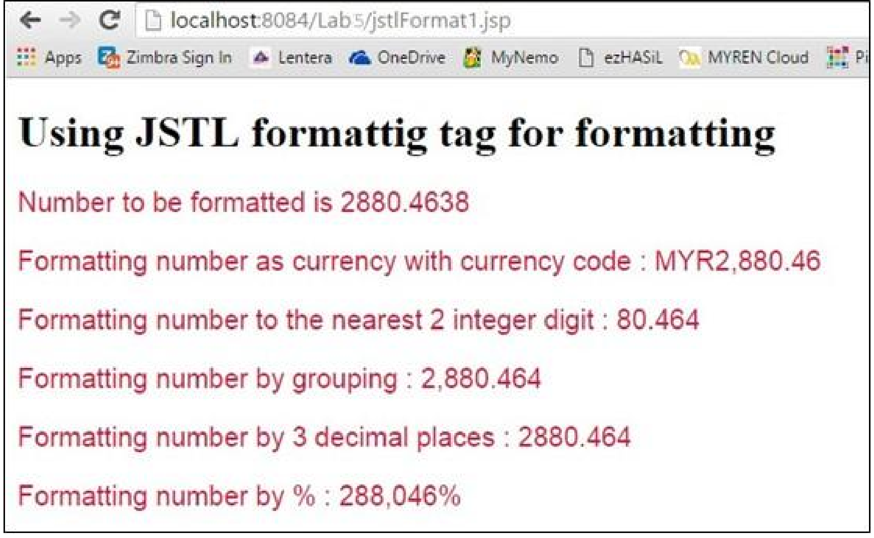
4. Continue to do formatting for

    - 3 decimal places
    - percentage

5. Save *jstlFormat1.jsp*

6. Compile and run *jstlFormat1.jsp*.

7. You will get the following output.



Using JSTL formattig tag for formatting

```
Number to be formatted is 2880.4638
Formatting number as currency with currency code : MYR2,880.46
Formatting number to the nearest 2 integer digit : 80.464
Formatting number by grouping : 2,880.464
Formatting number by 3 decimal places : 2880.464
Formatting number by % : 288,046%
```

Reflection

1. What the purpose of using JSTL's tag library?

 - provides a set of tags that simplify the development of JSP pages by encapsulating common tasks, such as iteration, conditionals, formatting, and database operations, into easy-to-use tags.

2. List **FIVE(5)** categories of JSTL library.

Task 5: Using JSP Standard Tag Library

Objective: Using JSTL tags for parsing and formatting the dates.

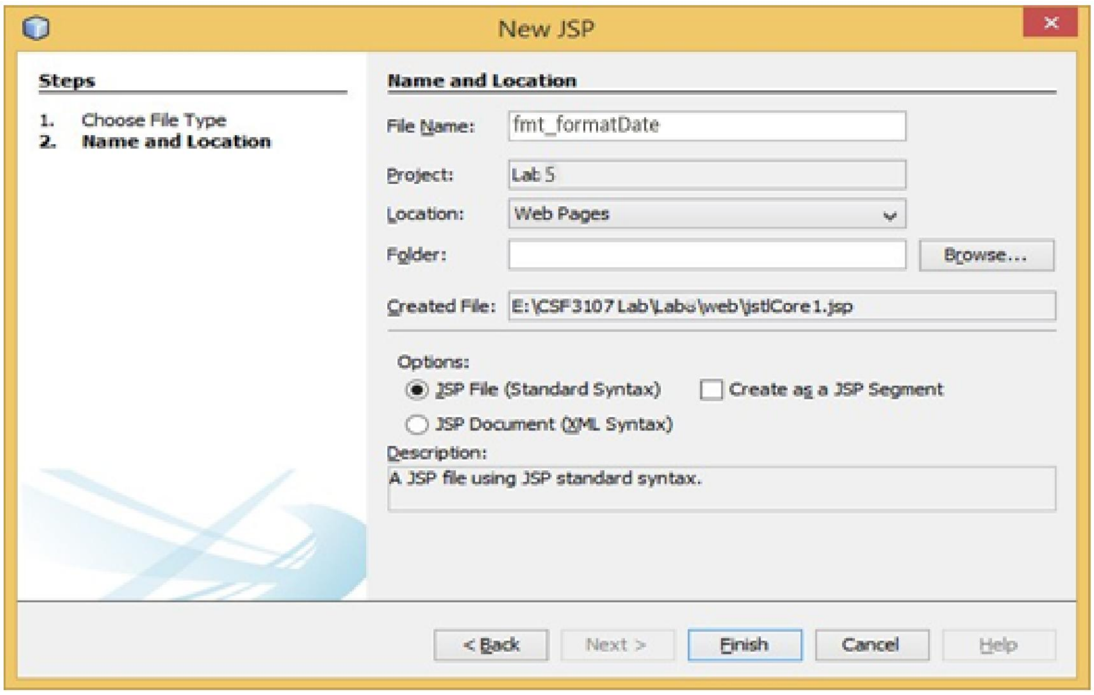
Problem 1. Using JSTL's fmt to format Dates. **Description:**
2. Using JSTL's fmt to parse Dates.

Estimated 30 minutes **time:**

Task 1 Using JSTL's fmt to format Date

1. Go to project *Lab5*.

2. Create a new JSP file as *fmt_formatDate*.


New JSP

Steps
1. Choose File Type
2. Name and Location

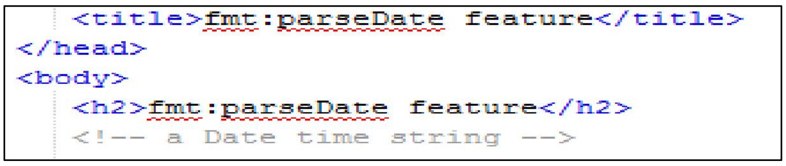
Name and Location

File Name: fmt_formatDate
Project: Lab 5
Location: Web Pages
Folder:
Created File: E:\CSF3107 Lab\Labs\web\jstlCore1.jsp

Options:
● JSP File (Standard Syntax) ☐ Create as a JSP Segment
○ JSP Document (XML Syntax)

Description:
A JSP file using JSP standard syntax.

2. Rename the title as *Using JSTL tag library* and *<h2>* as *Use fmt_formatDate features*.



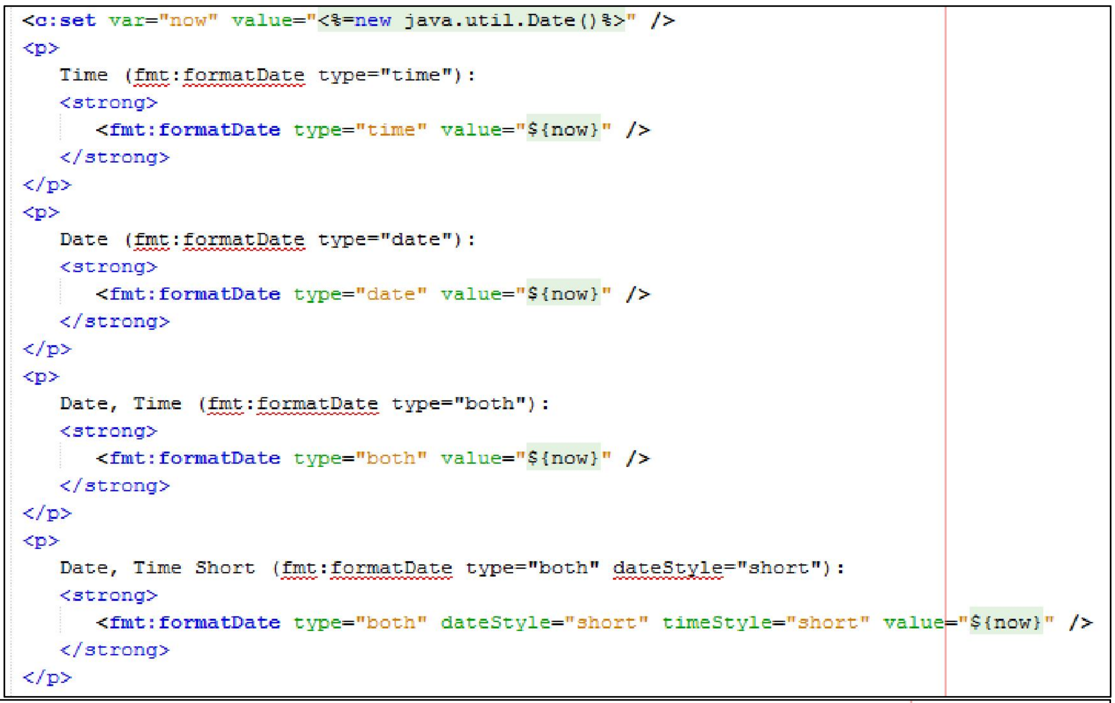
```
    <title>fmt:parseDate feature</title>
</head>
<body>
    <h2>fmt:parseDate feature</h2>
    <!-- a Date time string -->
```

4. Before using JSTL's tag, we need to declare JSTL's library tag using Taglib directive.

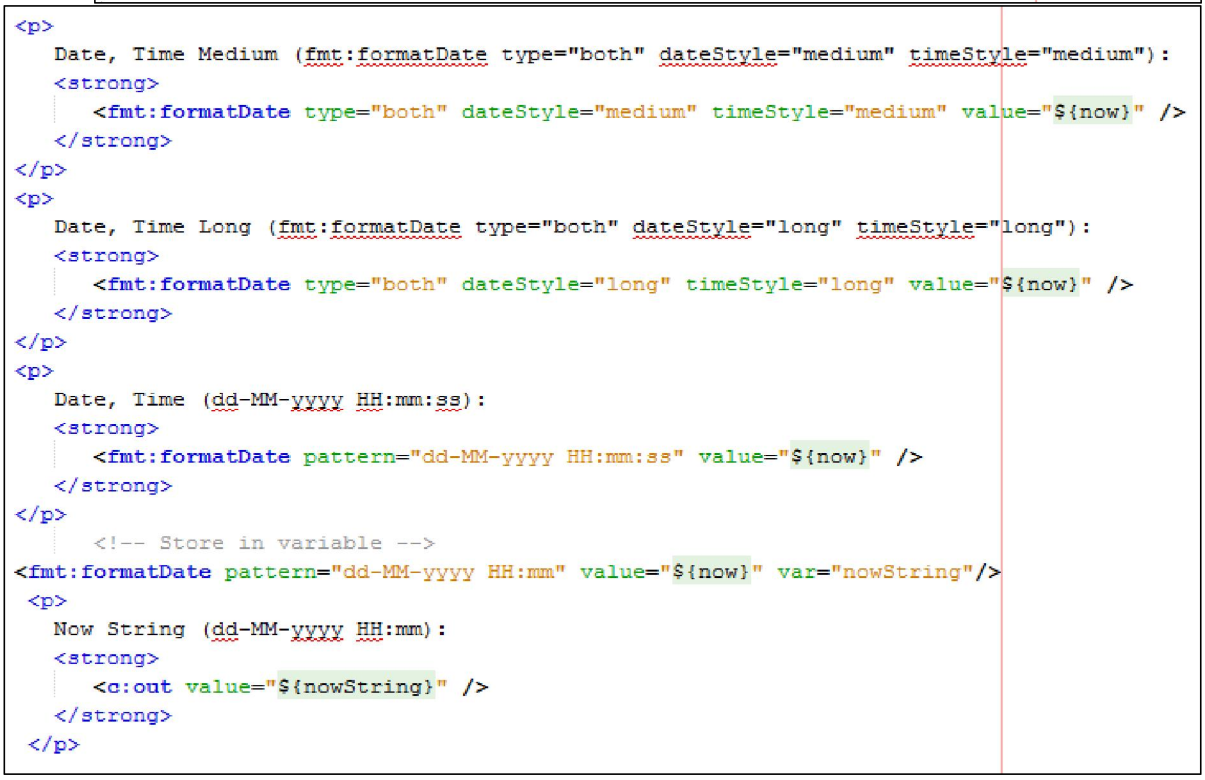
```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>
```

5. Add JSTL Taglib directive for JSTL core and JSTL formatting into the current code.

6. Add the following code in your page.



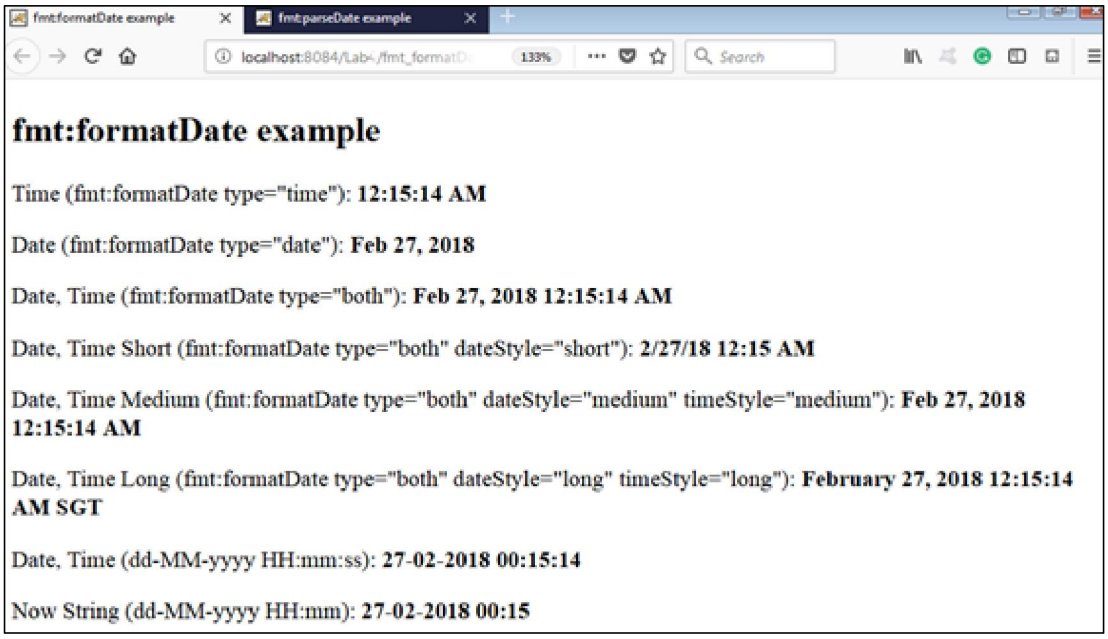
```
<c:set var="now" value="<%=new java.util.Date()%>" />
<p>
   Time (fmt:formatDate type="time"):
   <strong>
      <fmt:formatDate type="time" value="${now}" />
   </strong>
</p>
<p>
   Date (fmt:formatDate type="date"):
   <strong>
      <fmt:formatDate type="date" value="${now}" />
   </strong>
</p>
<p>
   Date, Time (fmt:formatDate type="both"):
   <strong>
      <fmt:formatDate type="both" value="${now}" />
   </strong>
</p>
<p>
   Date, Time Short (fmt:formatDate type="both" dateStyle="short"):
   <strong>
      <fmt:formatDate type="both" dateStyle="short" timeStyle="short" value="${now}" />
   </strong>
</p>
```



```
<p>
   Date, Time Medium (fmt:formatDate type="both" dateStyle="medium" timeStyle="medium"):
   <strong>
      <fmt:formatDate type="both" dateStyle="medium" timeStyle="medium" value="${now}" />
   </strong>
</p>
<p>
   Date, Time Long (fmt:formatDate type="both" dateStyle="long" timeStyle="long"):
   <strong>
      <fmt:formatDate type="both" dateStyle="long" timeStyle="long" value="${now}" />
   </strong>
</p>
<p>
   Date, Time (dd-MM-yyyy HH:mm:ss):
   <strong>
      <fmt:formatDate pattern="dd-MM-yyyy HH:mm:ss" value="${now}" />
   </strong>
</p>
      <!-- Store in variable -->
<fmt:formatDate pattern="dd-MM-yyyy HH:mm" value="${now}" var="nowString"/>
 <p>
   Now String (dd-MM-yyyy HH:mm):
   <strong>
      <c:out value="${nowString}" />
   </strong>
 </p>
```

7. Compile and run *fmt_formatDate.jsp* page.

8. You will get the following output.

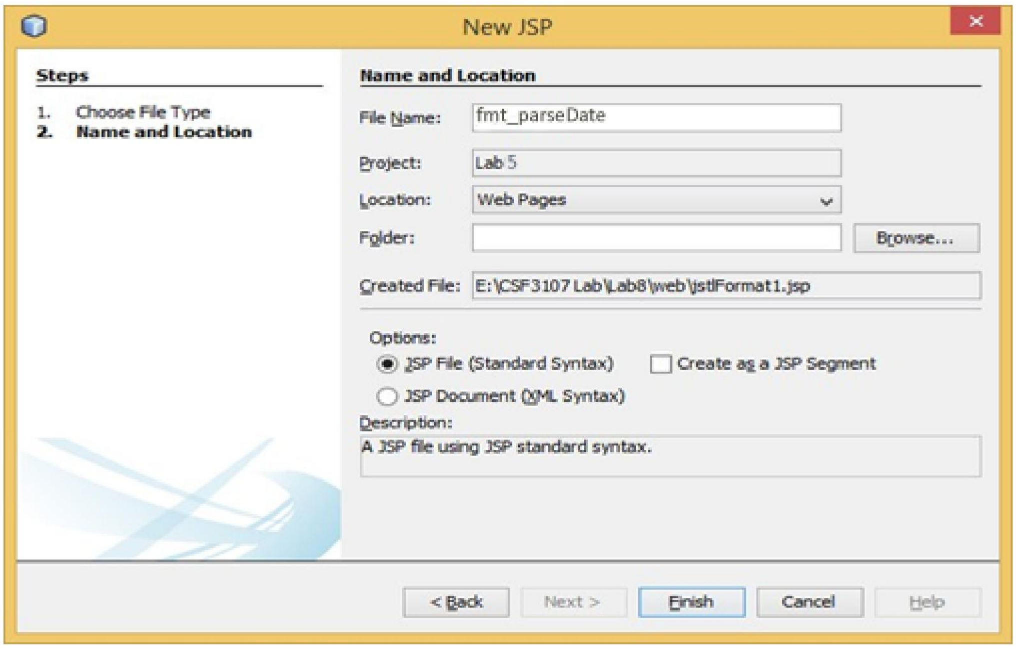


```
fmt:formatDate example

Time (fmt:formatDate type="time"): 12:15:14 AM

Date (fmt:formatDate type="date"): Feb 27, 2018

Date, Time (fmt:formatDate type="both"): Feb 27, 2018 12:15:14 AM

Date, Time Short (fmt:formatDate type="both" dateStyle="short"): 2/27/18 12:15 AM

Date, Time Medium (fmt:formatDate type="both" dateStyle="medium" timeStyle="medium"): Feb 27, 2018
12:15:14 AM

Date, Time Long (fmt:formatDate type="both" dateStyle="long" timeStyle="long"): February 27, 2018 12:15:14
AM SGT

Date, Time (dd-MM-yyyy HH:mm:ss): 27-02-2018 00:15:14

Now String (dd-MM-yyyy HH:mm): 27-02-2018 00:15
```

Task 2 - Using JSTL’s fmt to Parse Date

Step 1

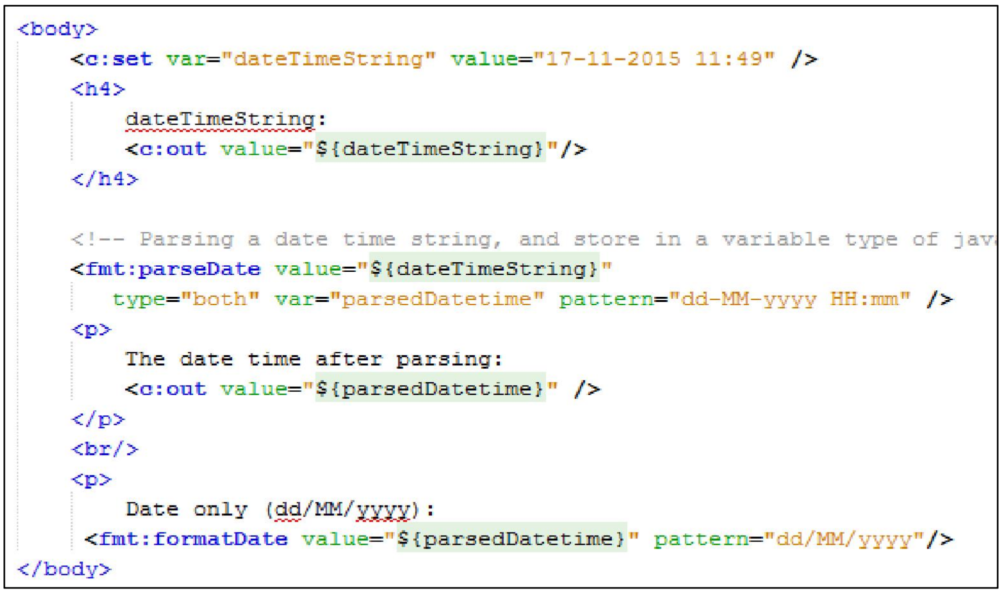
1. Create JSP’s file and rename as *fmt_parseDate*.



File Name: fmt_parseDate
Project: Lab 5
Location: Web Pages
Folder:
Created File: E:\CSF3107 Lab\Lab8\web\jstlFormat1.jsp
Options:
(●) JSP File (Standard Syntax) ☐ Create as a JSP Segment
( ) JSP Document (XML Syntax)
Description:
A JSP file using JSP standard syntax.

2. Add JSTL taglib directive for JSTL *core* and JSTL *formatting* into the current code.

3. Add the following code in your page.

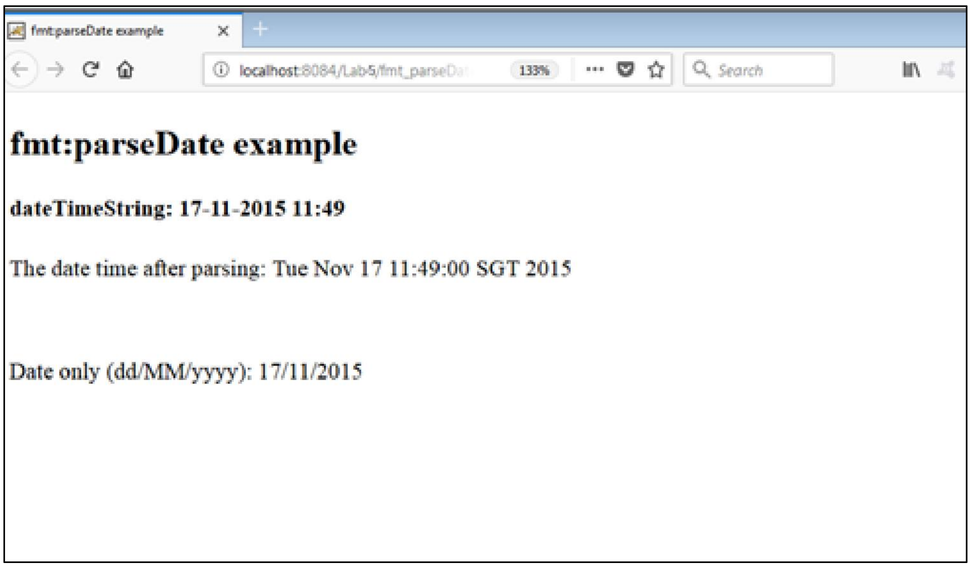


```
<body>
    <c:set var="dateTimeString" value="17-11-2015 11:49" />
    <h4>
        dateTimeString:
        <c:out value="${dateTimeString}"/>
    </h4>

    <!-- Parsing a date time string, and store in a variable type of jav
    <fmt:parseDate value="${dateTimeString}"
       type="both" var="parsedDatetime" pattern="dd-MM-yyyy HH:mm" />
    <p>
        The date time after parsing:
        <c:out value="${parsedDatetime}" />
    </p>
    <br/>
    <p>
        Date only (dd/MM/yyyy):
     <fmt:formatDate value="${parsedDatetime}" pattern="dd/MM/yyyy"/>
</body>
```

5. Save *fmt_parseDate.jsp*

6. Compile and run *fmt_parseDate.jsp*.

7. You will get the following output.



```
fmt:parseDate example

dateTimeString: 17-11-2015 11:49

The date time after parsing: Tue Nov 17 11:49:00 SGT 2015


Date only (dd/MM/yyyy): 17/11/2015
```

Reflection

1. What you have learnt from this exercise?

- Understanding how to include JSTL core and formatting libraries using the <%@ taglib %> directive.

Exercise

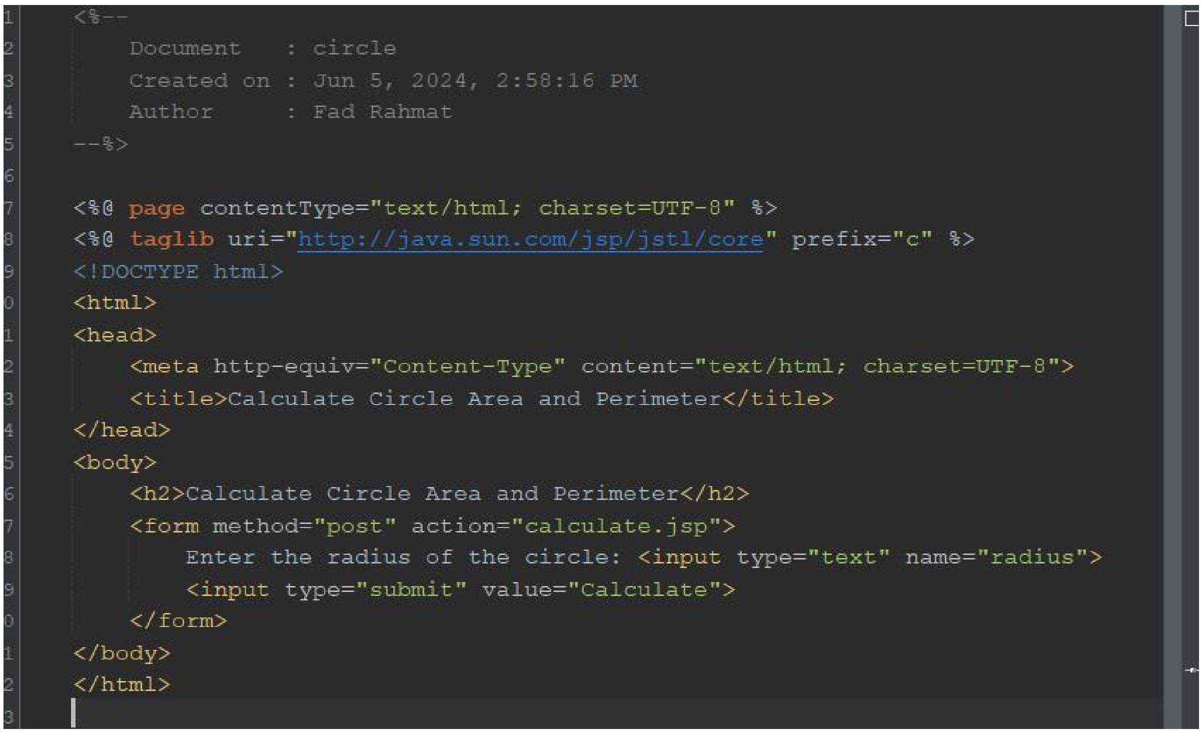
1. Write a JSP's form that asks the user to key-in the radius of circle. The program should calculate the area of circle and the perimeter of circle. Finally, use JSTL library to format your result into 3 decimal places.

2. Rahim bought 800 shares of stock at a price of RM10.50 per share. He must pay her stock broker a 5 percent commission for the transaction.
   Write a web based program that calculates and displays the following:
   - The amount paid for the stock alone without the commission.
   - The amount of the commission.
   - The total amount paid (for the stock plus the commission).

You should use JavaBeans to implement business logic and JSTL for display purposes.

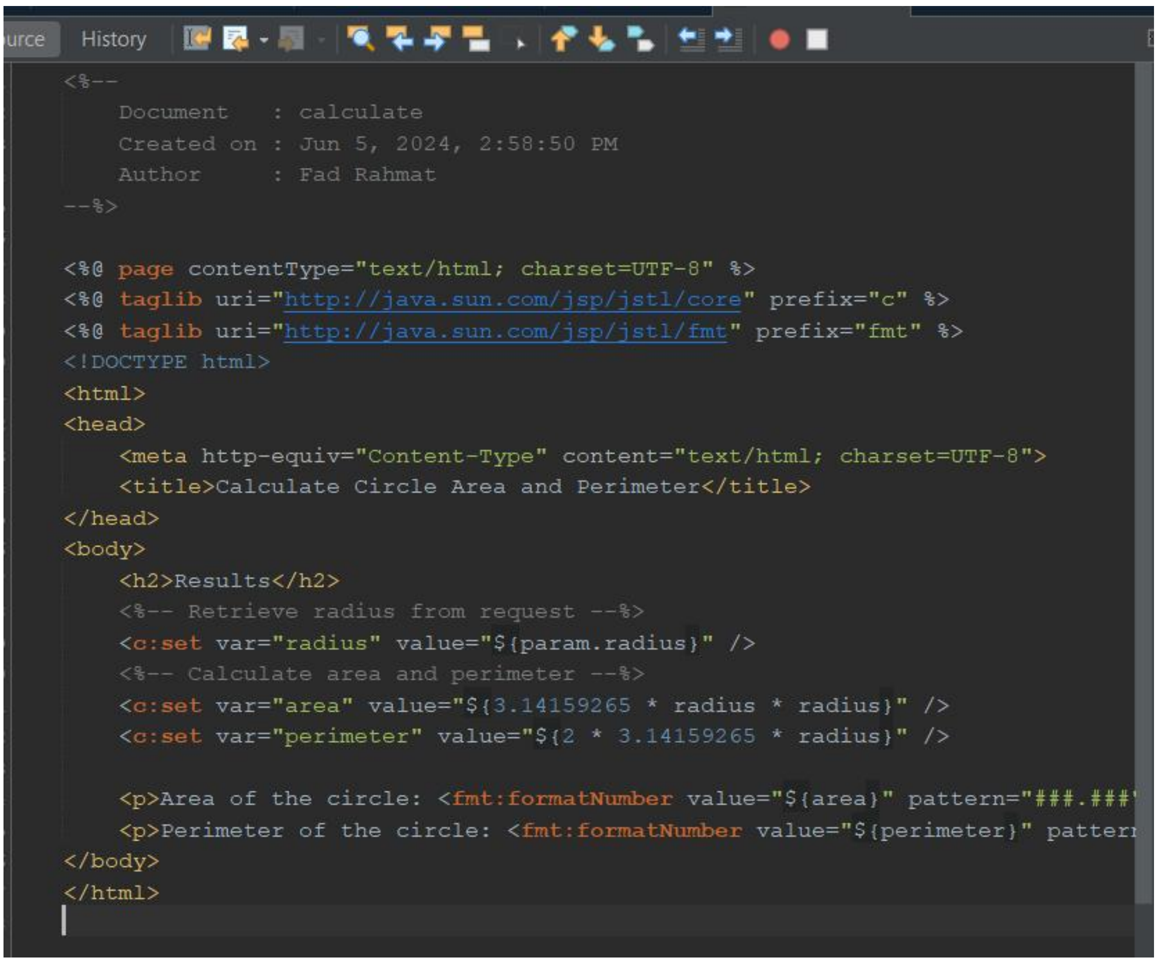
Code:

Circle.jsp:



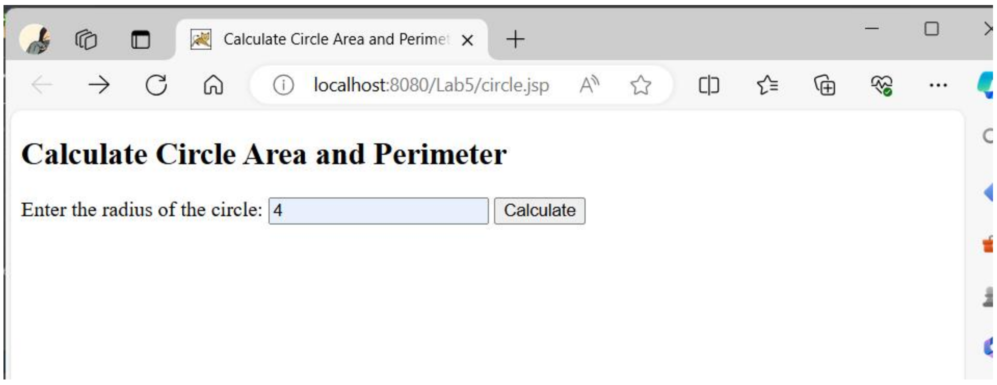
```
<%--
    Document   : circle
    Created on : Jun 5, 2024, 2:58:16 PM
    Author     : Fad Rahmat
--%>

<%@ page contentType="text/html; charset=UTF-8" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Calculate Circle Area and Perimeter</title>
</head>
<body>
    <h2>Calculate Circle Area and Perimeter</h2>
    <form method="post" action="calculate.jsp">
        Enter the radius of the circle: <input type="text" name="radius">
        <input type="submit" value="Calculate">
    </form>
</body>
</html>
```

Calculate.jsp:



```
<%--
    Document   : calculate
    Created on : Jun 5, 2024, 2:58:50 PM
    Author     : Fad Rahmat
--%>

<%@ page contentType="text/html; charset=UTF-8" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Calculate Circle Area and Perimeter</title>
</head>
<body>
    <h2>Results</h2>
    <%-- Retrieve radius from request --%>
    <c:set var="radius" value="${param.radius}" />
    <%-- Calculate area and perimeter --%>
    <c:set var="area" value="${3.14159265 * radius * radius}" />
    <c:set var="perimeter" value="${2 * 3.14159265 * radius}" />

    <p>Area of the circle: <fmt:formatNumber value="${area}" pattern="###.###'
    <p>Perimeter of the circle: <fmt:formatNumber value="${perimeter}" patter
</body>
</html>
```

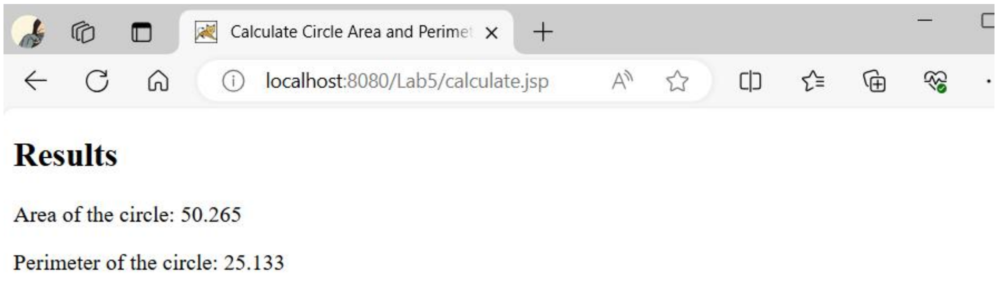
Output:



Calculate Circle Area and Perime
localhost:8080/Lab5/circle.jsp

Calculate Circle Area and Perimeter

Enter the radius of the circle: 4 Calculate



Calculate Circle Area and Perime
localhost:8080/Lab5/calculate.jsp

Results

Area of the circle: 50.265

Perimeter of the circle: 25.133