Java Server Pages v2.2

Lab Book

Document Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Revision No. | Author | **Summary of Changes** |
| Nov-2009 | 2.0 | PradnyaJagtap, Habib Shaikh, and Mahima Sharma | Revamped from JSP 2.0 to JSP 2.1 |
| 25-Nov-2009 |  | CLS Team | Review  Template application |
| Feb-2010 | 2.1 | Mahima Sharma | Refined based on feedback received from faculties. |
| Sep-2012 | 2.2 | Sarbananda Behera | Refined to make problem based lab exercises |
| Sep - 2014 | 2.3 | Anjulata | Refinement in problem statements. |
| May-2015 | 2.4 | Anjulata | Revamped from JSP 2.1 to JSP 2.2 |
| May-2016 | 2.5 | Anjulata | Refinement in problem statements as per new ToC |

Table of Contents

[Document Revision History 2](#_Toc452124095)

[Table of Contents 3](#_Toc452124096)

[Getting Started 4](#_Toc452124097)

[Overview 4](#_Toc452124098)

[Setup Checklist for JSP 4](#_Toc452124099)

[Lab 1. JSP Basics 5](#_Toc452124100)

[1.1: Developing Simple JSP Page 5](#_Toc452124101)

[1.2: JSP Scripting Elements and Method Declarations 5](#_Toc452124102)

[1.3 : JSP Directive 6](#_Toc452124103)

[Lab 2. JSP Actions 7](#_Toc452124104)

[2.1: Dynamic Include action 7](#_Toc452124105)

[2.2: Integrating Servlet, JSP, and Java Bean (Implementing JSP Model 2 – Model-View-Controller pattern) 8](#_Toc452124106)

[2.3: JSP configuration using web.xml 14](#_Toc452124107)

[Lab 3. JSP Standard Tags Libraries (JSTL) 15](#_Toc452124108)

[3.1: Using Core tag library : <c:set> and <c:out> 15](#_Toc452124109)

[3.2: Using Core tag tag library: <c:if> 15](#_Toc452124110)

[3.3: Using Lambda Expression with EL 16](#_Toc452124111)

[Appendix a: 16](#_Toc452124112)

Getting Started

## Overview

This lab book is an unguided tour for learning Java Server Pages version 2.2. It comprises of problem statements and diagrams to complete a sequences of exercises. Flow diagrams and screen snap shots are provided wherever necessary. It will expose you to create Java Server Pages Applications using eclipse IDE, deploy them on WildFly server and invoke from browser.

## Setup Checklist for JSP

Here is what is expected on your machine in order for the lab to work.

Minimum System Requirements

* Intel Core i5 or higher.
* Microsoft Windows 7/8 or higher.
* Memory: 2GB of RAM (more recommended)
* 500MB hard disk space
* VGA or higher resolution monitor
* Mouse or other pointing device
* JDK version 1.8 with help
* Netscape or IE web browser
* Eclipse 4.4(Luna)
* Wildfly Server version 8.x

Please ensure that the following is done:

* JDK 1.8 is installed.
* Wildfly Server is installed.( Wildfly8.x)
* Eclipse 4.4(Luna) or higher version with JEE support with Wildfly8.x Adapter is installed.

1. JSP Basics

|  |  |
| --- | --- |
| **Goals** | * Create simple JSP programs. |
| **Time** | 2 hrs. |

## 1.1: Developing Simple JSP Page

**Problem:**

Develop a simple JSP application to show the current Date and Time from Server.

[Note: Use Java8 Date and Time API]

## 1.2: JSP Scripting Elements and Method Declarations

**Problem:**

Create a JSP page to accept username and password from user as shown in figure 1. Authenticate the user using JSP scriptlets and display appropriate success or error message on a new page.

Authentication should be done using a java method that is declaredin the JSP page.

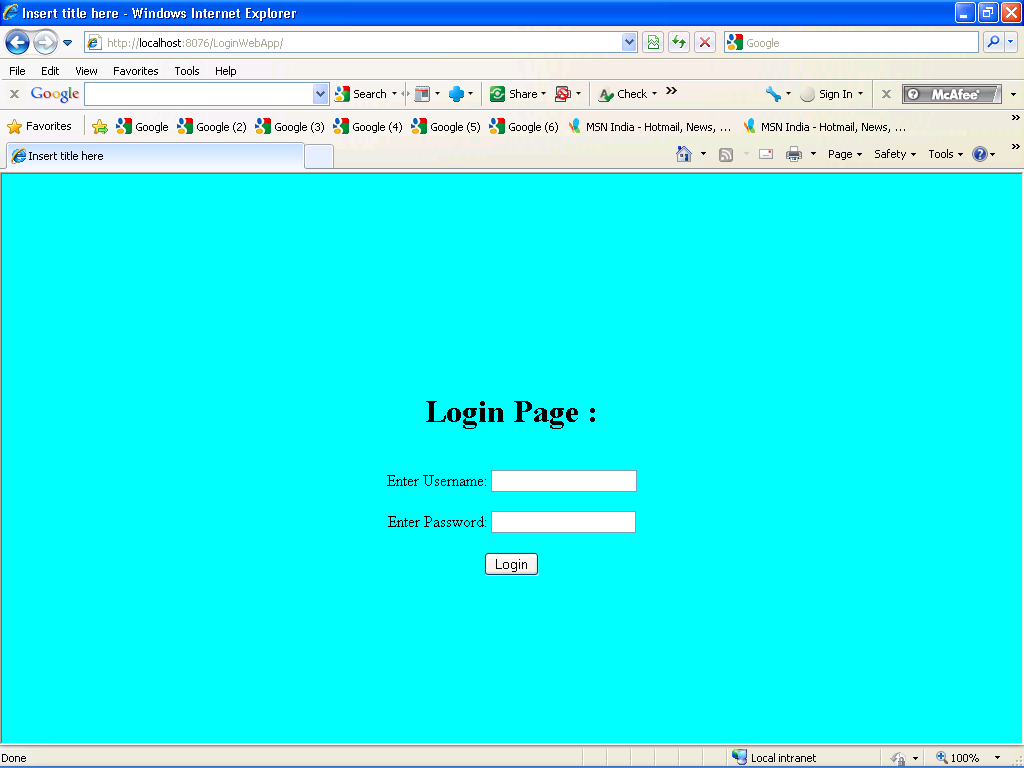


Figure 1: Login.jsp

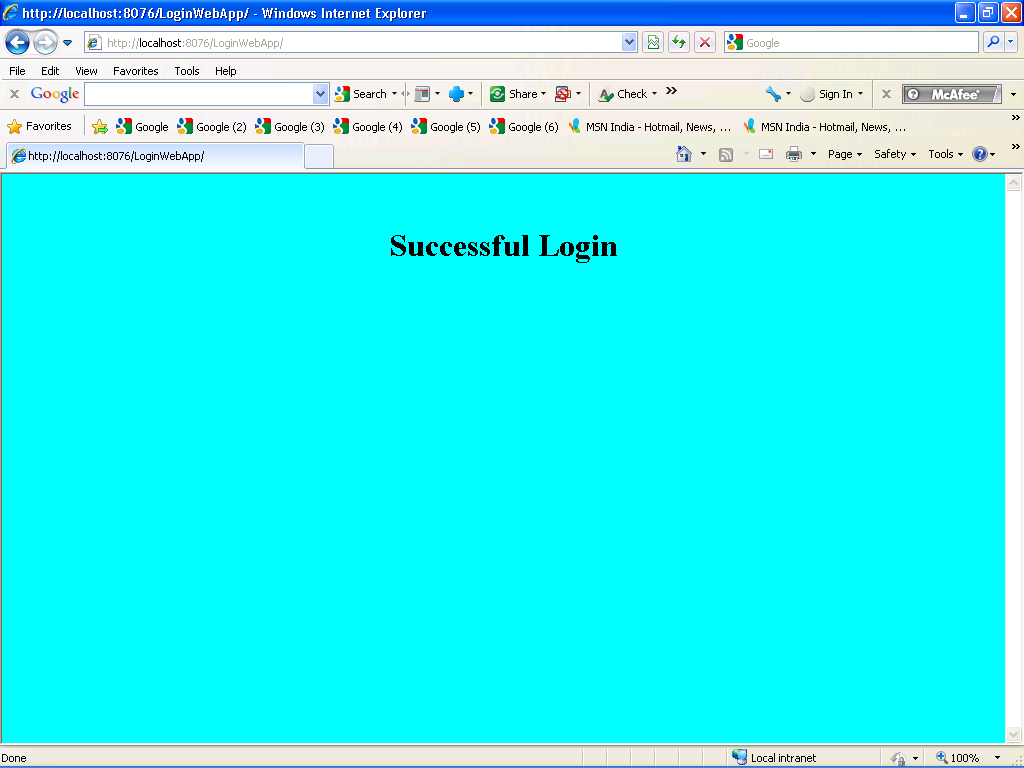


Figure 2: Success.jsp

## 1.3 : JSP Directive

**Problem:**

Modify the assignment 5.1 from Servlet Lab book. Covert User\_Info page from html to jsp and convert InfoView and ErrorView from servlet to jsp. Give the name of views as Bill\_Info.jsp and Error.jsp. Include a common Header page to all these pages with the help of include directive.

In the **Bill\_Info.jsp** page set **errorPage** attribute as **“Error.jsp”** and in Error.jsp page set isErrorPage attribute as “true”.

In the Bill\_Info.jsp page write a scriplet code to check the consumer number.

If consumer number is null then throw an **EBillException** from scriptlet code with a user defined exception message..

1. JSP Actions

|  |  |
| --- | --- |
| **Goals** | * Using include action to perform a dynamic include * Integrating JSP and Servlets using forward action * Developing Java Beans and interacting with it from a JSP page using bean related actions |
| **Time** | 3 hours |
| **Pre-requisites** | JSP Action tags and Java Beans Concept |

## 2.1: Dynamic Include action

Develop a **header.jsp** page which contains the header section as shown in the figure 3. The contents of the header section are not static. Develop a **home.jsp** page which includes the **header.jsp** page dynamically. The contents of the header section in **header.jsp** should be provided by **home.jsp** page.

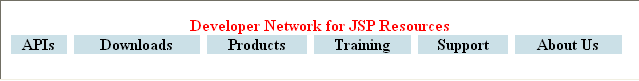


Figure 3: Header.jsp page

Note:

The contents of the **header** should not be static. Instead retrieve them from the **request** parameters. The **Home.jsp** page can include the **Header.jsp** page dynamically by using **jsp:include** action and can pass content required by the **header section** of **Header.jsp** page, as well.

Develop the Home.jsp page that includes the Header.jsp page shown above and also passes the content for menu\_item1 to menu\_item6 as request parameters to build the header dynamically

Header Page

Include

Home Page

Figure 4

## 2.2: Integrating Servlet, JSP, and Java Bean (Implementing JSP Model 2 – Model-View-Controller pattern)

Develop an online **E Bill** application that allows the user to accept readings for a consumer and calculate bill amount and persist the details. The work flow of the application is shown in the figure 5 given below:

Note :

1. This is stretched assignment of Servlet assignment 5.1.
2. Use the pages designed in Lab 1.3

EBillException

Error.jsp

(View)

EBillController

(Controller)

Index.html

(View)

Database

DAO Layer

Service Layer

Show\_ConsumerList.jsp (View)

BillDTO

ConsumerDTO

Show\_Bills.jsp (View)

Search\_Consumer.jsp (View)

Bill\_Info.jsp

(View)

Figure 5: Implementing MVC pattern

Solution:

Follow the **MVC (Model View Controller)** approach wherein a controller servlet will intercept all the requests and select the appropriate model to process the request. After the processing is complete, the controller can forward the request to the appropriate JSP (view) for presentation.

**Spec1:**Create the following database tables in Oracle

Table Script

CREATE TABLE Consumers(

consumer\_num NUMBER(6) PRIMARY KEY,

consumer\_name VARCHAR2(20) NOT NULL,

address VARCHAR2(30)

);

INSERT INTO Consumers VALUES(100001,'Sumeet','Shivaji Nagar, Pune');

INSERT INTO Consumers VALUES(100002,'Meenal','M G Colony Panvel, Mumbai');

INSERT INTO Consumers VALUES(100003,'Neeraj','Whitefield, Bangalore');

INSERT INTO Consumers VALUES(100004,'Arul','Karapakkam, Chennai');

CREATE TABLE BillDetails(

bill\_num NUMBER(6) PRIMARY KEY,

consumer\_num NUMBER(6) REFERENCES Consumers(consumer\_num),

cur\_reading NUMBER(5,2),

unitConsumed NUMBER(5,2),

netAmount NUMBER(5,2),

bill\_date DATE DEFAULT SYSDATE);

CREATE SEQUENCE seq\_bill\_num START WITH 100;

**Spec2:** Create the view **index.html**as shown in following figure 6.

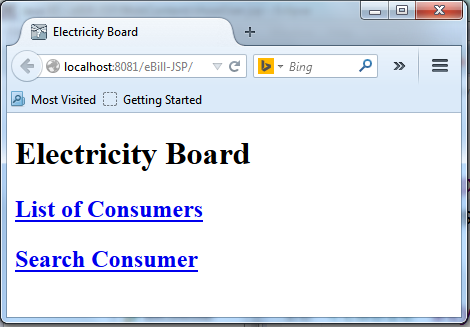


Figure 6:Index.html

**Spec3**: Develop the controller **EBillController**that receive the request received from the **Index.html**anddo the appropriate processing.

It will display page as shown in thefollowing figure 7forfirst link:

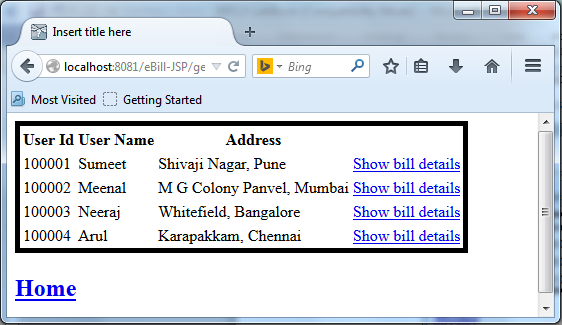
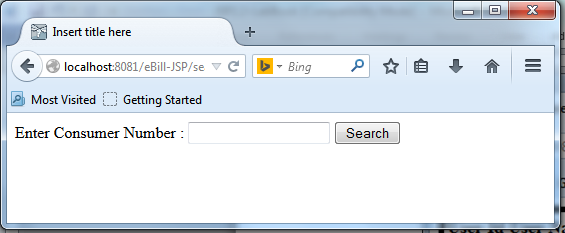


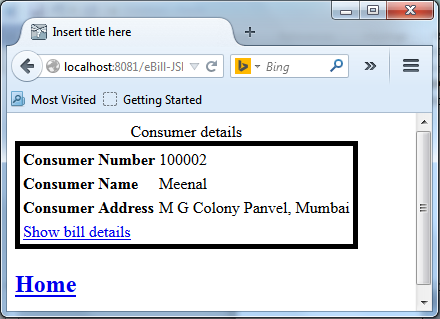
Figure 7: Show\_ConsumerList.jsp

And display page as shown in the figure 8 for second link from index page:



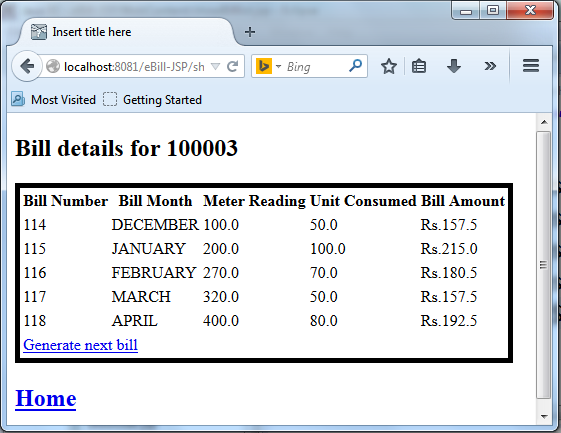
**Figure 8:**Search\_Consumer.jsp

**Spec 4:**After clicking on the Search button for a valid user id in **Search\_Consumer** pageshown in the figure 9 should be displayed in output.



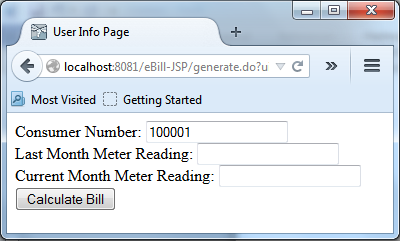
**Figure 9:Show\_Consumer.jsp**

**Spec 5:**After clicking on the **Show bill details** link it will display the bill details for that consumer as below in the figure 10. The below page should be displayed when user clicks on the **Show bill details** link in the **Show\_ConsumerList** page.



**Figure 10:Show\_Bills.jsp**

**Spec6**: After clicking on **Generate nextbill** linkfollowing form as shown in figure 11 should be displayed to accept meter readings for that consumer.



**Figure 11:User\_Info.jsp**

**Spec7**: After clicking on the Calculate BillButton ,Bill must be generated as per the Servlet assignment 5.1 . Insert the details in the database and display the bill as shown in the following figure 12:

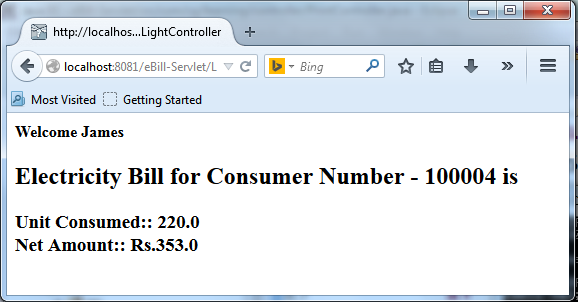


Figure 12:Bill\_Info.jsp

**Spec 8**: In case of error, user defined error message should be displayed as shown in figure 13:

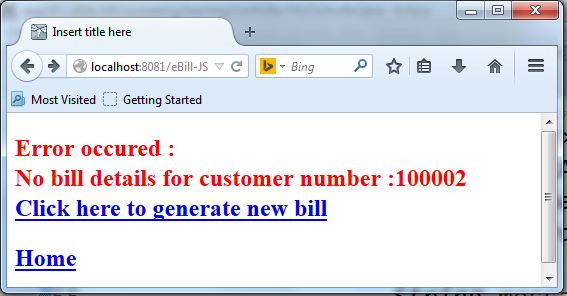


Figure 13:Error.jsp

## 2.3: JSP configuration using web.xml

Modify 2.1 Exercise with JSP configuration in web.xml.

**Spec 1:** Header.jsp created in 2.1 should be included to all JSP pages.

**Spec 2:** Scriptlet code should not be used to all JSP pages.

**Spec 3:** All JSP pages should evaluate the EL expressions.

**Spec 4**: Buffer size for all JSP pages should be 8 Kb.

1. JSP Standard Tags Libraries (JSTL)

|  |  |
| --- | --- |
| **Goals** | * Using core Tag Libraries |
| **Time** | 2hrs |
| **Pre-requisite** | XML Basics and Java |

## 3.1: Using Core tag library : <c:set> and <c:out>

Write a JSP code that uses **c:set** tag to assign “**Hello World!**” value to the hello variable and then display the contents of the variable.

Start the WildFly Server to invoke **Hello\_Core.jsp**. You must see the output as shown in figure 14:



Figure 14:Output of Hello\_Core.jsp using JSTL tags

## 3.2: Using Core tag tag library: <c:if>

Create an html page that will accept username, age, and url. Check the age, if it lies above 18, then display a personalized welcome message to the user.

**For example:** Hello <username>, since you are <age> which is above 18 yrs, you will now be redirected to the url <url>.

**hint2Hint:** use <c:if> and <c:url> and <c:redirect> tags for this.

## 3.3: Using Lambda Expression with EL

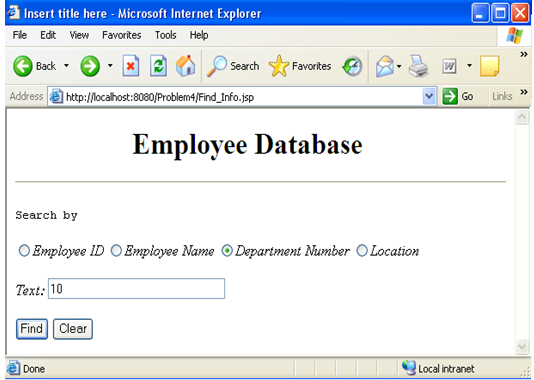
Modify 3.2 exercise with lambda expression with EL.

**3.4**Modify**Electricity Bill** application which was developed in previous lab by developing JSP pages using appropriate JSTL tags and Lambda Expression, replacing all the scripting elements.

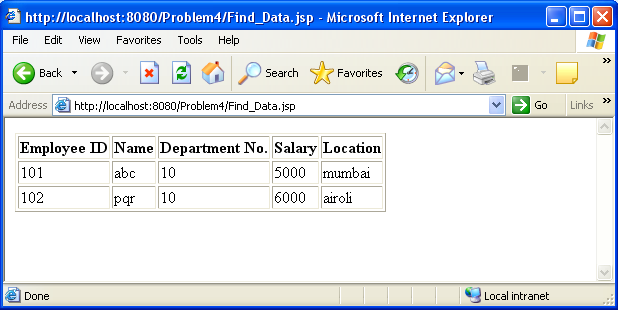
**Appendix a: Extra Assignment**

## Assignment 1: JSP Directive

Develop a **Find\_Info.jsp** page which accepts various inputs from user as shown in the figure 15 below. When user clicks the “**Find**” button, a **Find\_Data.jsp** page display the result in a tabular format as shown in the figure 16. The employee details are stored in oracle database [Refer the table script]. Use page Directive appropriately. For the wrong inputs User defined error message should be displayed on Error.jsp page. Use MVC with Layered architecture given in the figure 17.



**Figure 15:Find\_Info.jsp**



**Figure 16:Find\_Info.jsp**

The work flow of the application is shown in the figure given below:

Find\_Info.jsp

DAO Layer

Service layer

FindEmpController

Database

Find\_Data.jsp

EmpDTO

Error.jsp

**Figure 17: Architecture for application**

Table Script :

CREATE TABLE employee\_master(

Emp\_numNUBBER(4) PRIMARY KEY,

Emp\_nameVARCHAR2(25) NOT NULL,

Dept\_numNUMBER(2),

Salary NUMBER(9,2),

Location VARCHAR2(30)

);

INSERT INTO employee\_masterVALUES(1001,'Smith',10, 5000,'Pune');

INSERT INTO employee\_masterVALUES(1002,'Venkat',30, 7800,'Chennai');

INSERT INTO employee\_masterVALUES(1003,'Meerah',20, 9000,'Surat');

INSERT INTO employee\_masterVALUES(1004,'Saniya',30, 8000,'Noida');

INSERT INTO employee\_masterVALUES(1005,'Shivam',30, 8500,'Noida');

INSERT INTO employee\_masterVALUES(1006,'Kiran',10, 6000,'Pune');

COMMIT;