

# Java Server Pages v2.2

## Lab Book

## Document Revision History

Date	Revision No.	Author	Summary of Changes
Nov-2009	2.0	Pradnya Jagtap, 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
Table of Contents .....	3
Getting Started .....	4
Overview .....	4
Setup Checklist for JSP .....	4
Lab 1. JSP Basics .....	5
1.1: Developing Simple JSP Page .....	5
1.2: JSP Scripting Elements and Method Declarations .....	5
1.3 : JSP Directive .....	6
Lab 2. JSP Actions .....	7
2.1: Dynamic Include action .....	7
2.2: Integrating Servlet, JSP, and Java Bean (Implementing JSP Model 2 – Model-View-Controller pattern) .....	8
2.3: JSP configuration using web.xml .....	14
Lab 3. JSP Standard Tags Libraries (JSTL) .....	15
3.1: Using Core tag library : <c:set> and <c:out> .....	15
3.2: Using Core tag tag library: <c:if> .....	15
3.3: Using Lambda Expression with EL .....	16
Appendix a: .....	16

## 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.

## Lab 1. JSP Basics

<b>Goals</b>	<ul style="list-style-type: none"> <li>Create simple JSP programs.</li> </ul>
<b>Time</b>	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 declared in the JSP page.

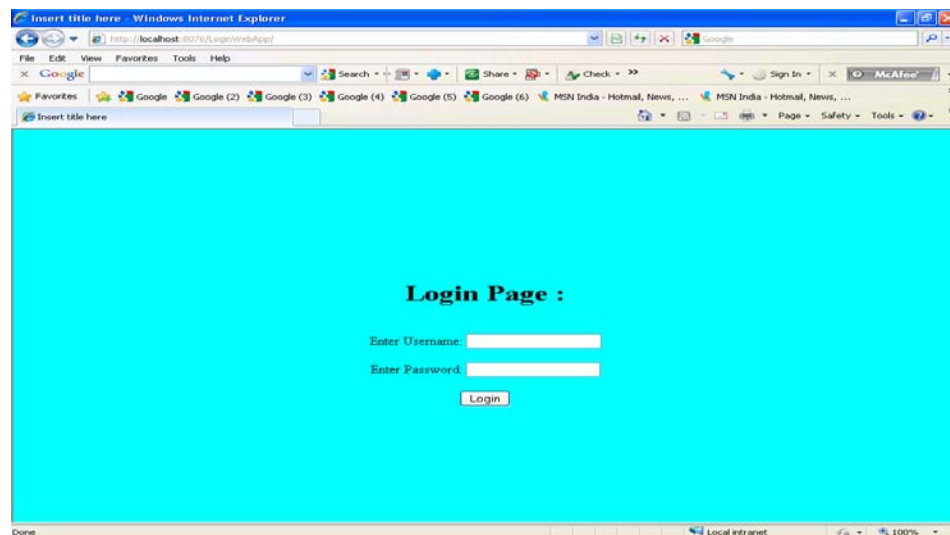


Figure 1: Login.jsp

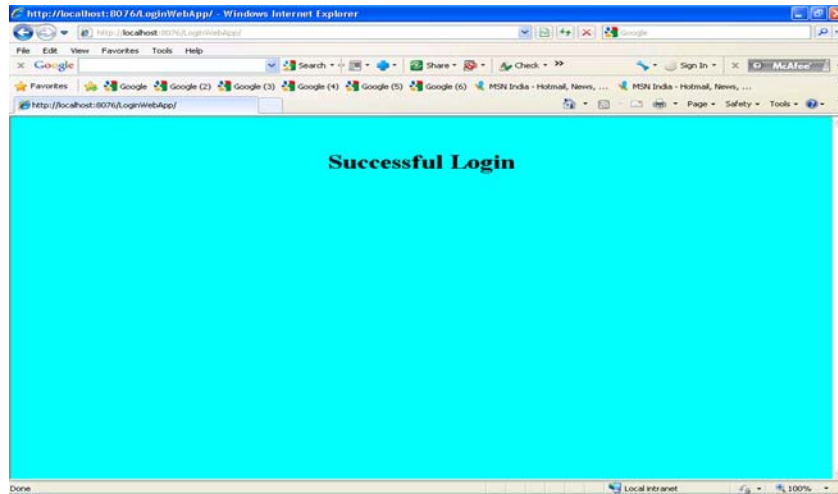


Figure 2: Success.jsp

### 1.3 : JSP Directive

#### Problem:

Modify the assignment 5.1 from Servlet Lab book. Convert 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 scriptlet code to check the consumer number.

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

## Lab 2. JSP Actions

<b>Goals</b>	<ul style="list-style-type: none"> <li>Using include action to perform a dynamic include</li> <li>Integrating JSP and Servlets using forward action</li> <li>Developing Java Beans and interacting with it from a JSP page using bean related actions</li> </ul>
<b>Time</b>	3 hours
<b>Pre-requisites</b>	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.

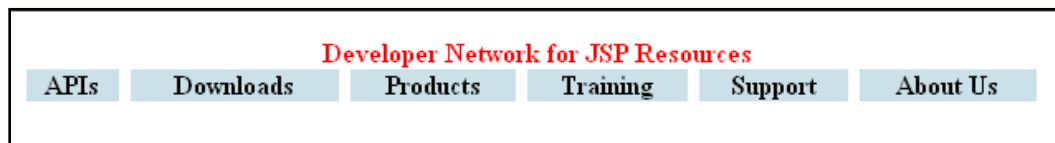


Figure 3: Header.jsp page

#### 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

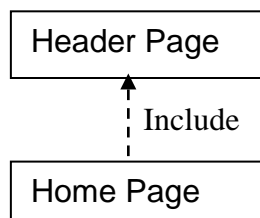


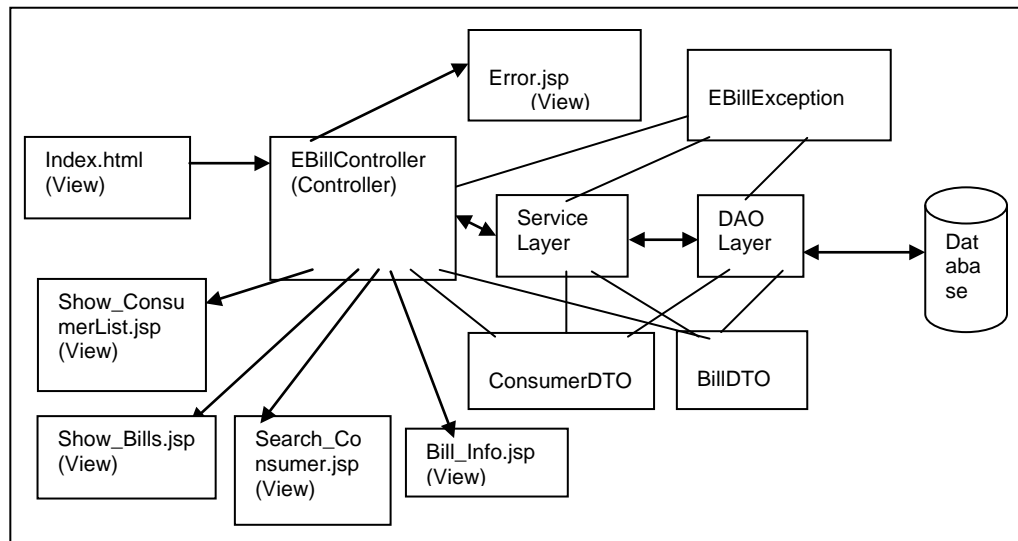
Figure 4

## 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



**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');

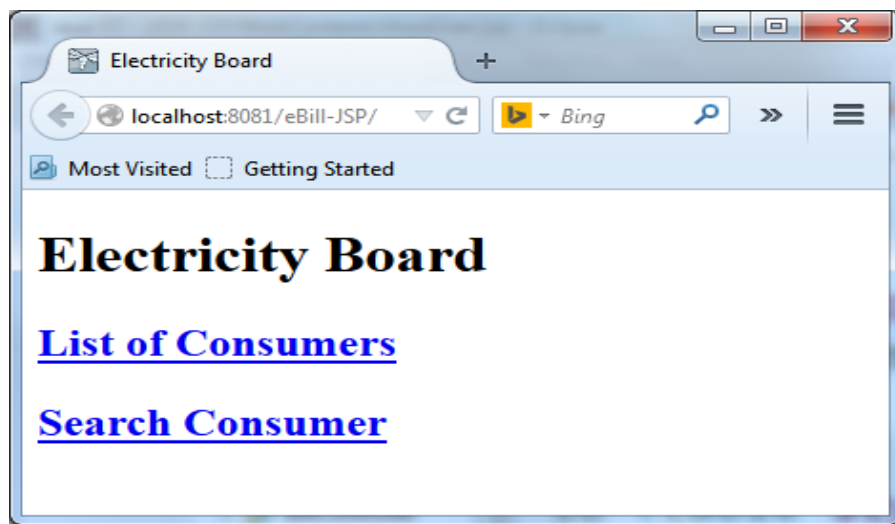
INSERT INTO Consumers VALUES(100003,'Neeraj','Whitefield, Bangalore');
INSERT INTO Consumers VALUES(100004,'Arul','Karapakkam, Chennai');
  
```



```
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;
```

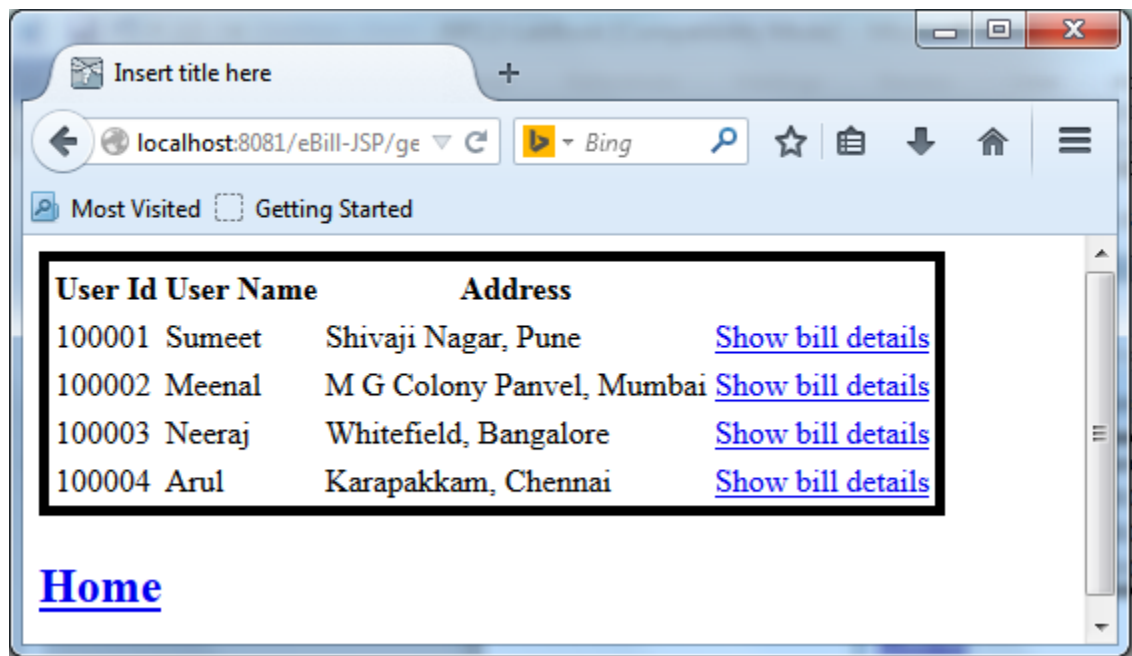
**Spec2:** Create the view **index.html** as shown in following figure 6.



**Figure 6:Index.html**

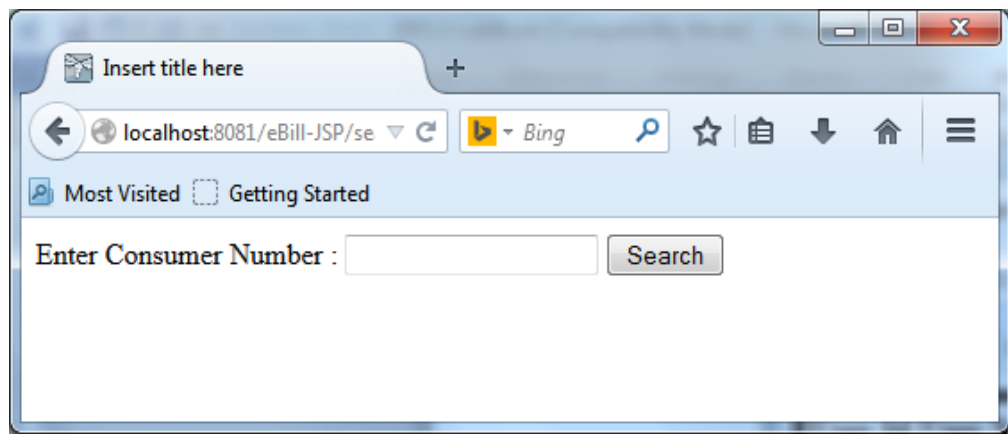
**Spec3:** Develop the controller **EBillController** that receive the request received from the **Index.html** and do the appropriate processing.

It will display page as shown in the following figure 7 for first 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.

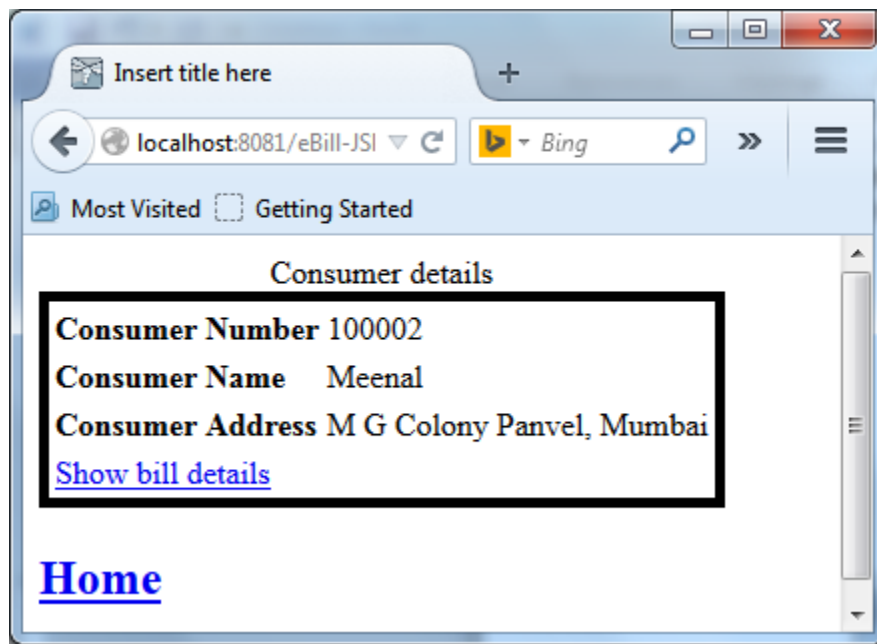
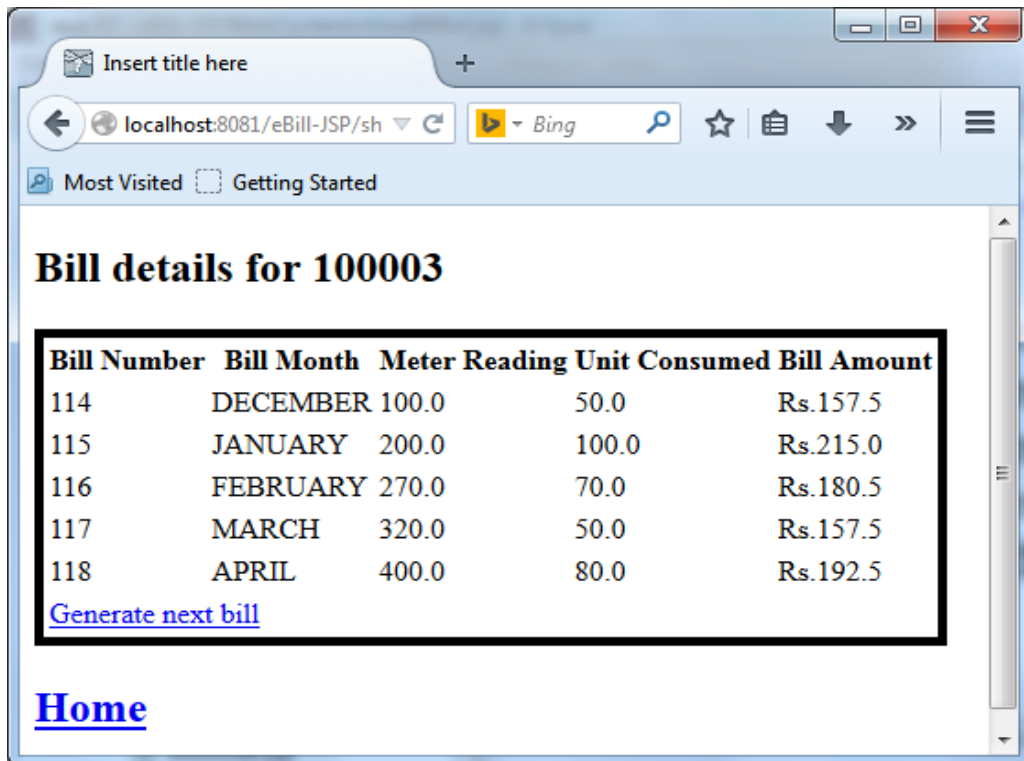


Figure 9:Show\_Consumer.jsp

**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.



Insert title here

localhost:8081/eBill-JSP/sh

Most Visited Getting Started

### Bill details for 100003

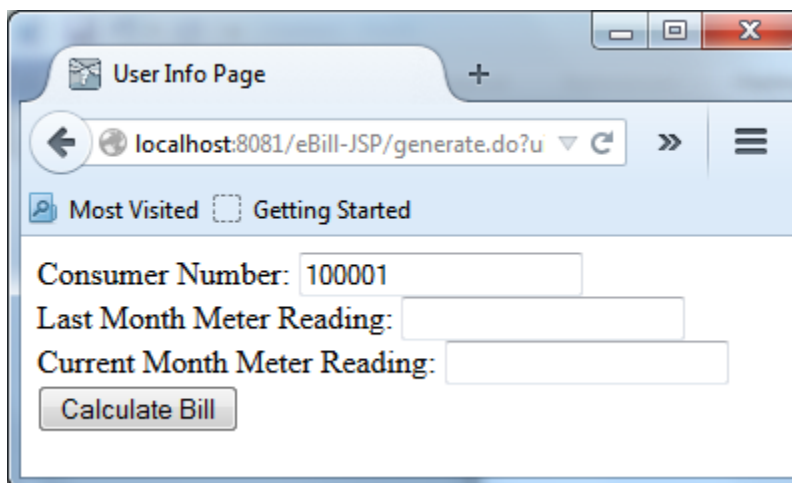
Bill Number	Bill Month	Meter Reading	Unit Consumed	Bill Amount
114	DECEMBER	100.0	50.0	Rs.157.5
115	JANUARY	200.0	100.0	Rs.215.0
116	FEBRUARY	270.0	70.0	Rs.180.5
117	MARCH	320.0	50.0	Rs.157.5
118	APRIL	400.0	80.0	Rs.192.5

[Generate next bill](#)

[Home](#)

Figure 10:Show\_Bills.jsp

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



User Info Page

localhost:8081/eBill-JSP/generate.do?u

Most Visited Getting Started

Consumer Number: 100001

Last Month Meter Reading:

Current Month Meter Reading:

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:

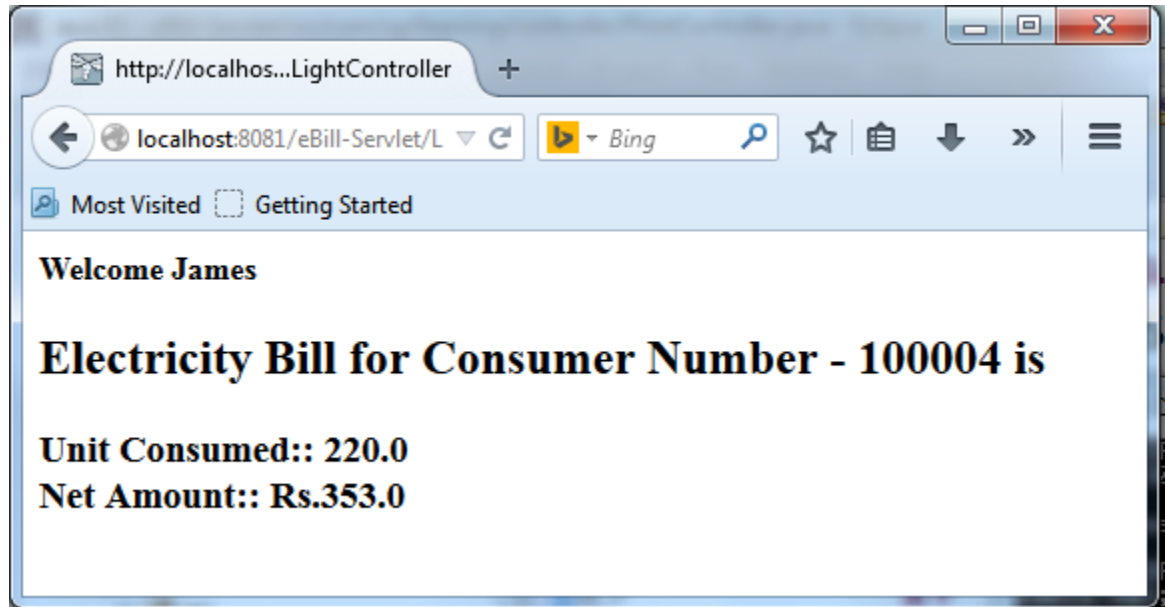


Figure 12:Bill\_Info.jsp

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

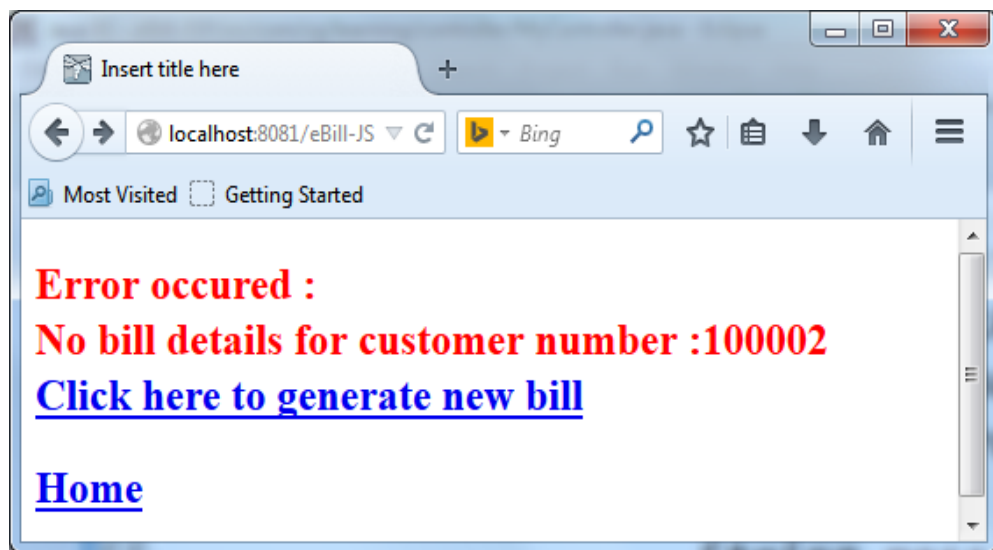


Figure 13:Error.jsp

### 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.

## Lab 3. JSP Standard Tags Libraries (JSTL)

<b>Goals</b>	<ul style="list-style-type: none"> <li>Using core Tag Libraries</li> </ul>
<b>Time</b>	2hrs
<b>Pre-requisite</b>	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:

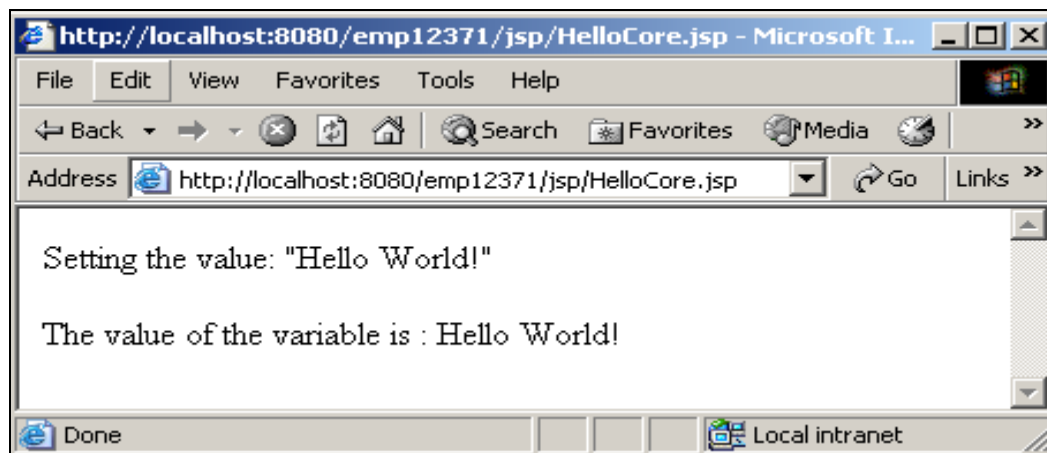


Figure 14:Output of Hello\_Core.jsp using JSTL tags

### 3.2: Using Core tag 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>.



**Hint:** 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.

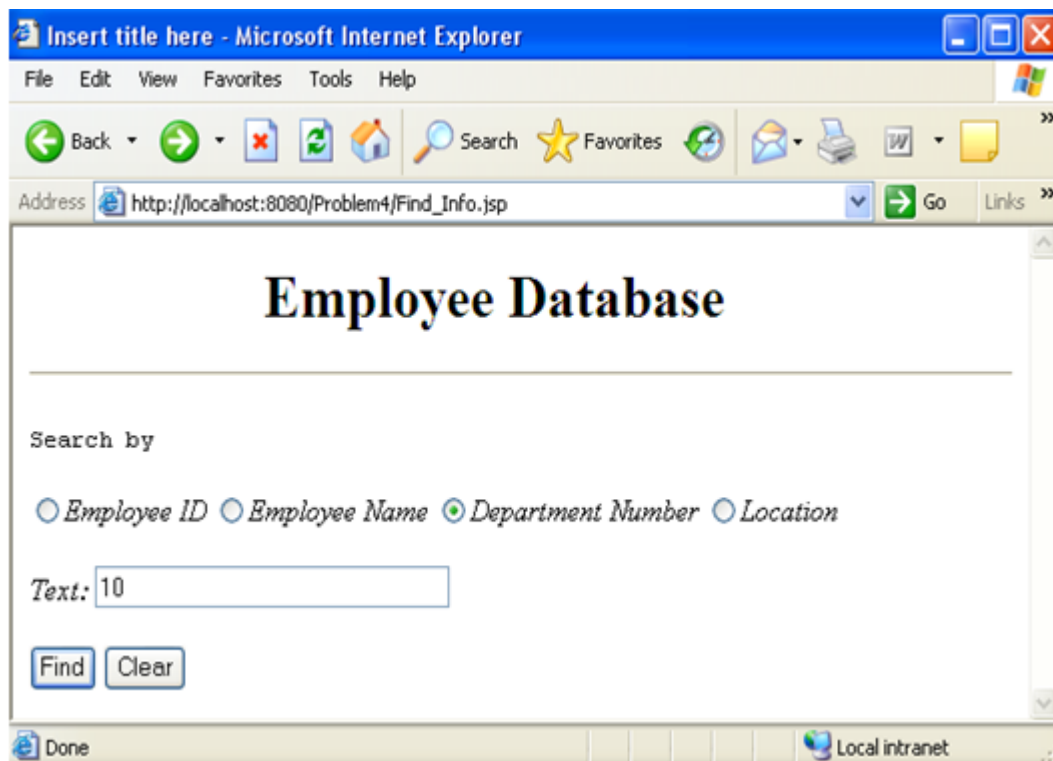
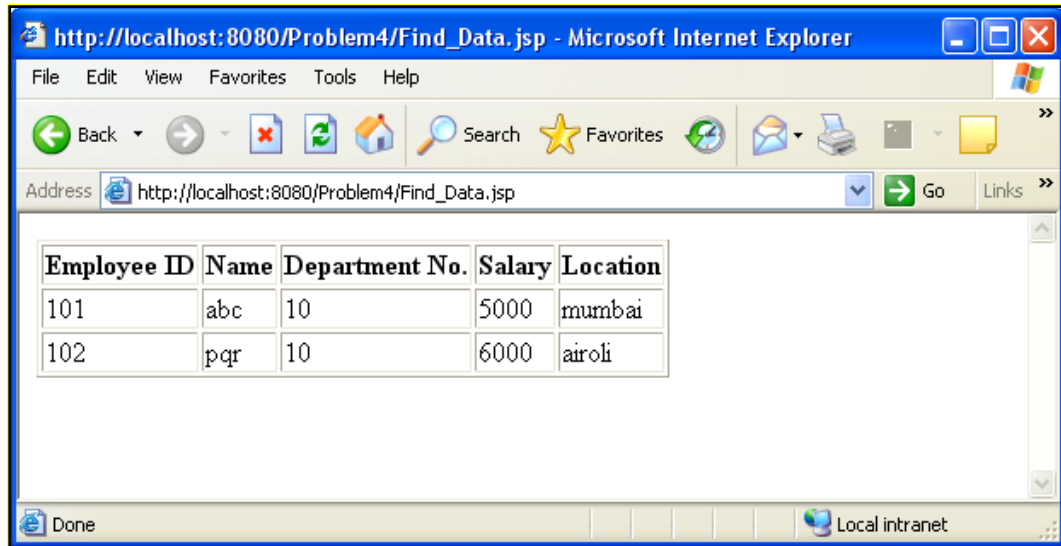


Figure 15:Find\_Info.jsp





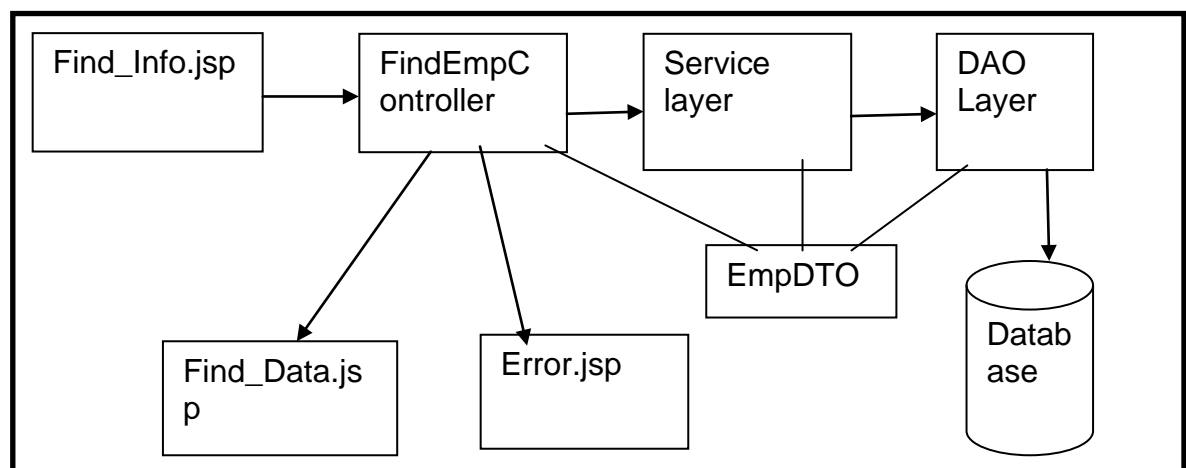
http://localhost:8080/Problem4/Find\_Data.jsp - Microsoft Internet Explorer

Employee ID	Name	Department No.	Salary	Location
101	abc	10	5000	mumbai
102	pqr	10	6000	airoli

Done Local intranet

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

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



**Figure 17: Architecture for application**

### Table Script :

```

CREATE TABLE employee_master(
    Emp_numNUMBER(4) PRIMARY KEY,
    Emp_nameVARCHAR2(25) NOT NULL,
    Dept_numNUMBER(2),
    Salary NUMBER(9,2),

```

Location VARCHAR2(30)

);

```
INSERT INTO employee_master VALUES(1001,'Smith',10, 5000,'Pune');
INSERT INTO employee_master VALUES(1002,'Venkat',30, 7800,'Chennai');
INSERT INTO employee_master VALUES(1003,'Meerah',20, 9000,'Surat');
INSERT INTO employee_master VALUES(1004,'Saniya',30, 8000,'Noida');
INSERT INTO employee_master VALUES(1005,'Shivam',30, 8500,'Noida');
INSERT INTO employee_master VALUES(1006,'Kiran',10, 6000,'Pune');
COMMIT;
```