



## THỰC HÀNH CHUYÊN ĐỀ CƠ SỞ 2 (JAVA NÂNG CAO)

TS. Võ Phương Bình – Email: [binhvp@dlu.edu.vn](mailto:binhvp@dlu.edu.vn)

Khoa CNTT – Trường Đại học Đà Lạt

Website: <http://sites.google.com/view/vophuongbinh>



## Lab 2: Java Server Page (JSP)

Program Code	Chuyên đề cơ sở 2 (Java nâng cao)
Issue/Revision	8/2018
Effective date	3/2011

**Contents**

1. Lab Guide 01: Developing Web Applications-An Overview.....	3
2. Lab Guide 02: Using Java Server Pages (JSP) Tags .....	11
3. Lab Guide 03: Using JSP Scriptlet.....	20
4. Lab Guide 04: Using JavaBeans in Java Server Pages.....	27

## 1. Lab Guide 01: Developing Web Applications-An Overview

1. Write a program to display a “Hello World” message in the Web browser. In addition, display the host name and Session Id. Write JSP code using HTML code.

### Solution:

```
<html>

  <head>

    <title>Welcome to our Website</title>

  </head>

  <body>

    <marquee><font size="3" color="#FF0033">Hello World!!
</font></marquee>

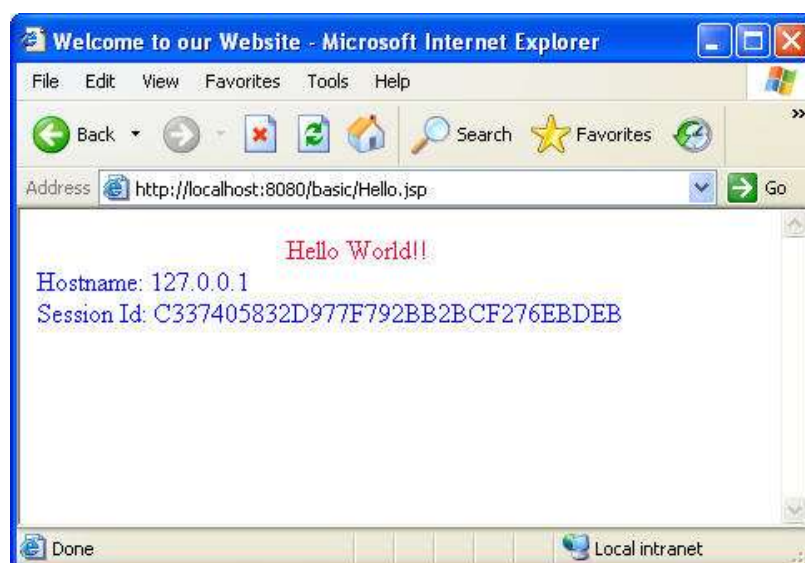
    <font color = "#0000FF"> Hostname: <%= request.getRemoteHost()
%><br>

    <font color = "#0000FF"> Session Id: <%= session.getId() %>

  </body>

</html>
```

The output of the program is as shown in Figure 2.1.



**Figure 2.1: Output of Hello.jsp**

2. Write a program using JSP to calculate the sum of numbers from 1 to 25.

**Solution:**

```
<html>

  <head>

    <title>JSP Example</title>

  </head>

  <body>

    <h1>Displaying sum</h1>

    <%

      int sum=0;

      for(int num=1;num<=25;num++)

      {

        sum=sum+num;

      }

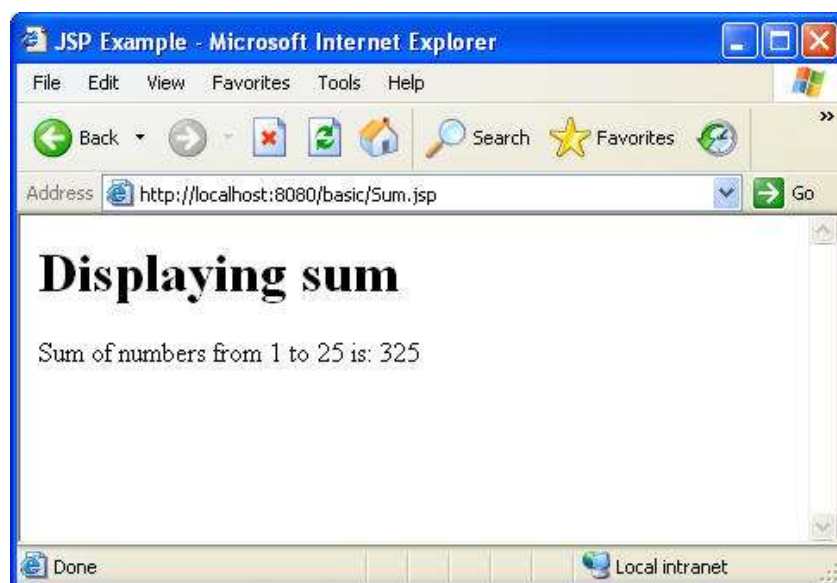
      out.println("Sum of numbers from 1 to 25 is : " + sum);

    %>

  </body>

</html>
```

The output of the program is as shown in Figure 2.2.



**Figure 2.2: Output of Sum.jsp**

3. Write a program to display the current date and the session Id.

**Solution:**

```
<html>

  <head>

    <title>JSP Example</title>

  </head>

  <body>

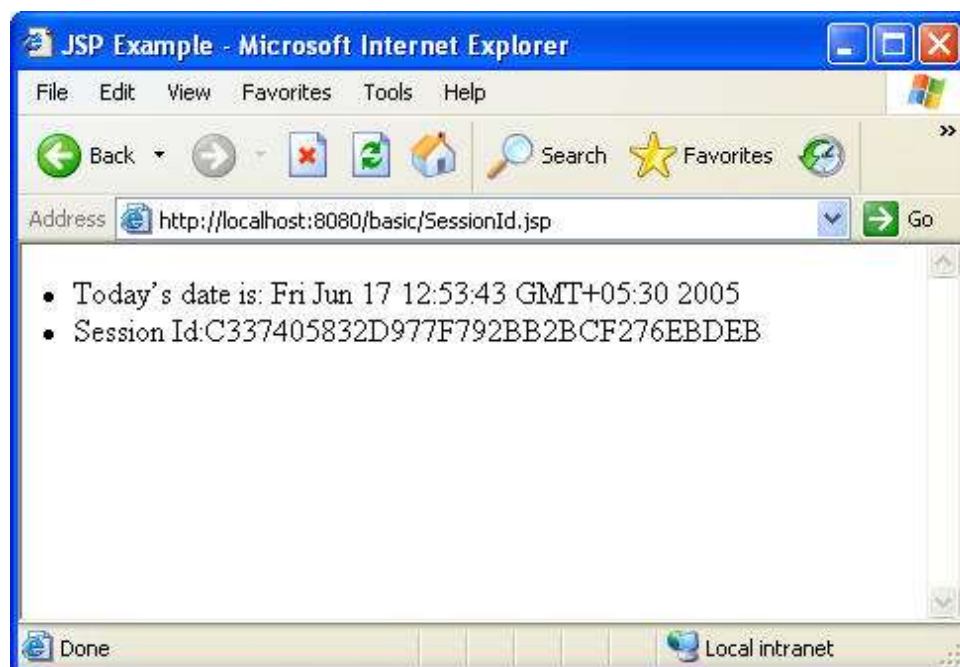
    <LI>Today's date is: <%= new java.util.Date() %>

    <LI>Session Id:<%= session.getId() %>

  </body>

</html>
```

The output of the program is as shown in Figure 2.3.



**Figure 2.3: Output of SessionId.jsp**

---

## Do It Yourself

---

1. Write a program welcoming the user to Online Banking Website. Display a text box and a “Submit” button to the user. Label the text box as “Enter name”. After the user enters the name, a “Welcome” message with the name should be displayed. If the user name is null, display a “Welcome” message to the user.

### Solution:

```
<html>

  <head>

    <title>Welcome to Online Banking</title>

  </head>

  <body>

    <% String user=request.getParameter("user"); %>

    <H3>Welcome <%= (user==null) ? "to Online Banking" : user + "
to Online  Banking" %>!/H3>

    <b>Enter name:</b>

    <form method=get>

      <input type="text" name="user" size=15>

      <input type="submit" value="Submit name">

    </form>

  </body>

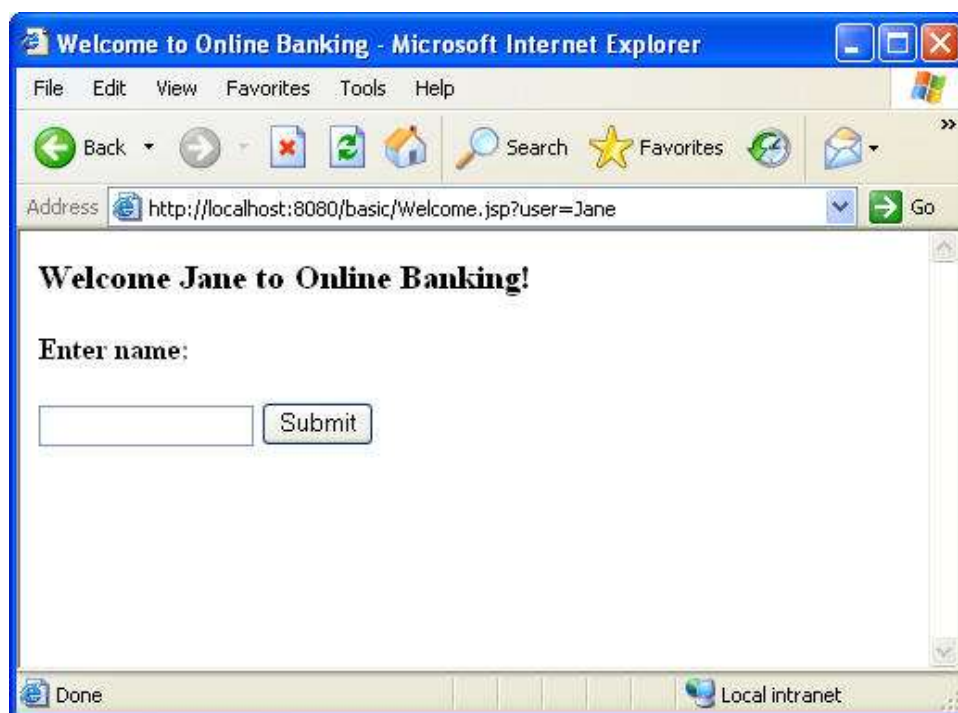
</html>
```

The output of the program is as shown in Figure 2.4 when the user name has not been entered, that is, user name = null.



**Figure 2.4: Output of Welcome.jsp**

When the user enters name in the text box and clicks on Submit button, the output appears as shown in Figure 2.5.



**Figure 2.5: Output of Welcome.jsp**

2. Write a program using JSP to display the message “Hello” ten times. The message should be displayed in blue. In addition, display the count with each “Hello” message.

**Solution:**

```
<html>

  <head>

    <title>Welcome to our Website</title>

  </head>

  <body>

    <%
      for(int num=0;num<=10;num++)
      {
        out.println("Hello" + num + "<br>");

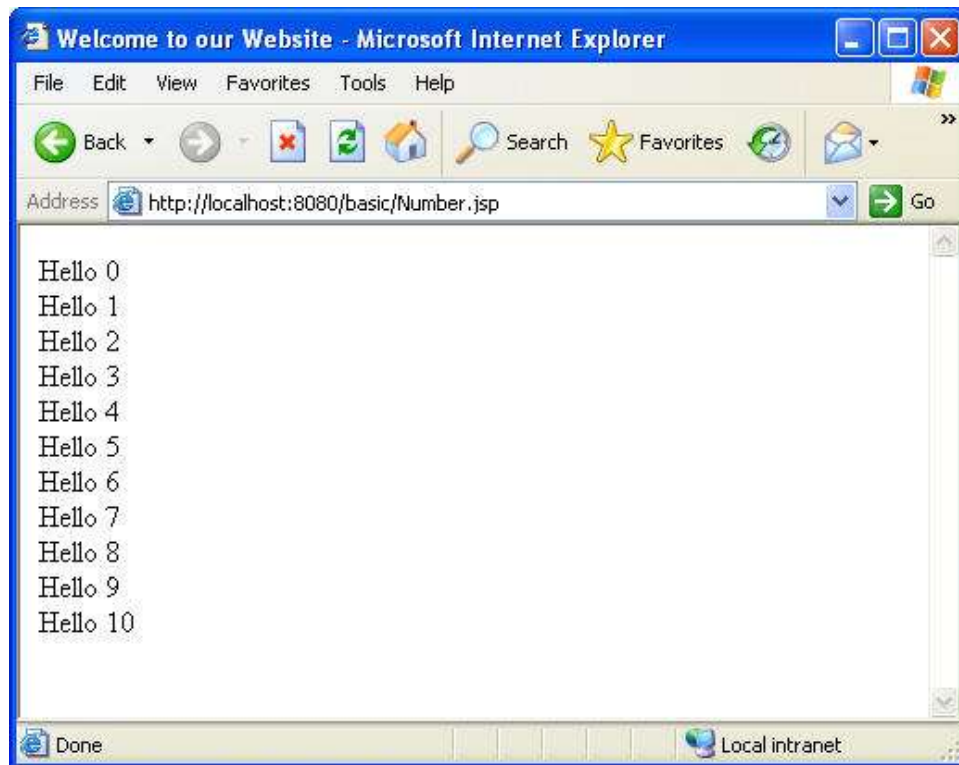
      }
    %>

  </body>

</html>
```

The output of the program is as shown in Figure 2.6.





**Figure 2.6: Output of Number.jsp**

3. Write a program using JSP to display the hostname and the path of the JSP file.

**Solution:**

```
<html>

  <head>

    <title>JSP Example</title>

  </head>

  <body>

    <p>

      <LI>The path of this JSP file is:
<%=request.getRequestURI()%>

      <LI>The Hostname is: <%= request.getRemoteHost() %><br>

    </p>

  </body>

</html>
```

The output of the program is as shown in Figure 2.7.

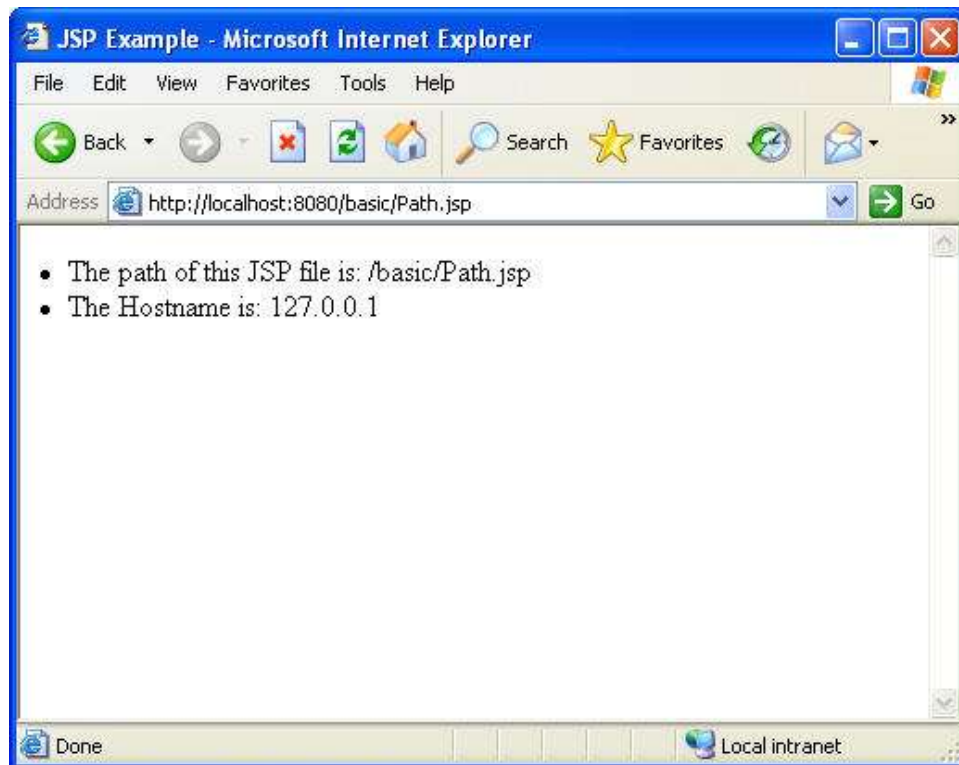


Figure 2.7: Output of Path.jsp

## 2. Lab Guide 02: Using Java Server Pages (JSP) Tags

1. Write a program using the `request.getParameter()` method to enter the Name and Password of a user and display the output on another JSP page.

### Solution:

The files used to run the application are:

1. User.jsp
2. UserDisplay.jsp

```
<html>

  <head>

    <title>Example of Implicit Objects</title>

  </head>

  <body>

    <h1>User submission form</h1>

    <form action="UserDisplay.jsp" method="post">

      Enter User Name:

      <input type="text" name="uname">

      <br>

      <br>

      Enter Password:

      <input type="password" name="pname">

      <br>

      <br>

      <input type="submit" name="submit" value="Submit">

    </form>

  </body>

</html>
```

Save the code as User.jsp in the C:\Tomcat 5.5\Webapps\basic directory.

```
<html>

  <head>

    <title>Example of Implicit objects</title>

  </head>

  <body>

    <font face=Times New Roman size=3>

      Thank you for your submission. Your information has been
      successfully added to the database:

      <br>

      <br>

      <%

        String sUName = request.getParameter("uname");
        String sPName = request.getParameter("pname");

        %>

        User Name:<%=sUName%><br>

        Password:<%=sPName%><br>

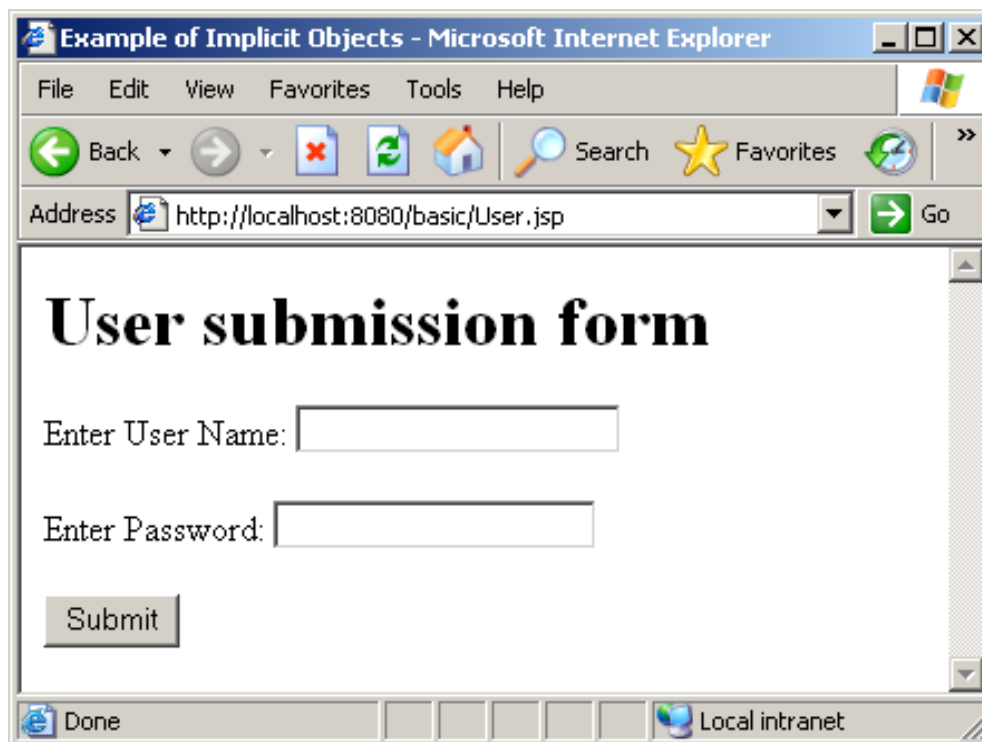
      </font>

    </body>

  </html>
```

Save the code as UserDisplay.jsp in the C:\Tomcat 5.5\Webapps\basic directory.

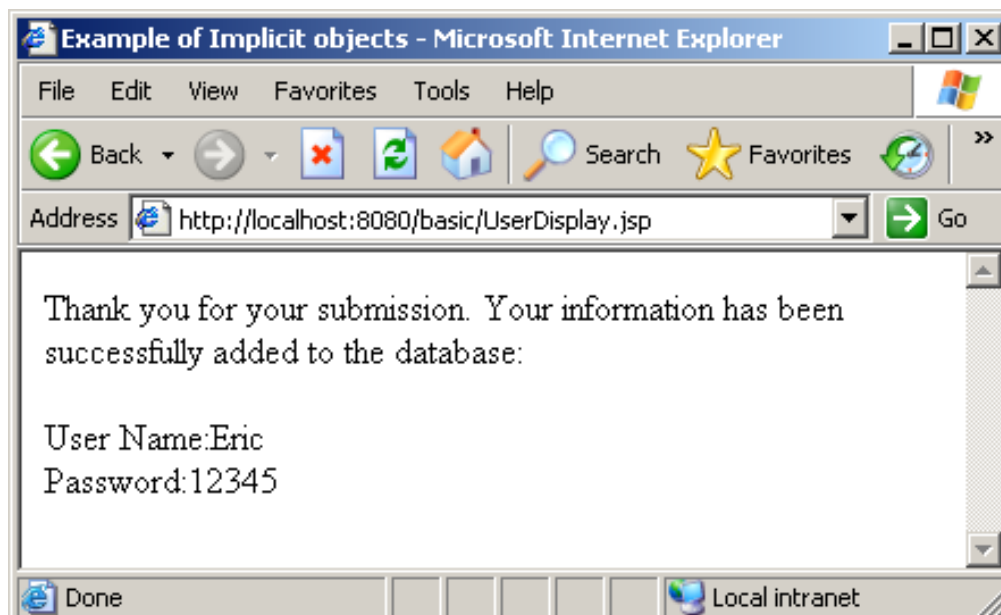
The output of the program is as shown in Figure 4.1.



**Figure 4.1: Output of User.jsp**

The user enters the information and clicks the Submit button. The control is transferred to the UserDisplay.jsp page.

The output of the program is as shown in Figure 4.2.



**Figure 4.2: Output of UserDisplay.jsp**

2. Write a program that uses the `response.setHeader()` method to display the date. In addition, set the response header attribute and utilize the JSP expression tag.

**Solution:**

The file required to run the application is **Date.jsp**. The file is to be saved in C:\Tomcat 5.5\Webapps\basic directory.

```
//Date.jsp

<html>

  <head>

    <title>Example of JSP Implicit Object</title>

    <%@ page import="java.util.Date" %>

  </head>

  <body bgcolor=#ffffff>

    <font color="Black">

      <h2> This example gives the Current Date </h2>

      <h3>

        <% response.setHeader("Refresh", "6"); %>

        Current date: <%= new Date() %>.

      </h3>

    </body>

</html>
```

The output of the program is as shown in Figure 4.3.

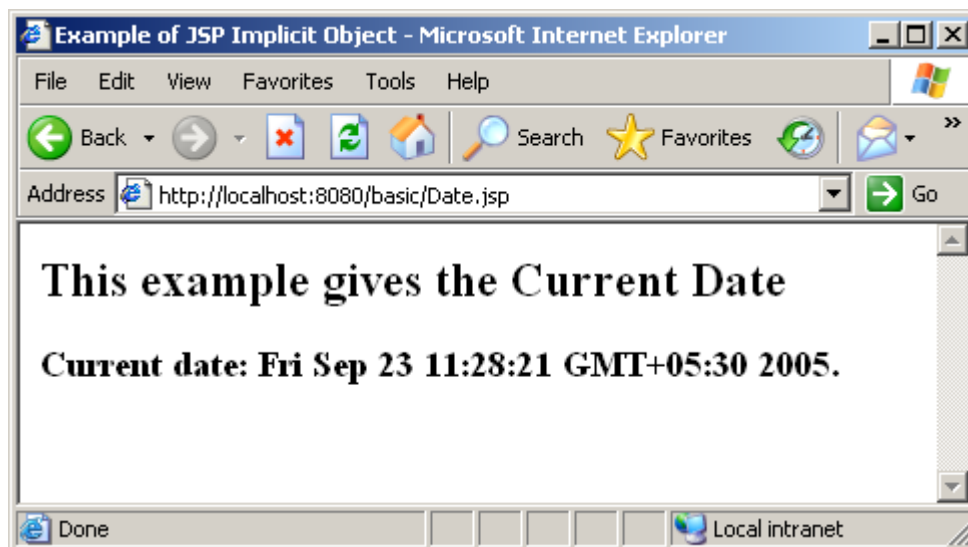


Figure 4.3: Output of Date.jsp

### Do It Yourself

1. Write a program to retrieve the name and IP address of a remote computer.

**Hint:** Use the request implicit object to retrieve the values.

### Solution:

The file used to run the application is **IPAdd.jsp**. The file is to be saved in C:\Tomcat 5.5\Webapps\basic directory.

```
<html>

  <head>

    <title>System Information</title>

  </head>

  <body>

    <h3>Details of Remote Computer:</h3>
```

```
<b>Computer Name:</b>:  
  
<%=request.getRemoteHost() %>  
  
  
  
  
IP Address:</b>:  
  
<%=request.getRemoteAddr() %>  
  
  
</body>  
</html>
```

The output of the program is as shown in Figure 4.4.

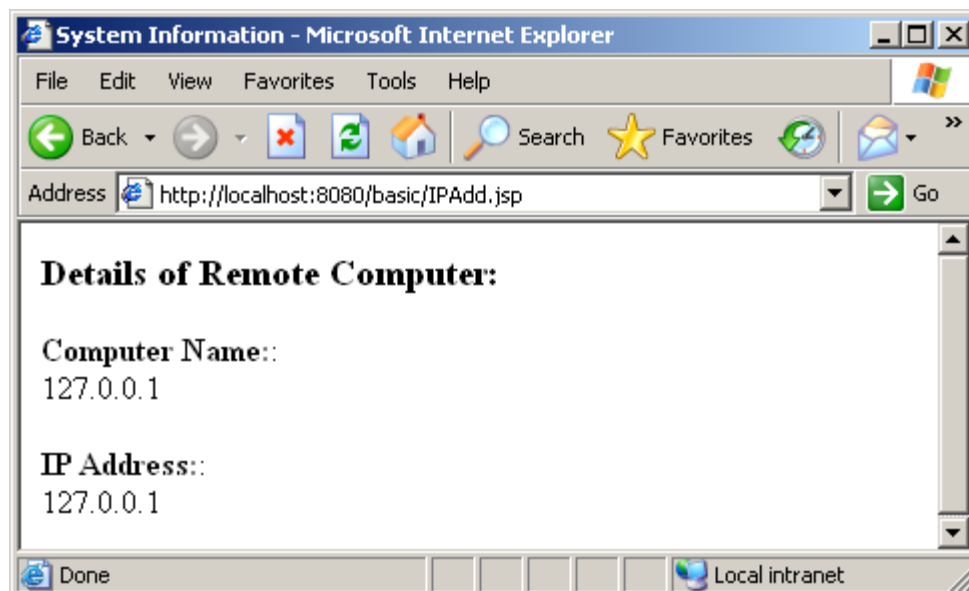


Figure 4.4: Output of IPAdd.jsp



2. Write a program that passes three parameters to a JSP page. In addition, override a request parameter when a JSP page is called. Specify the three parameters as param1, param2 and param3.

**Solution:**

The files used to run the application are:

1. ParamId.jsp
2. ParamObj.jsp

```
<html>

    <head></head>

    <body>

        <jsp:include page="ParamObj.jsp" >

            <br>

            <jsp:param name="param1" value="val1" />

            <br>

            <jsp:param name="param2" value="val2" />

            <br>

            <jsp:param name="param3" value="val3" />

        </jsp:include>

        <br>

        <b>Calling page:<b>

        <br>

        Name1: <%= request.getParameter("param1") %>

        <br>

        Name2: <%= request.getParameter("param2") %>

        <br>

        Name3: <%= request.getParameter("param3") %>

        <br>
```

```
    </body>
</html>
```

Save the code as ParamId.jsp in the C:\Tomcat 5.5\Webapps\basic directory.

```
<html>

  <body>

    <h2>Called page: </h2>

    <b>Names:</b>

    <br>

    Name1: <%= request.getParameter("param1") %>

    <br>

    Name2: <%= request.getParameter("param2") %>

    <br>

    Name3: <%= request.getParameter("param3") %>

    <br>

  </body>

</html>
```

Save the code as ParamObj.jsp in the C:\Tomcat 5.5\Webapps\basic directory. The <http://localhost:8080/basic/ParamObj.jsp?param1= Alice&param2=Bob&param3=Cathy> path is entered into the Web browser. This passes the three parameters Eric, Alice, and Bob to the ParamId.jsp page.

The output of the program is as shown in Figure 4.5.

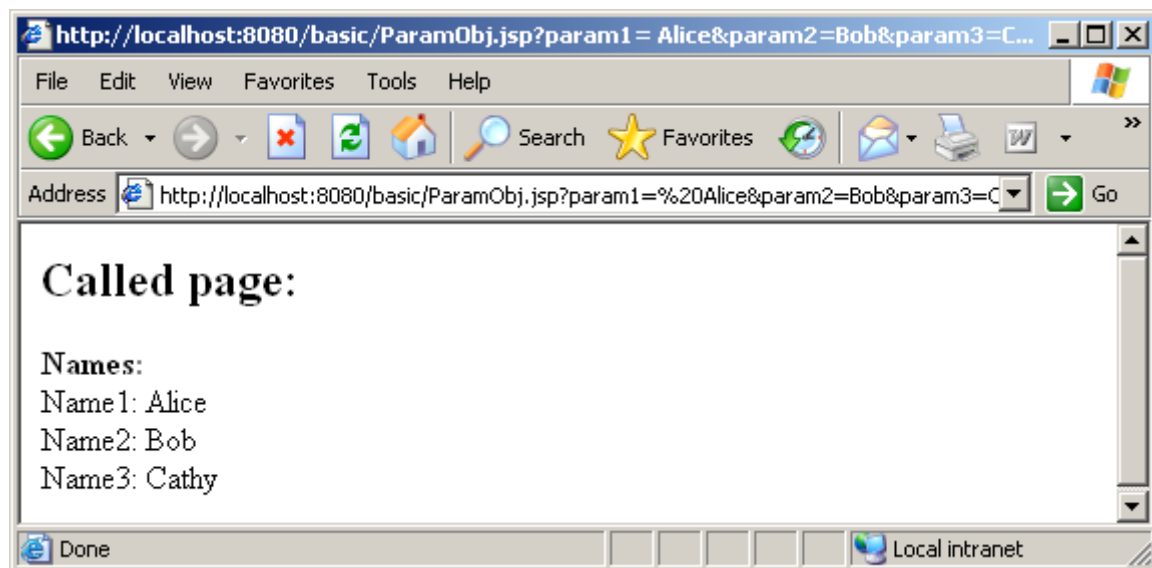


Figure 4.5: Output of ParamObj.jsp

### 3. Lab Guide 03: Using JSP Scriptlet

1. Write a program to display the multiples of two. Include `for` loop in the scriptlet block and display the numbers.

#### Solution:

The file used in this exercise is **Series.jsp**. The file is saved in basic directory.

```
<html>

  <body>

    <h1>Displaying </h1>

    <%

      int res;

      int j=2;

      for(int i=1;i<11;i++)

      {

        res=j*i;

        out.println(j + " * " + i + " = " + res);

        out.println("<br>");

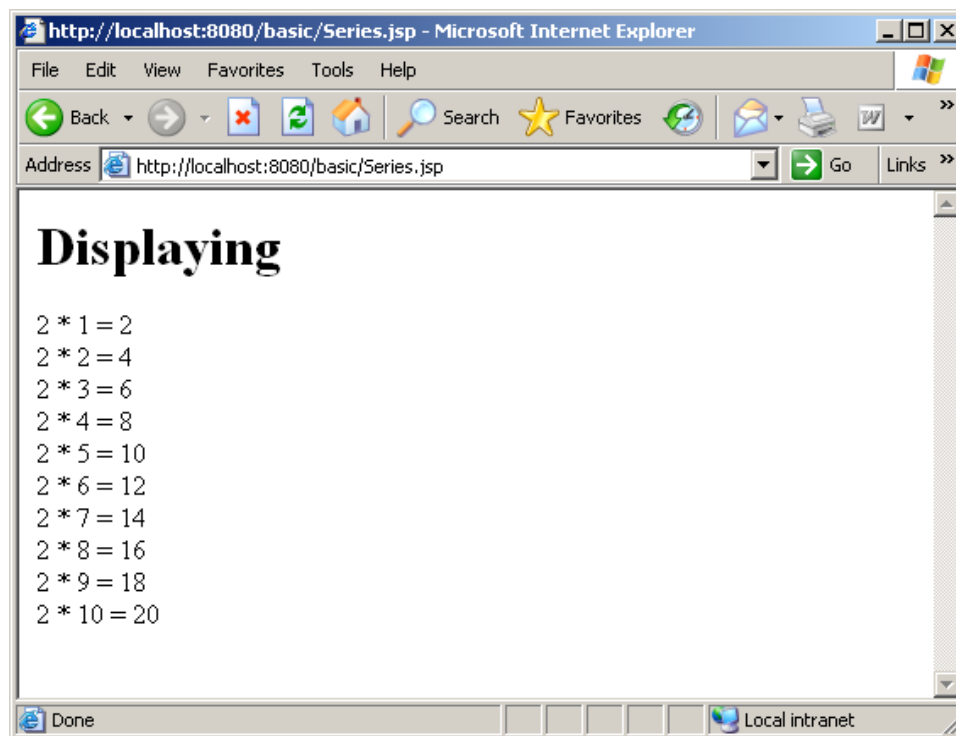
      }

    %>

  </body>

</html>
```

The output of the program is as shown in Figure 6.1.



**Figure 6.1: Output of Series.jsp**

2. Write a program to display the session ID, creation time, and the last accessed time of the Web page. Using `session.getId()`, `session.getCreationTime()`, and `session.getLastAccessedTime()`.

**Solution:**

The file used in this exercise is **Expression.jsp**. The file is saved in basic directory.

```
<html>

  <head>

    <title> JSP Expressions </title>

  </head>

  <body bgColor="white">

    <h2>Simple JSP Expression</h2>

    <ul><font color = "#0000FF">
```

```
<li> <font color = "#0000FF"> Session id:<%= session.getId()
%>

<li> <font color = "#0000FF"> Creation time:<%= new
java.util.Date(session.getCreationTime()) %>

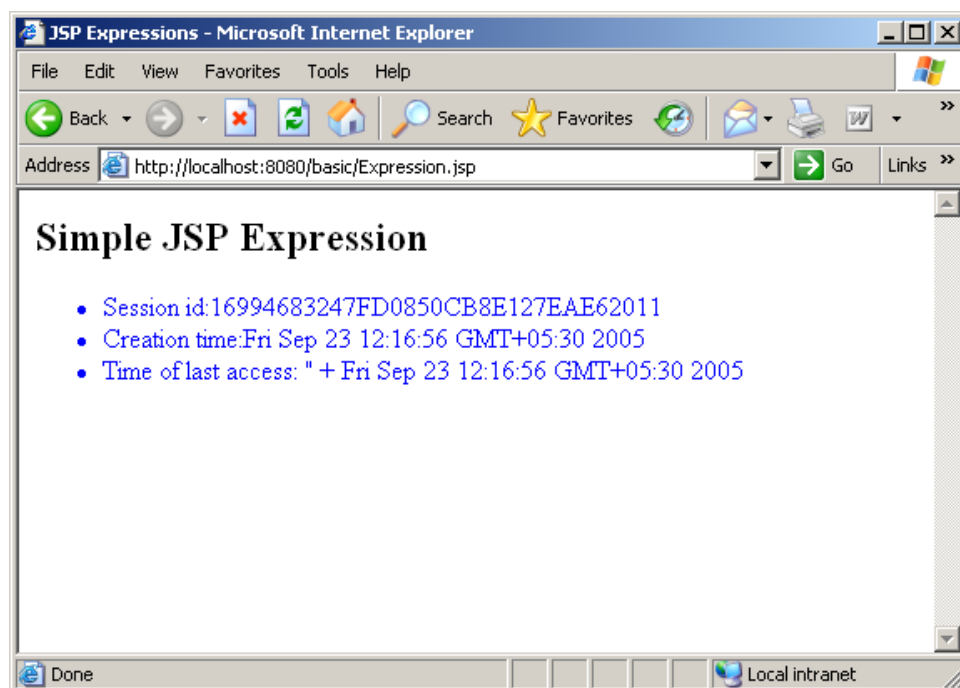
<li> <font color = "#0000FF"> Time of last access: " + <%= new
java.util.Date(session.getLastAccessedTime()) %>

</ul>

</body>

</html>
```

The output of the program is as shown in Figure 6.2.



**Figure 6.2: Output of Expression.jsp**

### Do It Yourself

1. Build a login page for online banking, accepting the user name and password. Use declarations to define the string variables. Include Java code in the scriptlet block to assign the details to the string variables entered by the user.

**Solution:**

The file used in this exercise is **Expression\_Login.jsp**. The file is saved in **basic directory**.

```
<html>

  <head>

    <title>Login Page</title>

  </head>

  <body bgcolor="white">

    <%if(request.getParameter("userid")==null&&
request.getParameter("password")==null ) { %>

      <center>

        <h2>Online Banking Login</h2>

      </center>

      <form method="GET"
action="http://localhost:8080/basic/Login.jsp">

        <b>User Id:</b> <input type="text" name="userid" size=15>

        <p>

          <b>Password:<b> <input type="password" name="password" size=8>

          <p>

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

      </form>

    <% } else { %>

    <%! String userid,password; %>

    <%

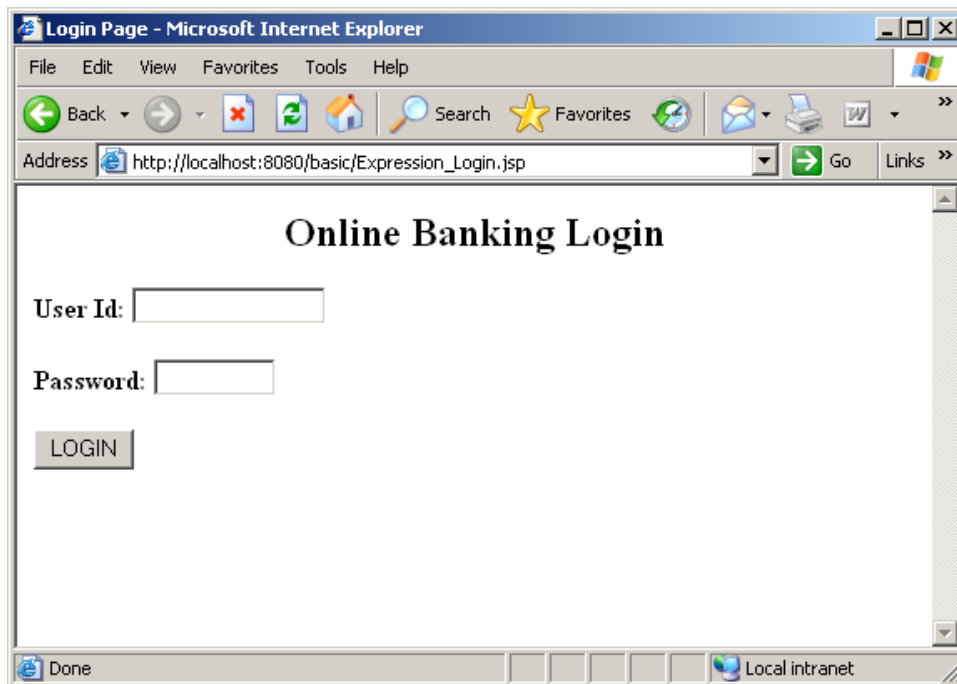
userid = request.getParameter("userid");
password = request.getParameter("password");

%>

    <% } %>
```

```
</body>  
</html>
```

The output of the program is as shown in Figure 6.3.



**Figure 6.3: Output of Login.jsp**

2. Use the above code and greet the login user. Make use of the expression elements to display the message.

**Solution:**

The file used in this exercise is **Login.jsp**. The file is saved in basic directory.

```
<html>  
  
  <head>  
  
    <title>Login Page</title>  
  
  </head>  
  
  <body bgcolor="white">  
  
    <%if(request.getParameter("userid")==null&&  
request.getParameter("password")==null ) { %>
```



```
<center>

<h2>Online Banking Login</h2>

</center>

<form method="GET"
action="http://localhost:8080/basic/Login.jsp">
  <b>User Id:</b> <input type="text" name="userid" size=15>

  <p>

  <b>Password:<b> <input type="password" name="password" size=8>

  <p>

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

</form>

<% } else { %>

<%! String userid,password; %>

<%

userid = request.getParameter("userid");

password = request.getParameter("password");

%>

<p>

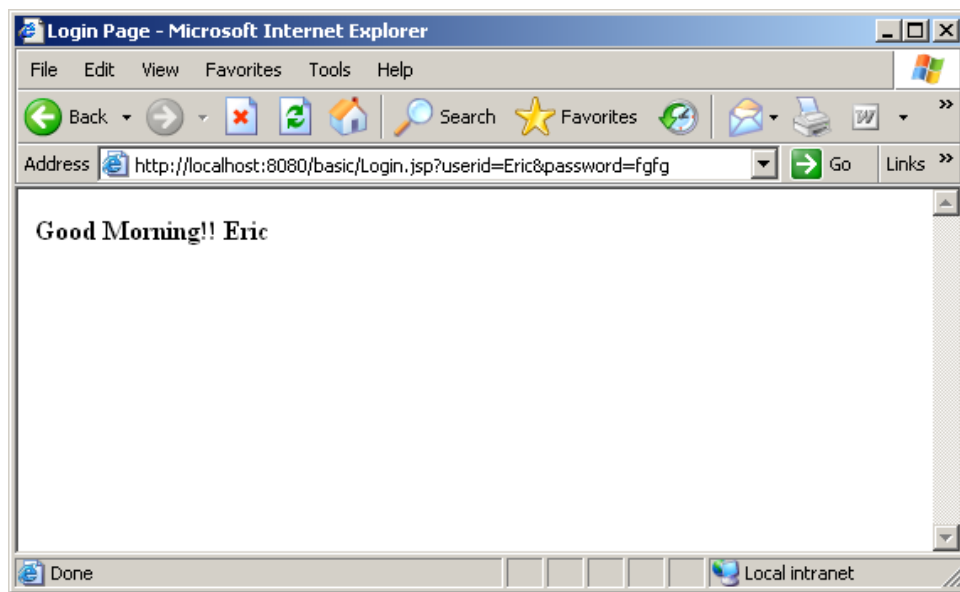
<b>Good Morning!! <%= userid %><p></b>

<% } %>

</body>
```

```
</html>
```

The output of the program is as shown in Figure 6.4.



**Figure 6.4: Output of Login.jsp**

## 4. Lab Guide 04: Using JavaBeans in Java Server Pages

1. Write a program to display a withdrawal amount screen to the user. Provide the user with three fields: Email Id, account number, and Balance amount. Display the withdraw status to the user after the user enters the email Id, account number and amount. In addition, send an e-mail to the user's email id informing about the withdrawal status.

**Solution:**

Enter the following code in the notepad and save the file as withdrawlstatus\_1.jsp in Session8 folder.

```
//withdrawlstatus_1.jsp

<html>
  <head>
    <title>MARKO</title>
  </head>
  <body>
    <H1 align="center">MARKO Withdrawal status checking
    Screen</H1>

    <form method ='post'>
      <h3>User Email ID <input type='text' name='email'></h3>
      <h3>Account no <input type='text' name='accNo'></h3>
      <h3>Balance Amount <input type='text' name='Bal'></h3>
      <input type='submit' value="Check Status">
    </form>
    <%
      String strAccount = request.getParameter("accNo");
      String strBal =    request.getParameter("Bal");
      String strEmail =  request.getParameter("email");
      if (strAccount == null || strBal == null || strEmail ==
        null)
      {
    %>
    <h3 align="center"> Please enter email address, account No
    and Balance and click the Check Status button to compute
    the withdrawal status</h3>
    <%
      }
      else
      {
    %>

    <jsp:useBean id="objBT" class="MARKO.BankTransactionMail"
    scope="session" >
    <jsp:setProperty name="objBT" property="accountNo"
    value="<%=strAccount%>" />
    <jsp:setProperty name="objBT" property="balance"
    value="<%=strBal%>" />
    </jsp:useBean>
```

```
<h3>For Account no <jsp:getProperty name="objBT"
property="accountNo"/> :
<%
    String strMessage = objBT.withdrawAllowed();
    out.println(strMessage);
%>

<br>
<br>
<%=objBT.sendMail(strEmail,strMessage)%>
</h3>
<%
}
%>
</body>
</html>
```

Enter the following code in notepad and save the file as **BankTransactionMail.java** in **Session8** folder.

```
//BankTransactionMail.java
```

```
package MARKO;
import javax.mail.*;
import javax.mail.internet.*;
import java.util.*;

public class BankTransactionMail implements java.io.Serializable
{
    int intAccountNo;
    int intBalance;

    public String getBalance()
    {
        return String.valueOf(intBalance);
    }

    public void setBalance(String strBal)
    {
        intBalance = Integer.parseInt(strBal);
    }

    public String getAccountNo()
    {
        return String.valueOf(intAccountNo);
    }

    public void setAccountNo(String strAccNo)
    {
        intAccountNo = Integer.parseInt(strAccNo);
    }
}
```

```
public String withdrawAllowed()
{
    if (intAccountNo < 5000)
    {
        if (intBalance > 10000)
        {
            return "withdrawal is allowed!!!<br>To withdraw please
            <a href='withdraw.jsp?bal= " +
            String.valueOf(intBalance) + "'>click here.</a>";
            //The file withdraw.jsp is created in Example 1
            of Do It Yourself.
        }
        else
        {
            return "withdrawal is not allowed!!!";
        }
    }
    else
    {
        return "withdrawal is allowed!!!<br>To withdraw please <a
        href='withdraw.jsp?bal=" + String.valueOf(intBalance) +
        "'>click here.</a>";
        //The file withdraw.jsp is created later in Do it
        Yourself part.
    }
}

public String sendMail(String strTo,String strMsg)
{
    String result = "<BR><BR><BR>";
    String host = "Name_of_Machine_Running_SMTP_Service";
    Properties prop = System.getProperties();
    prop.put("libmail.n-syst.com", host);
    Session session = Session.getDefaultInstance(prop, null);
    session.setDebug(true);

    try
    {
        MimeMessage msg = new MimeMessage(session);
        msg.setFrom(new InternetAddress("admin@MARKO.com"));
        msg.setRecipients(Message.RecipientType.TO,
        InternetAddress.parse(strTo, false));
        msg.setSubject("Automated Mail from MARKO JSP Page");
        MimeBodyPart mbp = new MimeBodyPart();
        mbp.setText(strMsg);
        Multipart mp = new MimeMultipart();
        mp.addBodyPart(mbp);
        msg.setContent(mp);
        msg.setSentDate(new Date());
        Transport.send(msg);
        result = result + "<H3 align='center'>Mail was
        successfully sent to </B></FONT>: "+ strTo +"<BR>";
        result = result+"</H3>";
    }
    catch (MessagingException ex)
    {
    }
```

```

        result = result + "<H3 style='color:red'
        align='center'>Error : </B><BR><HR> "+"<FONT SIZE=3
        COLOR=\"black\">"+ex.toString()+"<BR><HR></h3>";
    }
    catch (Exception e)
    {
        result = result + "<H3 style='color:red'
        align='center'>Error : </B><BR><HR> "+"<FONT SIZE=3
        COLOR=\"black\">"+e.toString()+"<BR><HR></h3>";
    }
    finally
    {
        return result;
    }
}
}
}

```

Enter the path [http://localhost:8080/Session8/withdrawlstatus\\_1.jsp](http://localhost:8080/Session8/withdrawlstatus_1.jsp) in the address bar. The output appears as shown in Figure 8.1.

MARKO Withdrawal status checking Screen

User Email ID

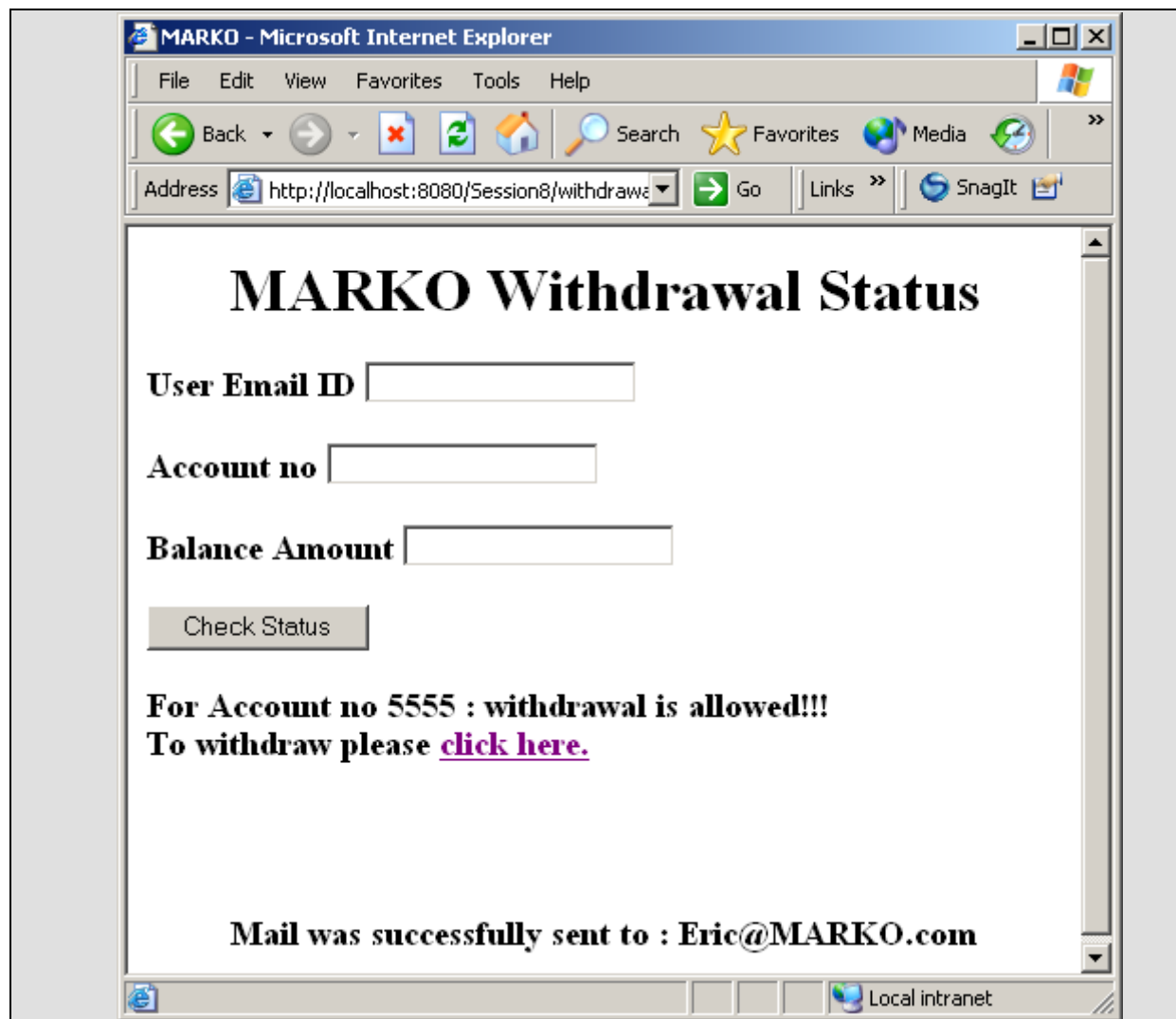
Account no

Balance Amount

Please enter email address, account No and Balance and click the Check Status button to compute the withdrawal status

**Figure 8.1: Check Withdrawal Status Screen**

Enter the e-mail Id, account number, and balance amount and click the Check Status button. The output appears as shown in Figure 8.2.



**Figure 8.2: Withdrawal Status Screen**

The automated mail sent to the user appears as shown in Figure 8.3.

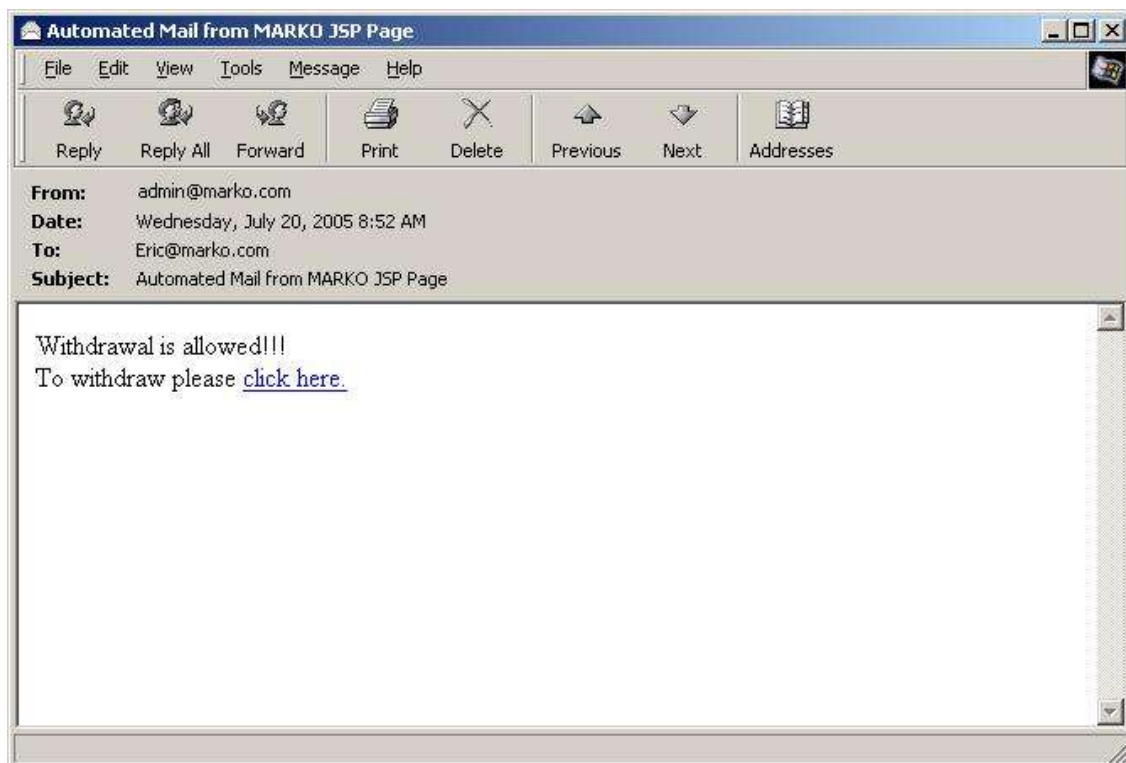


Figure 8.3: Automated Mail

To withdraw, click on the link provided in the figure above. The output appears as shown in Figure 8.4.

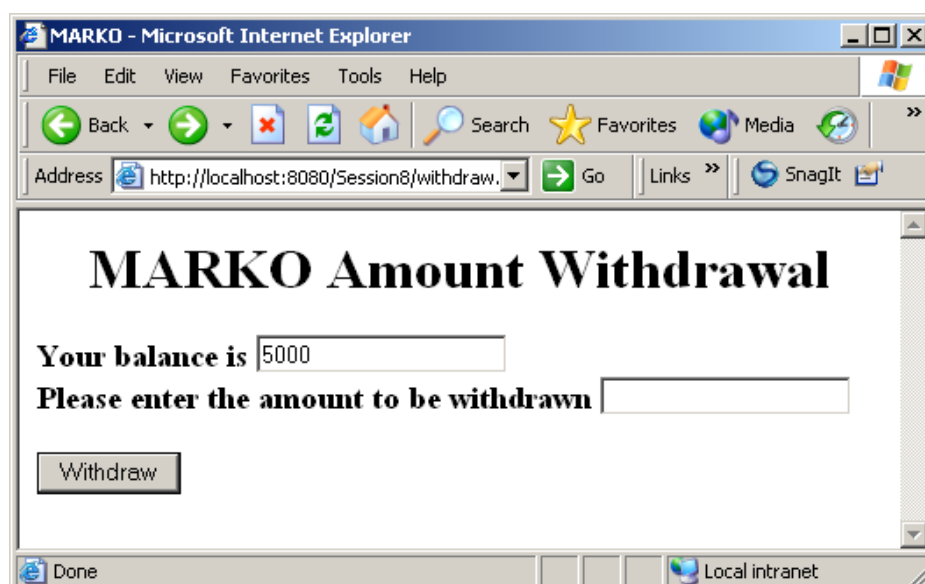
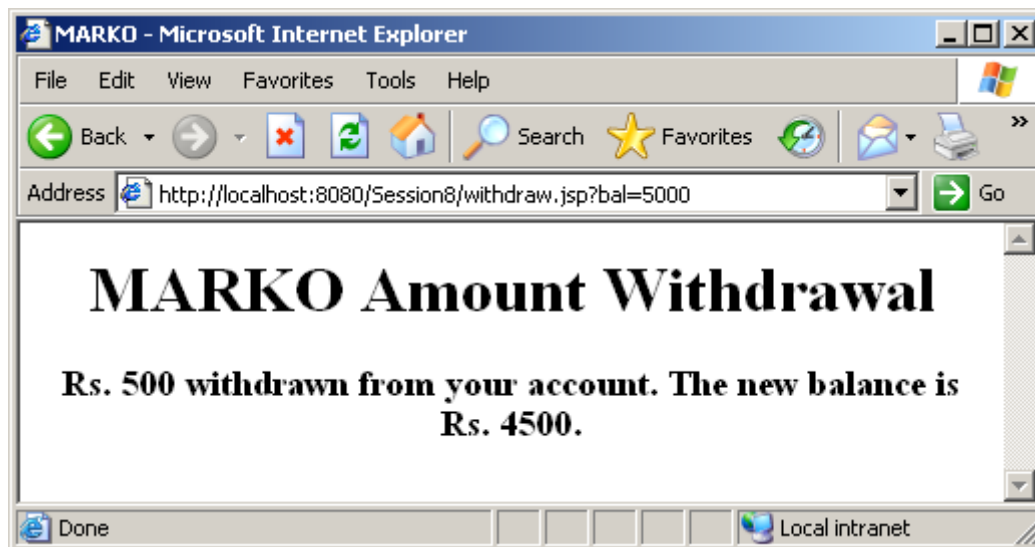


Figure 8.4: Withdrawal Screen



Enter the amount to be withdrawn and click the Withdraw button. The output appears as shown in Figure 8.5.



**Figure 8.5: Withdrawal Confirmation Screen**

### Do it Yourself

---

3. Write a program to display the withdrawal status to the user. If the account number is valid, display the withdrawal screen to the user. The withdrawal screen should display the balance entered by the user on the withdrawal status screen and ask the user to enter the amount to be withdrawn. Display the balance to the user after withdrawing the amount.

### Solution:

//withdraw.jsp

```
<html>
  <head>
    <title>MARKO</title>
  </head>
  <body>
    <H1 align="center">MARKO Amount Withdrawal Screen</H1>

    <%
```

```

        String amt = request.getParameter("amt") ;
        String balance = request.getParameter("bal");
        if (amt == null)
        {
    %>

    <form method="post">
    <h3>
        Your balance is <input type='text' readonly name='bal'
        value='<%=balance%>'><br>
        Please enter the amount to be withdrawn <input
        type="text"
        name="amt">
        <input type="submit" value="Withdraw">
    </form>

    <%
    }
    else
    {
    %>

    <jsp:useBean id="objW" class="MARKO.AccountBean"
    scope="page" />

    <jsp:setProperty name="objW" property="balance"
    value="<%=balance%>" />

    <jsp:setProperty name="objW" property="amount"
    value="<%=amt%>" />

    <H3 align="center">Rs. <%=amt%> withdrawn from your
    account
    the new balance is <jsp:getProperty name="objW"
    property="balance"/>
    </h3>

    <%
    }
    %>

    </body>
</html>

```

- 1. Save the file as withdraw.jsp in basic directory.**
- 2. Open Notepad and enter the following code.**

```
//AccountBean.java
```

```
package MARKO;
```

```
public class AccountBean implements java.io.Serializable
{
    private int amount = 0;
    private int balance = 0    ;

    public String getAmount()// throws Exception
    {
        return String.valueOf(amount);
    }

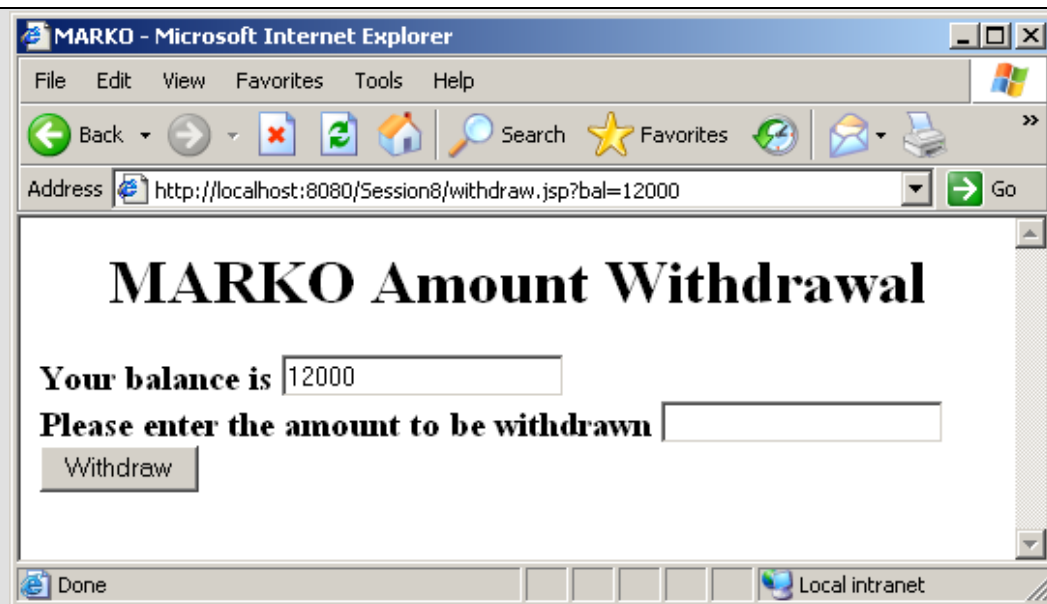
    public void setAmount(String strAmt)// throws Exception
    {
        this.amount=Integer.parseInt(strAmt);
    }

    public String getBalance()
    {
        if (balance < amount)
        {
            return "Balance is not sufficient to withdraw money.";
        }
        else
        {
            balance = balance - amount;
        }
        return ("Rs."+String.valueOf(amount)+" withdrawn from your
        account. The new balance is Rs"+String.valueOf(balance));
    }

    public void setBalance(String strBal)
    {
        balance = Integer.parseInt(strBal);
    }
}
```

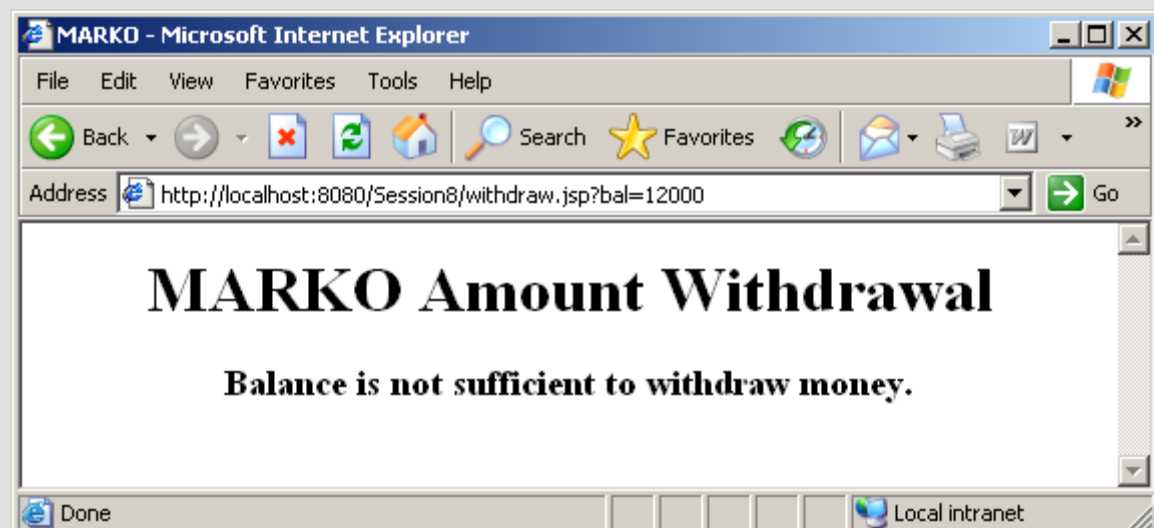
- 3. Save the file as AccountBean.java in Session8 directory.**
- 4. Stop and restart Tomcat server.**
- 5. Open Internet Explorer and enter <http://localhost:8080/Session8/withdraw.jsp> in the browser.**

The output appears as shown in Figure 8.6.



**Figure 8.6: Withdraw Amount Screen**

Enter an amount greater than the balance amount. A message will be displayed to the user informing that balance amount is insufficient.



**Figure 8.7: Output with invalid amount**

Enter the amount to be withdrawn and click the Withdraw button. The output appears as shown in Figure 8.7 when you click on **Withdraw** button.

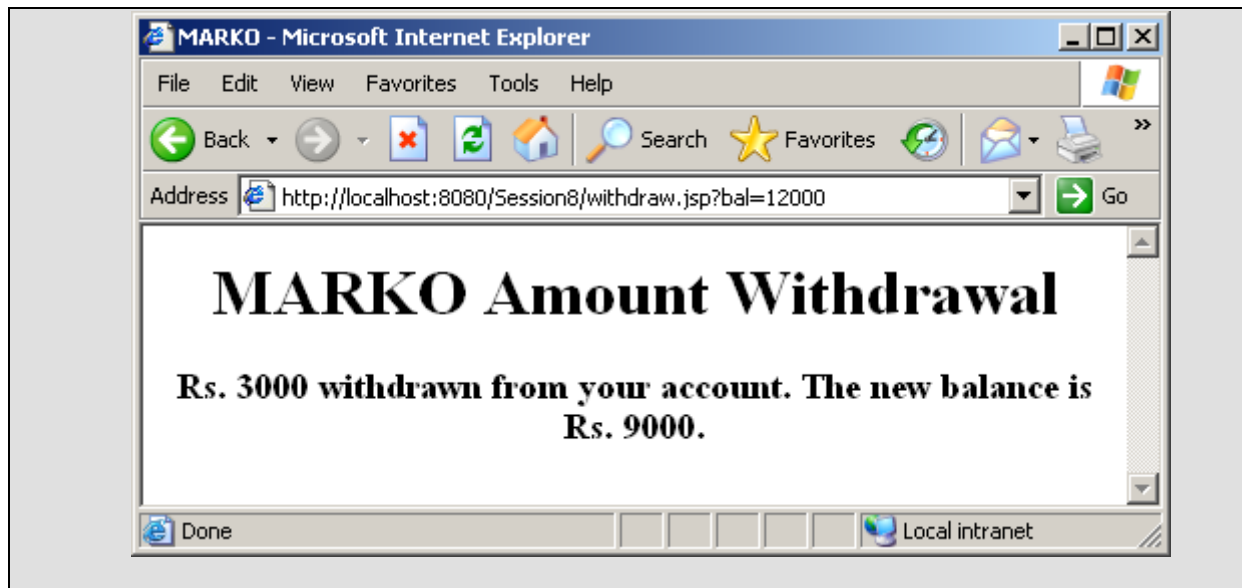


Figure 8.8: Output with valid amount