KARPAGAM ACADEMY OF HIGHER EDUCATION

(Deemed University Established Under Section 3 of UGC Act 1956)

Coimbatore – 641 021.

(For the candidates admitted from 2016 onwards)

DEPARTMENT OF COMPUTER SCIENCE

SUBJECT : J2EE Lab

SEMESTER : III **L T P C**

SUBJECT CODE: 16CSP311 **CLASS : II M.Sc.CS** **4 0 0 2**

LIST OF PROGRAMS

1. Create a sign in form in servlets.
2. Write a servlet Program to lock a server.
3. Write a servlet program that returns list of information in table format.
4. Design a counter that counts number of times user has visited the site in current browsing session.
5. Write a program to retrieve cookies information
6. Build a JAVA Bean for opening an applet from JAR file.
7. Write a program to add controls in BEAN.
8. Design a counter in JAVA BEAN.
9. Write a program to stream contents of a file using JSP.
10. Write a program to insert an applet into JSP page.

Enable | Enlighten | Enrich
(Deemed to be University)
(Under Section 3 of UGC Act 1956)

KARPAGAM ACADEMY OF HIGHER EDUCATION

(Deemed University Established Under Section 3 of UGC Act 1956)

Coimbatore – 641 021.

(For the candidates admitted from 2016 onwards)

DEPARTMENT OF COMPUTER SCIENCE

SUBJECT : J2EE Lab

SEMESTER : III **L T P C**

SUBJECT CODE: 16CSP311 **CLASS : II M.Sc.CS** **4 0 0 2**

LIST OF PROGRAMS

1. Create a sign in form in servlets.
2. Write a servlet Program to lock a server.
3. Write a servlet program that returns list of information in table format.
4. Design a counter that counts number of times user has visited the site in current browsing session.
5. Write a program to retrieve cookies information
6. Build a JAVA Bean for opening an applet from JAR file.
7. Write a program to add controls in BEAN.
8. Design a counter in JAVA BEAN.
9. Write a program to stream contents of a file using JSP.
10. Write a program to insert an applet into JSP page.

Ex.No:1 **SIGNIN FORM IN SERVLETS**

AIM:

To create a sign in form in the servlet.

ALGORITHMS:

Step 1: Start the process.

Step 2: Open the net beans application.

Start→All programs→Netbeans.

Step 3: Create a new **index.jsp**page.

Step 4: Give the name as login here username and password is given using text boxes and submit button.

Step 5: Create a **success.html** file to print the content as successful.

Step 6: Create a**error.html** file to print the content as unsuccessful.

Step 7: Create new servlet page.

Web pages→new→servlet.

Step 8: Use a method for **getwriter()** get the username and password.

Step 9: Check the condition as following.

If(un.equals(“user”)&&pw.equals(“user”))

Step10: If it is true it has to show **“success.html”** page else **“error.html”.**

Step 11: Save the program and display the result.

Step 12: Stop the process.

CODING:

index.jsp

```
<html>
<head>
<title>LOGIN FROM</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
<body>
<form action="prg1" method="post">
        Enter username:<input type="text" name="username"><br>
        Enter password:<input type="text" name="password"><br>
<input type="submit" value="login">
</form>
</body>
```

success.html

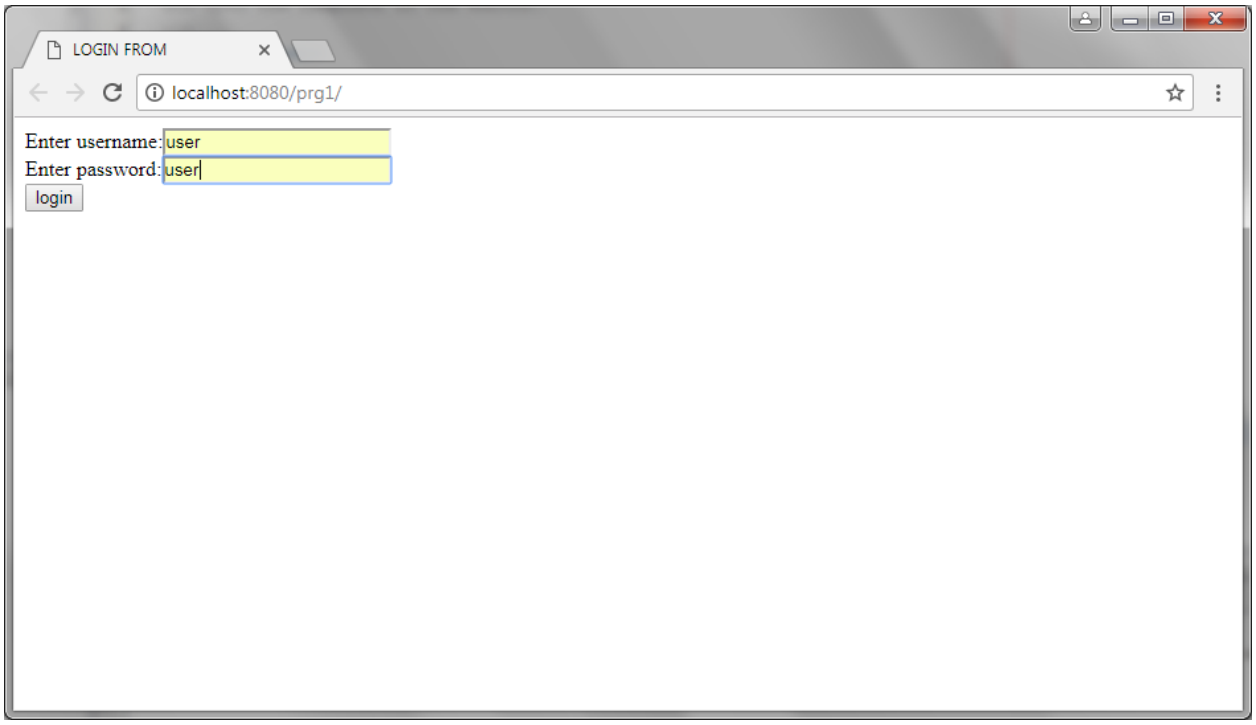
```
</html>
<html>
<head>
<title>LOGIN FORM</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
<body> LOGIN SUCCESSFULLY
```

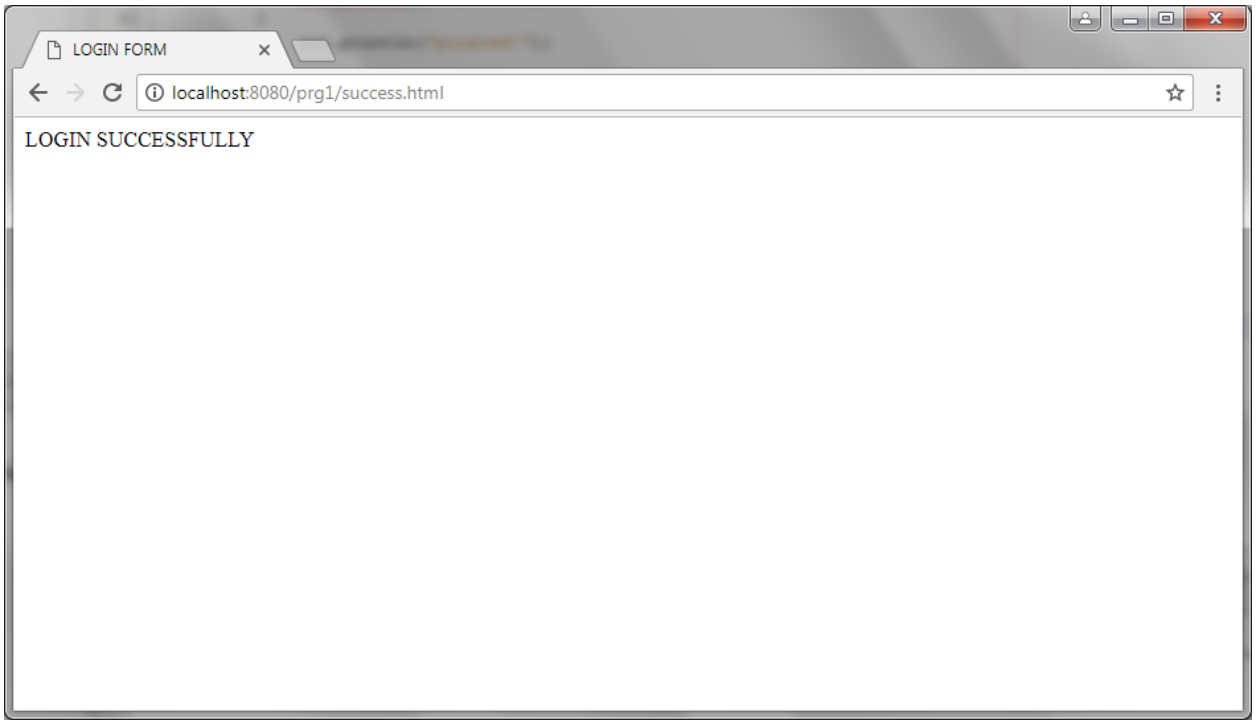
```
</body>
</html>
```

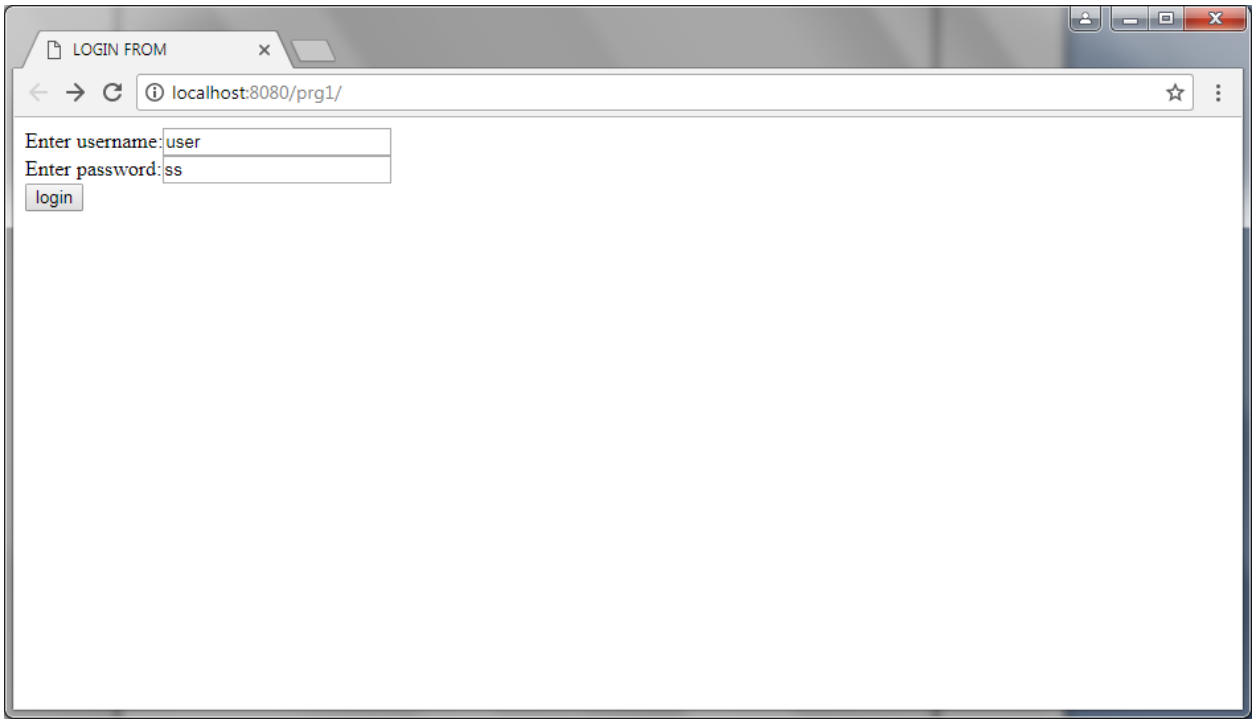
error.html

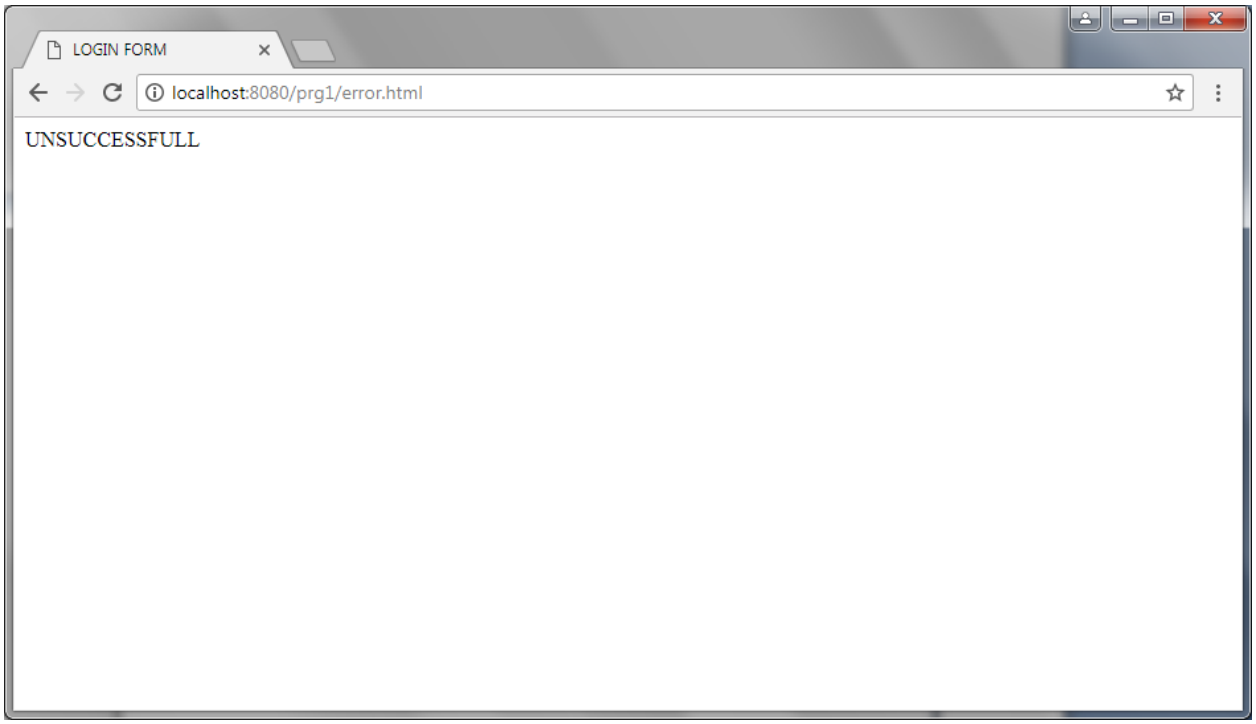
```
<html>
<head>
<title>LOGIN FORM</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
<body>
    UNSUCCESSFULL
</body>
</head>
</html>
```

OUTPUT:


LOGIN FROM
localhost:8080/prg1/
Enter username:user
Enter password:user
login


LOGIN FORM
localhost:8080/prg1/success.html
LOGIN SUCCESSFULLY


LOGIN FROM
localhost:8080/prg1/
Enter username: user
Enter password: ss
login



RESULT:

Thus the Login program has been successfully executed.

Ex.No:2 **SERVLET PROGRAM TO LOCK A SERVER**

AIM:

To create a servlet Program to Lock a server.

ALGORITHM:

Step1: Start the process.

Step2: Open the net beans application.

Start→All programs→Netbeans.

Step3: Create new servlet page named as **"prg2.java"**.

Web pages→new→servlet.

Step4: Declare the requiredpackages.keylockclasss is extends by genericServlet class.

Step5: Initialize the variables askey,host,port.

Step6: Check the condition as following.

If(!keyfitsserver(key,host,port))

Step7: If above the condition is true print as "pirated",or else "valid".

Step8: Similarly check the host and port address.

Step9: Save the program and display the result.

Step 10: Stop the process.

CODING:

```
importjava.io.IOException;
importjava.io.PrintWriter;
importjava.net.InetAddress;
importjava.net.UnknownHostException;
importjavax.servlet.*;
importjavax.servlet.http.HttpServletRequest;
importjavax.servlet.http.HttpServletResponse;
public class keylock extends GenericServlet {
privateinti;
public void service(ServletRequest request, ServletResponse response)
throwsServletException, IOException {{
response.setContentType("text/html;charset=UTF-8");
PrintWriter out = response.getWriter();
        String key=getInitParameter("key");
        String host=request.getServerName();
int port=request.getServerPort();
if(!keyfitsserver(key,host,port))
        {
out.println("pirated!");
        }
else
        {
```

```
out.println("valid");
        }}
response.setContentType("text/plain");
PrintWriter out = response.getWriter();
        String key=getInitParameter("key");
        String host=request.getServerName();
int port;
port=request.getServerPort();
if(!keyfitsserver(key,host,port))
        {
out.println("pirated!");
        }
else
        {
out.println("valid");
        }}
privatebooleankeyfitsserver(String key,Stringhost,int port)
{
if(key==null){
return false;
    }
longnumerickey=0;
try
```

```
    {
numerickey=Long.parseLong(key);
    }
catch(NumberFormatException e){
return false;
        }
bytehostIP[];
try{
hostIP=InetAddress.getByName(host).getAddress();
}catch(UnknownHostException e){
return false;     }
longServerCode=0;
for(inti=0;i<4;i++)
          {
ServerCode<<=8;
ServerCode=hostIP[i];
          }
ServerCode<<=32;
ServerCode=port;
longaccesscode=~numerickey;
return(ServerCode==accesscode);
}
}
```

OUTPUT:

RESULT:

Thus locking program has been successfully executed.

Ex.No: 3 **LIST OF INFORMATION IN TABLE FORMAT**

AIM:

To write a program for information in table format.

ALGORITHM:

Step 1: Start the process.

Step2: Open the net beans application.

Start→All programs→Netbeans.

Step 3: Create **"index.jsp"** page.

Step4: Write a select statement for select country name with submit button.

Step 5: Create new servlet page named as **"cricket.java."**

Web pages→new→servlet.

Step 6: Use **getParameter( )** get the country_name from jsp page.

Step 6: Display the corresponding details of team using following,

condition**if(country.equals("country_name"))**

Step 7: If the country_name as India to show the Indian team details else country_name as Australia to show the Australia team details.

Step 8: Save the program and display the result.

Step 9: Stop the process.

CODING:

index.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>cricket</title>
</head>
<body>
<table width="100">
<trbgcolor="pink">
<td colspan="2" align="center">
<font size="8"> CricketNext.com</font><br><hr size="4">
</td>
</tr>
</table><br><br>
<form methopd="post" action="cricket1">
<table align="center">
<tr>
<td> Select country name</td>
<td><select name="count">
<option>india</option>
```

```
<option> Australia</option>
</select></td>
</tr>
<tr><td colspan="2" align="center" ><br>
<input type="submit" value="go">
</td></tr>
</table>
</form>
</body>
</html>
```

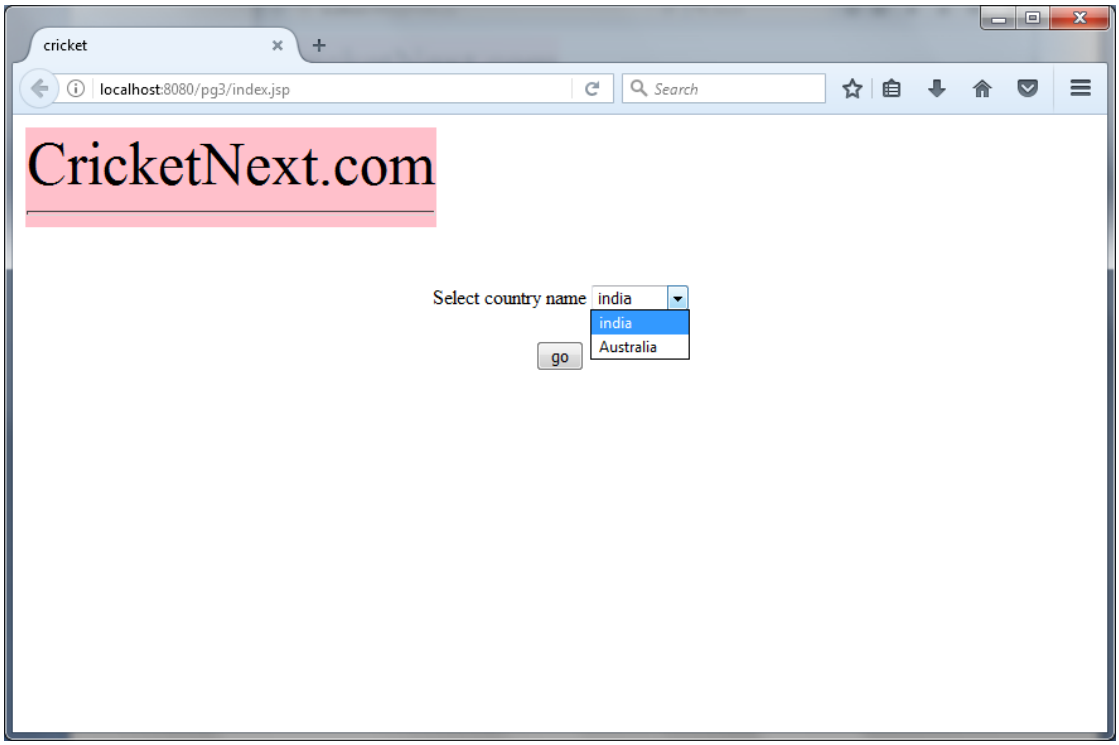
cricket.java

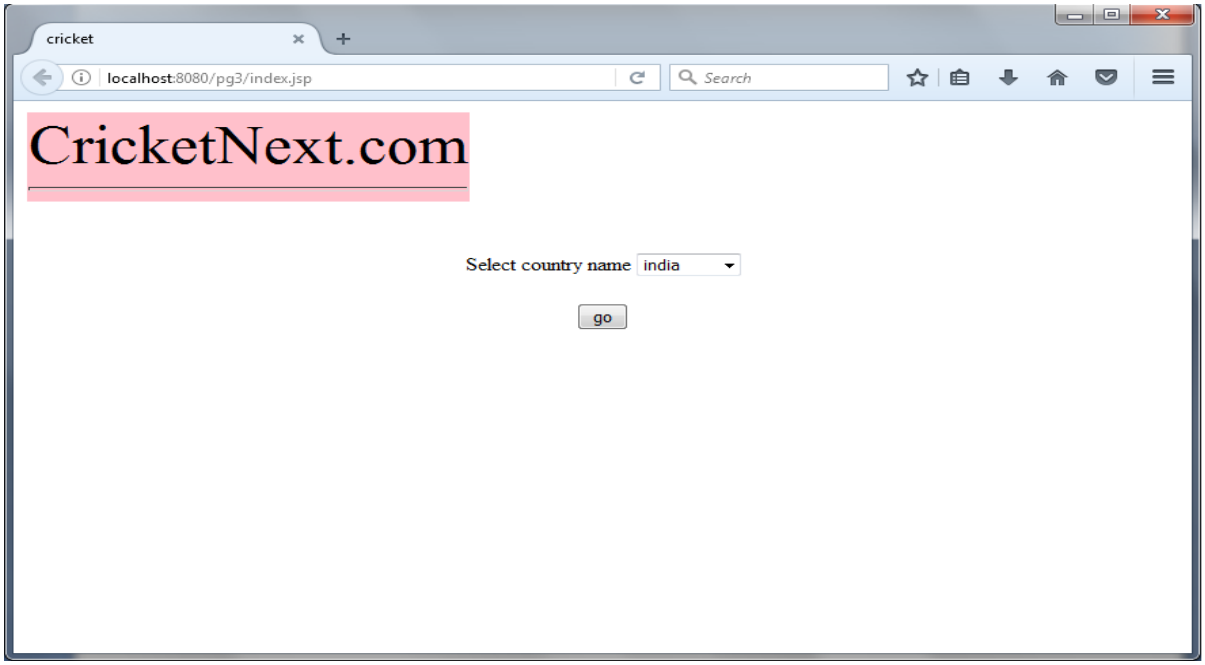
```
importjava.io.IOException;
importjava.io.PrintWriter;
importjavax.servlet.ServletException;
importjavax.servlet.http.HttpServlet;
importjavax.servlet.http.HttpServletRequest;
importjavax.servlet.http.HttpServletResponse;


public class cricket1 extends HttpServlet
{
protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
throwsServletException, IOException
```

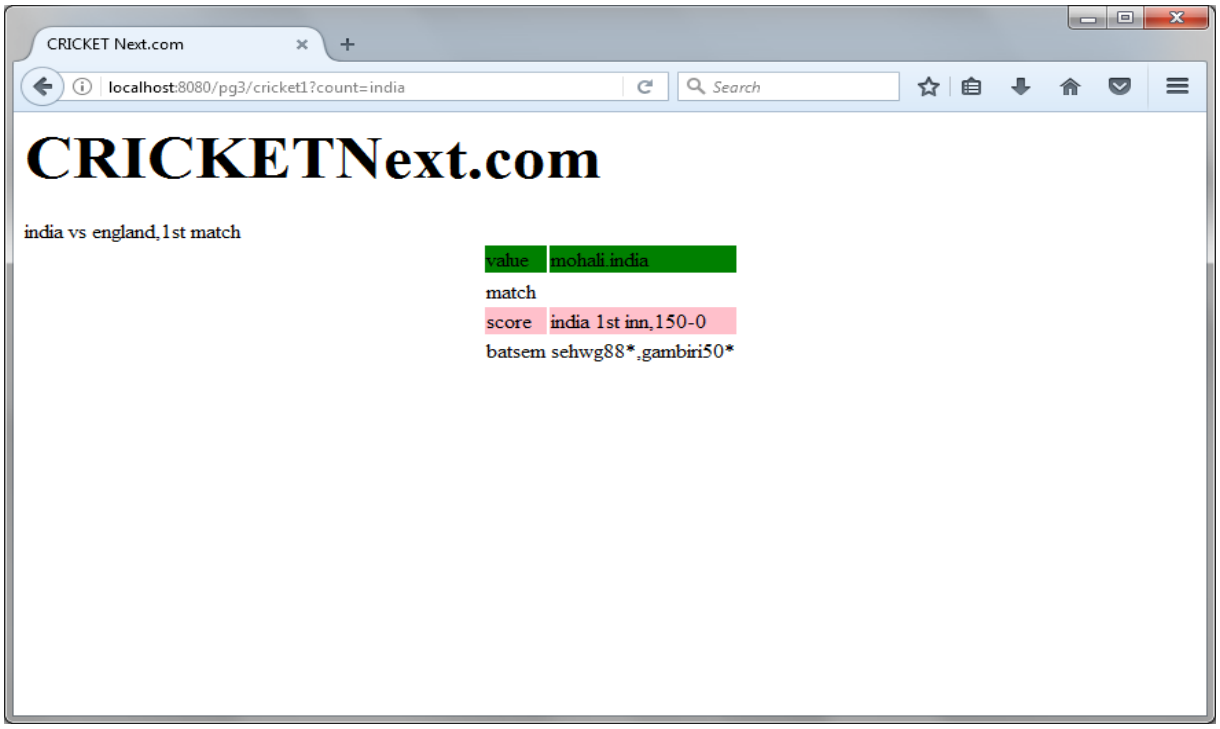
```
{
response.setContentType("text/html;charset=UTF-8");
PrintWriter out = response.getWriter();
        String country=request.getParameter("count");
out.println("<html>");
out.println("<head>");
out.println("<title>CRICKET Next.com</title>");
out.println("</head>");
out.println("<body>");
out.println("<h1><font size=20>CRICKETNext.com</font></h1>");
out.println("<table align=center><trbgcolor=green>");
if(country.equals("india"))
              {
out.println("<td> value</td><td>mohali.india</td>");
out.println("<tr><tr><td>match</td></td>indiavs england,1st
match</td></tr><trbgcolor=pink>");
out.println("<td>score</td><td>india 1st inn,150-0</td><tr><td>batsem</td>");
out.println("<td>sehwg88*,gambiri50*</td>");
              }
else if(country.equals("Australia"))
             {
out.println("<td>veune</td><td>mgg.australia</td>");
```

```
out.println("<tr><tr><td>match</td></td>Australia vs southafrica,3st
match</td></tr><trbgcolor=pink>");

out.println("<td>score</td><td> Australia 1st inn,150-0</td><tr><td>batsem</td>");

out.println("<td>hayden88*,karthik50*</td>");

        }

out.println("</body>");

out.println("</html>");

   }
```

OUTPUT


cricket
localhost:8080/pg3/index.jsp
Search
CricketNext.com
Select country name
india
india
Australia
go


cricket
localhost:8080/pg3/index.jsp
Search
CricketNext.com
Select country name
india
go



RESULT:

Thus the Cricket program has been successfully executed.

Ex.No:4 **WEBPAGE HITS**

AIM:

Create the java program using servlet to hit the page on the browser.

ALGORITHM:

Step1: Start the process.

Step2: Open the net beans application.

Start→All programs→Netbeans.

Step 3: Create new servlet page named as pagehit.java

Web pages→new→servlet.

Step 4: Declare the necessary packages, and initialize the class name as **"pagehit"**

Step 5: In the servlet coding enable session as true.

Step 6: In the servlet page the **getattribute()** and the **setattribute()** are used to get the request and response to display.

Step 7: Save the program and display the result.

Step 8: Stop the process.

.

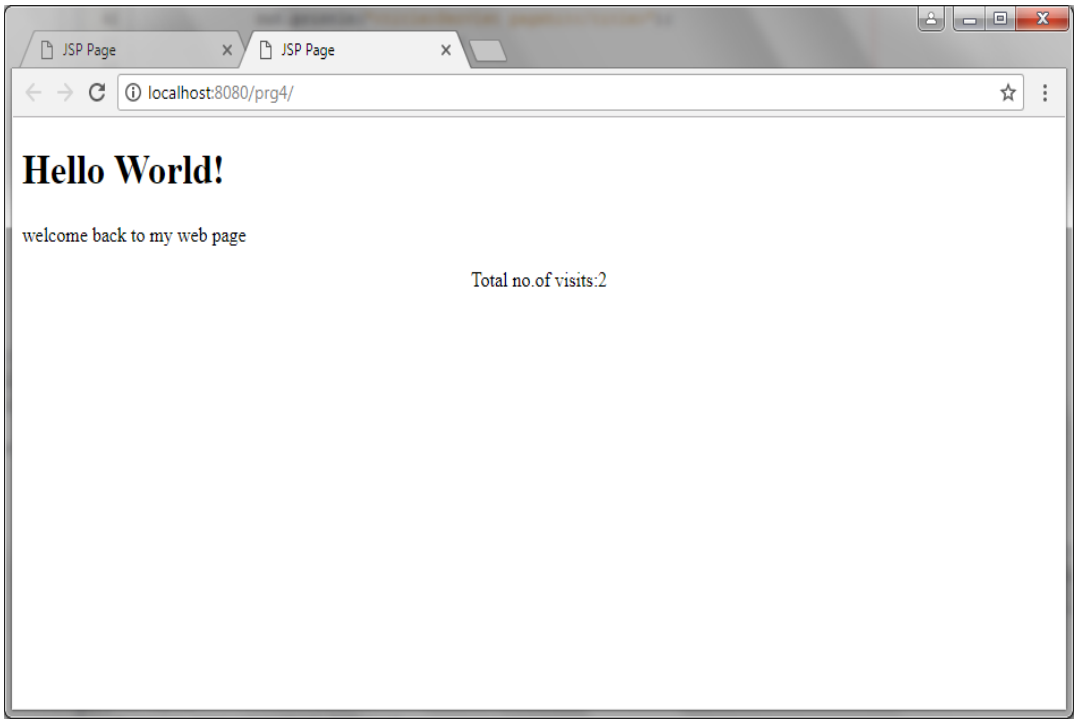
CODING:

pagehit.java

```
importjava.io.IOException;

importjava.io.PrintWriter;

importjavax.servlet.ServletException;

importjavax.servlet.annotation.WebServlet;

importjavax.servlet.http.HttpServlet;

importjavax.servlet.http.HttpServletRequest;

importjavax.servlet.http.HttpServletResponse;

importjavax.servlet.http.HttpSession;


public class pagehit extends HttpServlet {

protected void processRequest(HttpServletRequest request, HttpServletResponse
response)

throwsServletException, IOException {

response.setContentType("text/html;charset=UTF-8");

PrintWriter out = response.getWriter();

out.println("<html>");

out.println("<head>");

out.println("<title>Servlet pagehit</title>");

out.println("</head>");

out.println("<body>");
```

```
HttpSession session1=request.getSession(true);
        Integer param=(Integer)session1.getAttribute("my session");
if(param!=null)
        {
session1.setAttribute("my session",new Integer(param.intvalue()+1));
}else
        {
param=new Integer(1);
session1.setAttribute("mysession",param);
        }
out.println("hit the browser",param.intvalue()+"times");
out.println("</body>");
out.println("</html>");
out.close();
      }
   }
```

OUTPUT:



RESULT:

Thus the Cricket program has been successfully executed.

Ex.No.5 **COOKIES**

AIM:

To write a program to retrieving cookies information.

ALGORITHM:

Step1: Start the process.

Step2: Open the net beans application.

Start→All programs→Netbeans.

Step 3: Create new jsp page named as **"index.jsp"**.

Step 4: To create a form with input text field and submit button

Step 5: Create new servlet page named as **"servlet1.java"**

Web pages→new→servlet.

Step 6: Declare the necessary packages, initialize the class name as **"servlet1"**

Step 7: Use **getParameter()** get the firstname from **"index.jsp"** file.

Step 8: Create new servlet page named as **"servlet2.java"**

Web pages→new→servlet.

Step 9: Using **doPost( )** create a cookies value & print the result. In this **servlet1.java** is move to the **servlet2.java.**

Step 10: Using **doGet( )** get the value of **servlet1.java**& print the result.

Step 11: Stop the process.

CODING:

index.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>JSP Page</title>
</head>
<body>
<form action="servlet1"method="post">
name:<input type="text" name="usernmae"/><br>
<input type="submit" valur="go"/>
</form>
</body>
</html>
```

servlet1.java

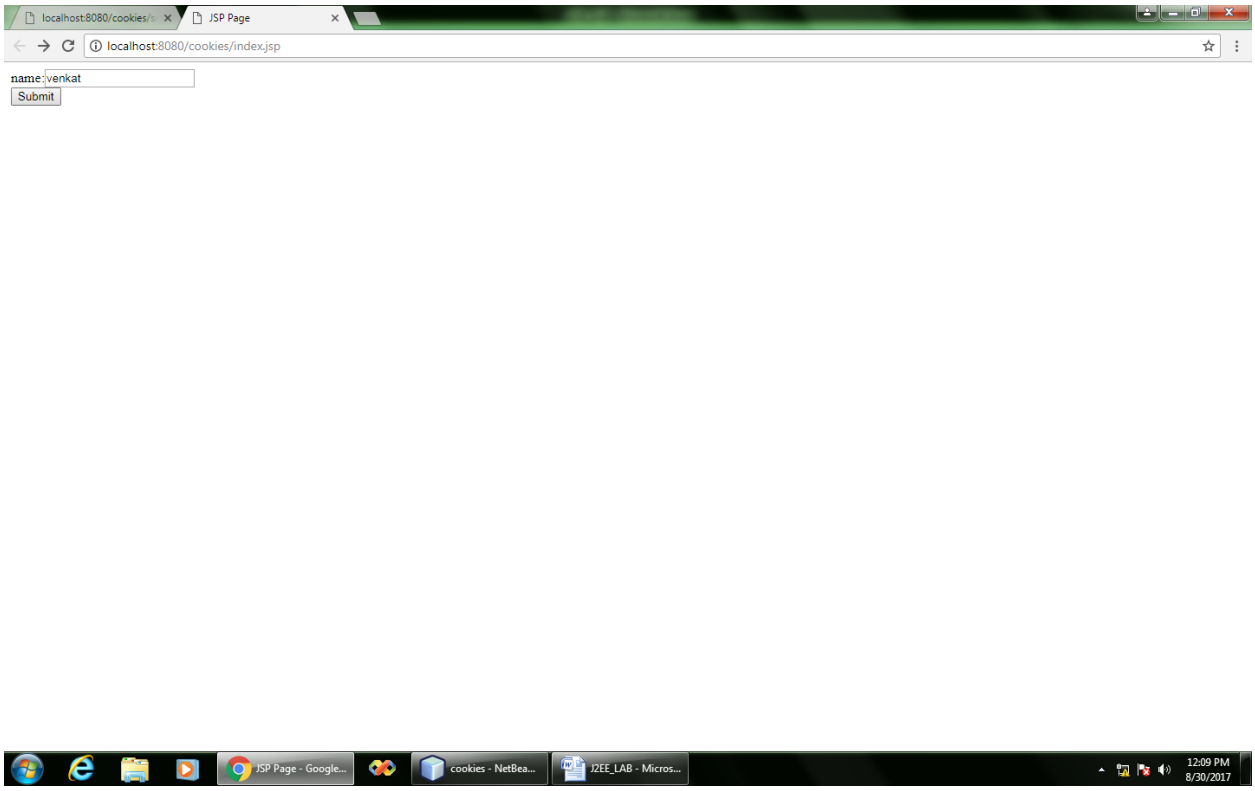
```
importjava.io.IOException;
importjava.io.PrintWriter;
importjavax.servlet.ServletException;
importjavax.servlet.http.Cookie;
importjavax.servlet.http.HttpServlet;
importjavax.servlet.http.HttpServletRequest;
```

```
importjavax.servlet.http.HttpServletResponse;
public class servlet1 extends HttpServlet {
public void doPost(HttpServletRequestrequest,HttpServletResponse response)
  {
try{
response.setContentType("text/html");
PrintWriter out=response.getWriter();
            String n=request.getParameter("usernmae");
out.print("welcome"+n);
            Cookie ck=new Cookie("usernmae",n);
response.addCookie(ck);
out.print("<form action='servlet2'>");
out.print("<input type='submit' value='go'>");
out.print("</form>");
out.close();
        }
catch(Exception e) {
System.out.println(e);
        }
  }
}}}
```

servlet2.java

```
importjava.io.IOException;
importjava.io.PrintWriter;
importjavax.servlet.ServletException;
importjavax.servlet.http.Cookie;
importjavax.servlet.http.HttpServlet;
importjavax.servlet.http.HttpServletRequest;
importjavax.servlet.http.HttpServletResponse;
public class servlet2 extends HttpServlet {
   @Override
public void doGet(HttpServletRequest request, HttpServletResponse response)
   {
try{
response.setContentType("text/html");
PrintWriter out=response.getWriter();
        Cookie ck[]=request.getCookies();
out.print("hello"+ck[1].getValue());
out.close();
      }
catch(Exception e)
        {
System.out.println(e);
        }}
```

OUTPUT:



RESULT:

Thus the above program has been successfully completed and verify the output.

Ex.No:6 **OPENING AN APPLET FROM JAR FILE**

AIM:

To build a javabean for opening an applet from jar file.

ALGORITHM:

Step 1: Start the process.

Step 2: Open the net beans application.

Start->All programs->Netbeans.

Step 3: Create new Japplet named as Numberaddition

Web pages->new->Japplet.

Step 3: Place corresponding labels and textboxes.

Step 4: Place three buttons named as add,clear,exit.

Step 5: Right click on add button->action performed.UsingString.valueOf() to add the values.

Step 6: Right click on clear button->action performed.Clear the textboxes.

Step 7: Right click on exit button->action performed.Exit from the file.

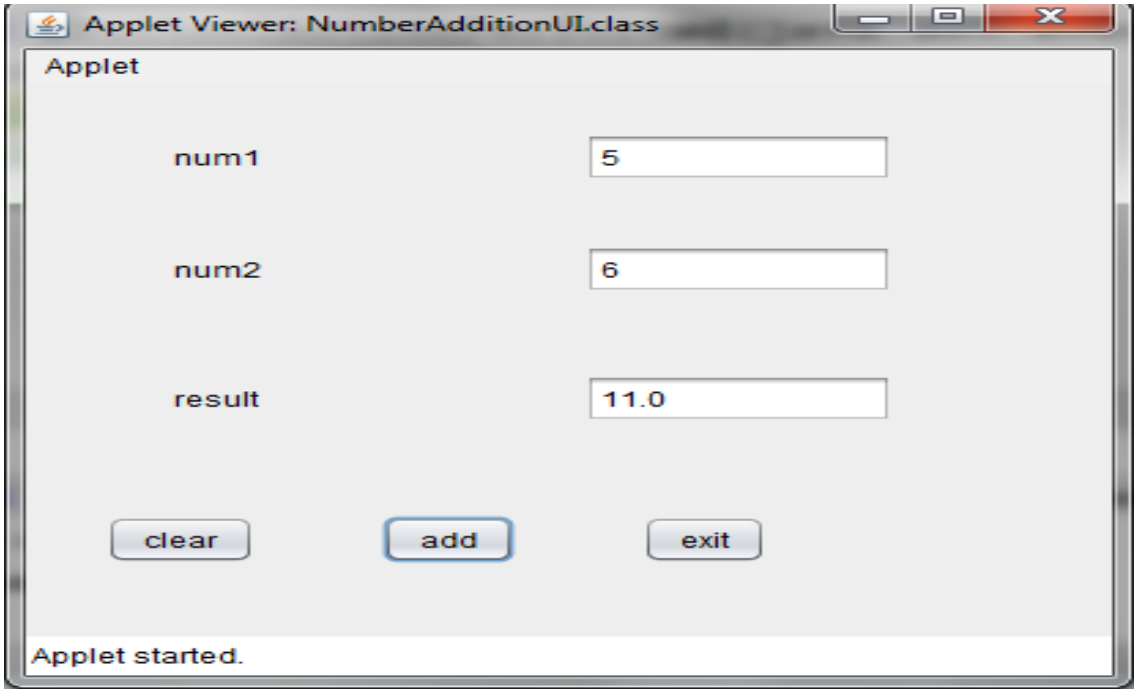
Step 8: Save the program and display the result.

Step 9: Stop the process.

CODING:

```
public class NumberAdditionUI extends javax.swing.JApplet
{
private void jButton1ActionPerformed(java.awt.event.ActionEventevt)
      {
        jTextField1.setText("");
        jTextField2.setText("");
        jTextField3.setText("");
      }
private void jButton3ActionPerformed(java.awt.event.ActionEventevt)
     {
System.exit(0);
   }
private void jButton2ActionPerformed(java.awt.event.ActionEventevt)
 {
float num1,num2,result;
      num1=Float.parseFloat(jTextField1.getText());
       num2=Float.parseFloat(jTextField2.getText());
result=num1+num2;
jTextField3.setText(String.valueOf(result));
  }
}
```

OUTPUT:



RESULT:

Thus the Applet program has been successfully executed.

Ex.No:7 **ADD CONTROLS IN BEAN**

AIM:

To write a program to add controls in bean.

ALGORITHM:

Step1: Start the process.

Step2: Open the net beans application.

Start->All programs->Netbeans.

Step3: Create new jsp page named as **index.jsp**

Step 4: Give the name as login here username and password is given using text boxes and submit button.

Step 5: Create a another html file named as **receive.jsp**

Step 6:setProperty and getProperty is used for set and get the values from the index.html File.

Step 7: Create new servlet page named as **localfile.java**

Web pages->new->servlet

Step 8: Create a class named as local file,declare the two string variable as user and pass.

Step 9:setUser() ,getUser() is used to check the username and return the name and similarly return the password.

Step 10: If(s1.equals(user)&&s2.equals(pass)) if it is true print "valid" or else "invalid".

Step 11: Stop the process

CODING:

index.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
        <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
        </head>
<body bgcolor="#dddddd">
        <h1>Using Java Beans with JSP</h1>
        <form method="get" action="receive.jsp">
                Enter user name<input type="text" name="user"><br>
                Enter password<input type="password" name="pwd"><br>
        <input type="submit">
        </form>
</body>
</html>
```

receive.jsp

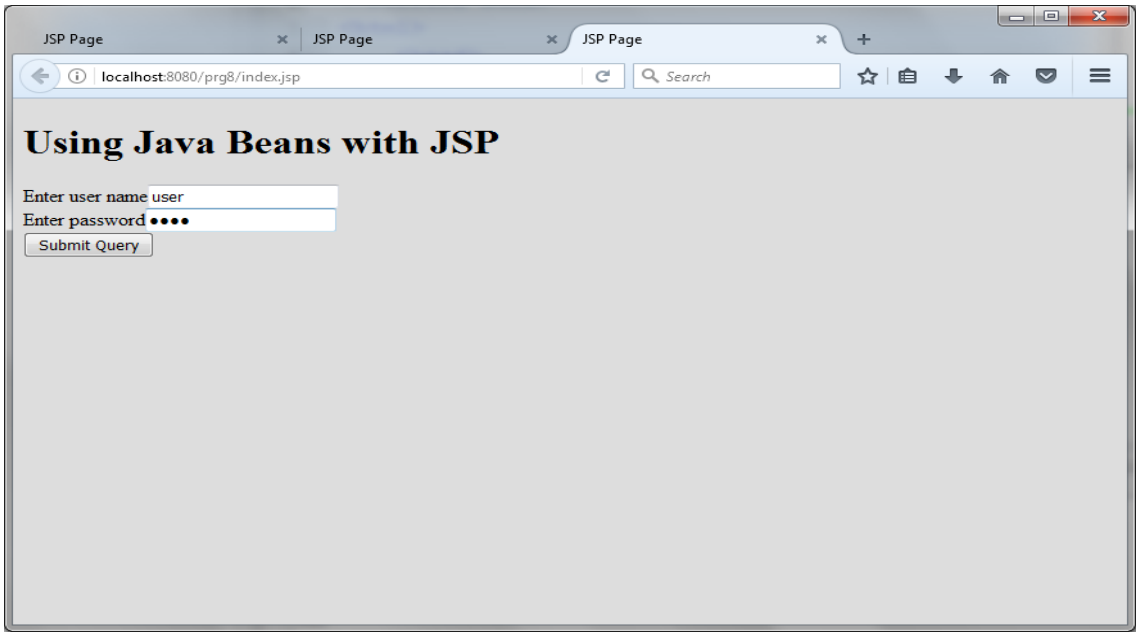
```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>JSP Page</title>
</head>
<body>
<jsp:useBean id="snr" class="pack.localfile"/>
<jsp:setProperty name="snr" property="user"/>
<jsp:setProperty name="snr" property="pass"/>
    You entered user name as <jsp:getProperty name="snr" property="user"/>
    You entered user password as <jsp:getProperty name="snr"
property="pass"/><br>
    You are a<%=snr.validate("user","user")%>user<br>
<b>Thank You</b>
</body>
</html>
```

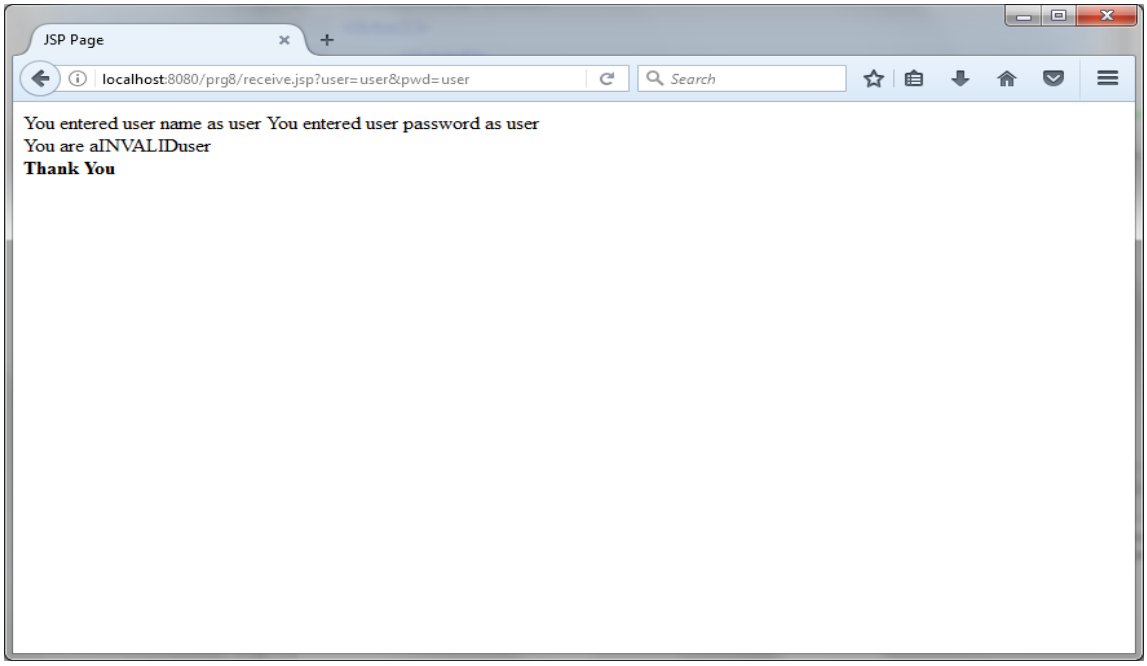
localfile.java

```
package pack;
public class localfile
{
   String user;
   String pass;
publiclocalfile()
 {
 }
public void setUser(String user)
 {
this.user=user;
 }
public String getuser()
  {
return user;
  }
public void setPass(String pass)
 {
this.pass=pass;
 }
public String getpass()
```

```
 {
return user;
    }
public String validate(String s1,String s2)
  {
if(s1.equals(user)&&s2.equals(pass))
return "VALID";
else
return "INVALID";
   }
}
```

OUTPUT:





RESULT:

Thus the Add control program has been successfully executed.

Ex.No:8 **COUNTER IN JAVABEAN**

AIM:

To design a counter in javabean.

ALGORITHM:

Step 1: Start the process.

Step 2: Open new project→Java web→Web Application.

Step 3: Create a servlet page as **counter.java** and

Initialize the **bean package** and create a class name as **counter**.

Step 4: Initialize the variable count value as "0".

Step 5: Create a jsp page as **index.jsp.**

Step 6: To increase the page hit value using **count++**in **getcount()**.

Step 7: Using **getcount()** and **setcount()** to display the counting values from jsp.

Step 8: Use a bean package specify the id as counter.

Step 9: To print the count value using following coding

Code:"+counter.getcount()+"

Step 10: Stop the process.

CODING:

index.jsp

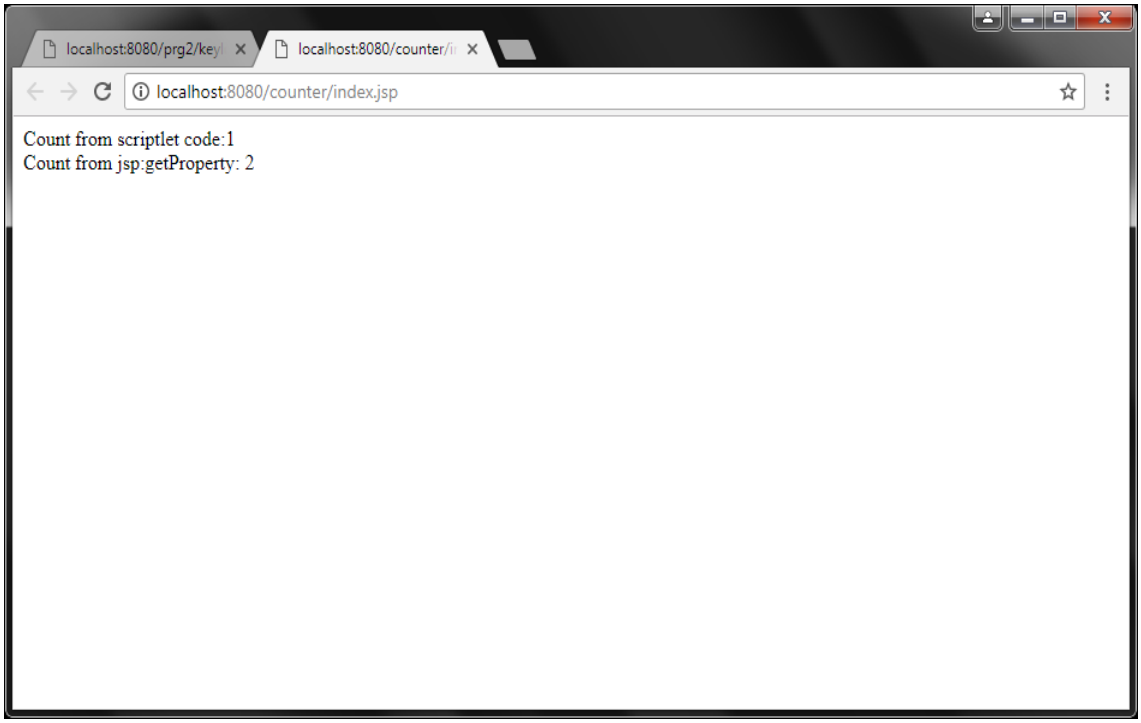
```
<html>
<head>
</head>
<body>
<%@ page language="java"%>

<jsp:useBean id="counter" scope="session" class="beans.counter"/>
<jsp:setProperty name="counter" property="count" param="count"/>
<%
out.println("Count from scriptlet code:" + counter.getCount()+"<BR>");
      %>
       Count from jsp:getProperty:
<jsp:getProperty name="counter" property="count"/><BR>
</body>
</html>
```

counter.java

```
package beans;
importjava.io.Serializable;
public class counter implements Serializable
{
```

```
int count=0;
public counter()
{
}
publicintgetCount()
{
count++;
returnthis.count;
}
public void setCount(int count)
{
this.count=count;
}
}
```

OUTPUT:



RESULT:

Thus the Counter program has been successfully executed.

Ex.No:9 **STREAM CONTENTS OF A FILE USING JSP**

AIM:

To write a program to display a file content in web browser.

ALGORITHM:

Step 1: Start the process.

Step 2: Open the net beans application.

Start->All programs->Netbeans.

Step3: Create a text file using text editior**log.txt**

Step 4: Create new jsp page named as **index.jsp**

Step 5: Usebufferreader to get the file path from the corresponding location.

Step 6: Use "read line()" function to display a content on the browser.

Step 7: Use toString() to print the file.

Step 8: Save the program and display the result.

Step 9: Stop the process.

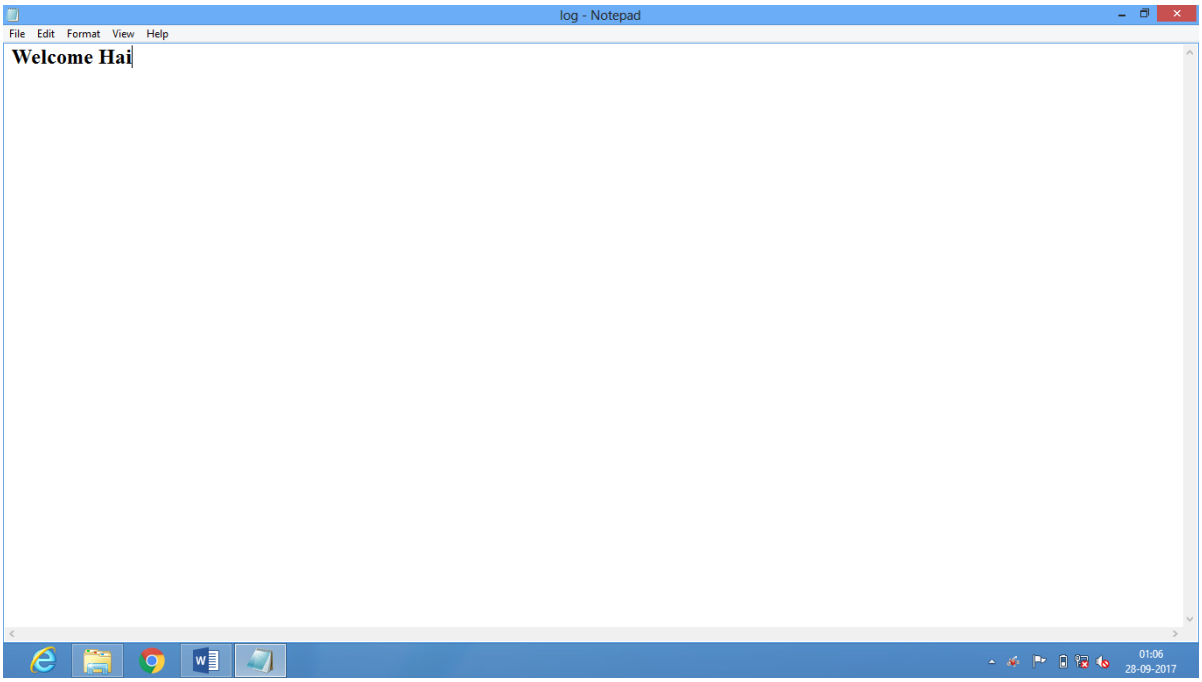
CODING:

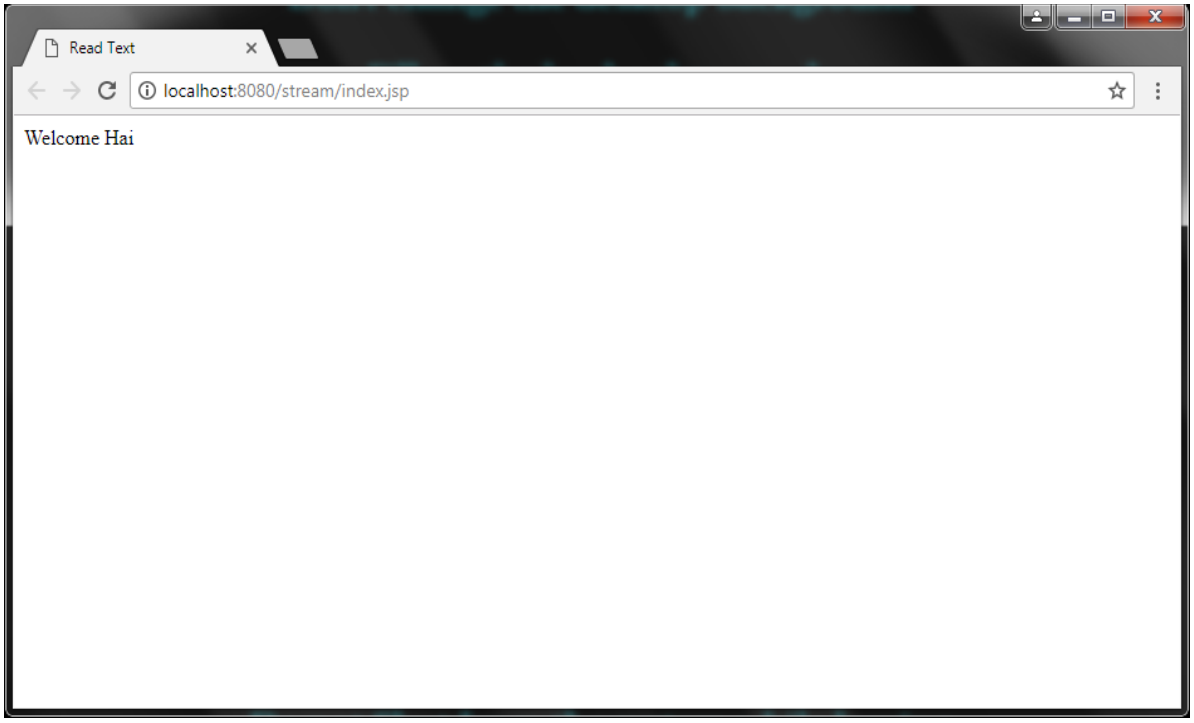
index.jsp

```
<%@page import="java.io.FileInputStream"%>
<%@page import="java.io.File"%>
<%@page import="java.io.InputStreamReader"%>
<%@page import="java.net.URL"%>
<%@page import="java.io.FileReader"%>
<%@page import="java.io.BufferedReader"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Read Text</title>
</head>
<body>
<%
        String jspPath = "C:\\log\\";
        String fileName = "log.txt";
        String txtFilePath=jspPath + fileName;
BufferedReader reader = new BufferedReader (new FileReader(txtFilePath));
        //BufferReaderbr=new InputStreamReader(new FileInputStream(txtFilePath));
StringBuildersb=new StringBuilder();
```

```
        String line;
while((line=reader.readLine())!= null){
sb.append(line+"\n");
      }
out.println(sb.toString());
        %>
</body>
</html>
```

OUTPUT:





RESULT:

Thus Stream file program has been successfully executed.

Ex.No:10 **INSERT AN APPLET INTO JSP**

AIM:

To write a program to insert applet into jsp page.

ALGORITHM:

Step1: Start the process.

Step 2: Open the net beans application.

Start→All programs→Netbeans.

Step 3: Create new jsppage named as **"index.jsp"**.

Step 4: To insert the jsp:plugin, **"jsp:fallback"** in **"index.jsp"** page

Step 5: To add jApplet, right click the web and add jApplet and specified name as **"applet1.java"**

Step 6: Declare the necessary packages, Class ButtonMoveApplet extends Applet class.

Step 7: Use **move.reshape()** to reshape the specific shape. Using setBackground() to set the background color and using setForeground() to set the foreground color.

Step 8: Save the program and display the result.

Step 9: Stop the process.

CODING:

index.jsp

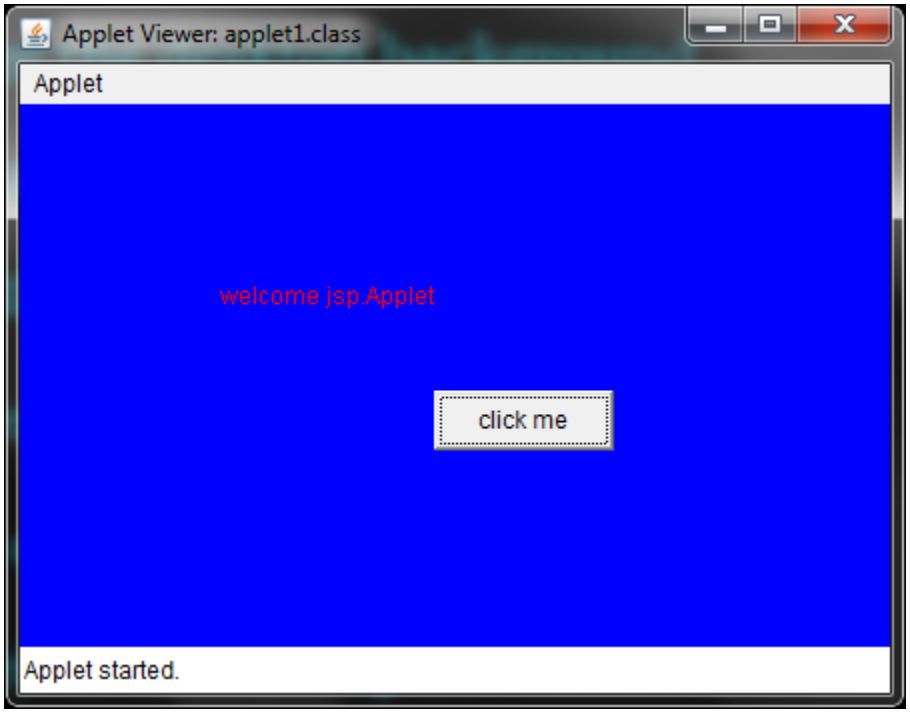
```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>JSP Page</title>
</head>
<body>
<jsp:plugin type="applet" code="applet1.class" width="400" height="400">
<jsp:fallback>
<p>Unable to load applet</p>
</jsp:fallback>
</jsp:plugin>
</body>
</html>
```

applet1.java

```
import java.io.*;
importjava.awt.*;
importjava.util.*;
importjava.applet.*;
importjava.awt.event.*;
public class applet1 extends Applet {
   Button move;
   Random r;
private String buttonLabel;
public void init()
   {
setLayout(null);
move=new Button("click me");
add(move);
move.reshape(10,10,70,30);
            r=new Random();
setBackground(Color.blue);
setForeground(Color.red);
   }
public void paint(Graphics g){
g.drawString("welcome jsp.Applet",100,100);
```

```
    }
publicboolean action(Event evt,ObjectwhatAction){
if(!(evt.targetinstanceof Button))
return false;
        String buttonLabel=(String)whatAction;
if(buttonLabel =="click me"){
move.reshape(Math.abs(r.nextInt())%(size().width-70),
Math.abs(r.nextInt())%(size().height-30),90,30);
repaint();
    }
return true;
    }
    }
```

OUTPUT:



RESULT:

Thus Inserting Applet program has been successfully executed.