

Hidden Form Field

Hidden Form Field

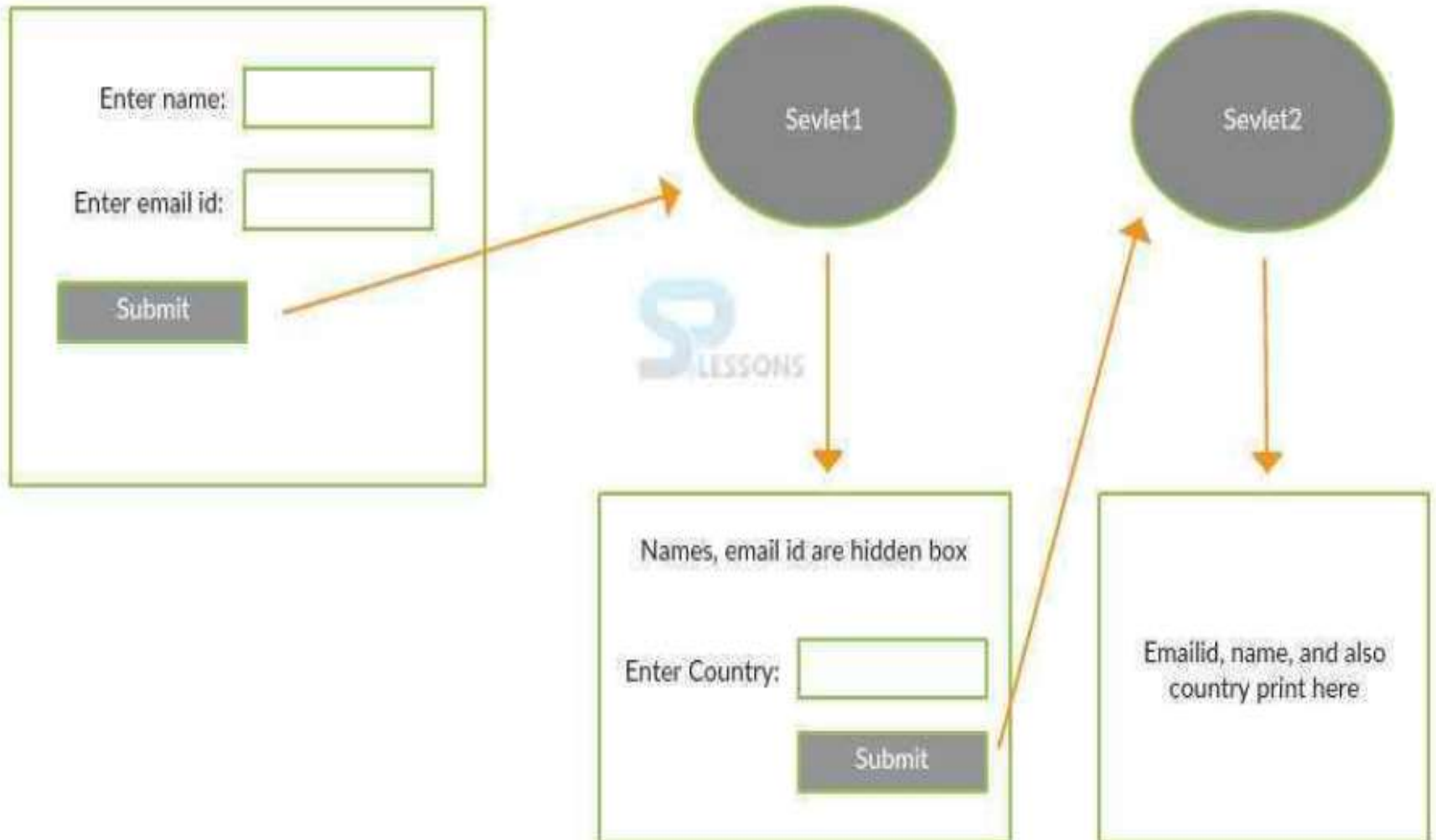
- Hidden Form Field, A hidden form field is an [HTML](#) element utilized to store the value similar to the text box, but here stored value will be invisible, so to extract that value servlet code or any server side code will be used.
- A hidden form field is a technique used to store the session information for a particular client and is one of an important **Session Tracking** Technique.

Advantages of Hidden Fields

- Hidden fields data are stored in the browser window. So they do not provide burden to the server.
- If [cookies](#) are disabled or not, it will work.

Example

Hidden Fields Process



index.html:

```
<html>
```

```
  <form action="welcome">
```

```
    Enter name:<input type="text" name="username"/>
```

```
    Enter Email:<input type="text" name="emailId"/>
```

```
    <input type="submit" value="submit"/>
```

```
  </form>
```

```
</html>
```

Sample rule of WEB.XML file as follows

```
<web-app>

  <servlet>
    <servlet-name>Demo</servlet-name>
    <servlet-class>Servlet</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>Demo</servlet-name>
    <url-pattern>/welcome</url-pattern>
  </servlet-mapping>

</web-app>
```

logical name or
servlet object

servlet class name

must match with
above logical name

must match with HTML
form action name

web.xml:

```
<web-app>
  <servlet>
    <servlet-name>DemoHiddenFields</servlet-name>
    <servlet-class>servlethiddenfield.DemoHiddenFields</servlet-class>
  </servlet>

  <servlet>
    <servlet-name>ShowAllData</servlet-name>
    <servlet-class>servlethiddenfield.ShowAllData</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>DemoHiddenFields</servlet-name>
    <url-pattern>/welcome</url-pattern>
  </servlet-mapping>

  <servlet-mapping>
    <servlet-name>ShowAllData</servlet-name>
    <url-pattern>/servlet2</url-pattern>
  </servlet-mapping>

</web-app>
```

DemoHiddenFields:

```
package servlethiddenfield;

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class DemoHiddenFields extends HttpServlet {

    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        String username = request.getParameter("username");
        String emailId = request.getParameter("emailId");
        out.println("<HTML>");
        out.println("<h2>your name is display in servlet1 :"+username+"</h2>");
        out.println("<h2>your email is display in servlet1 :"+emailId+"</h2>");
        out.println("</HTML>");

        out.print("<form action='servlet2'>");
        //username and emailid in hidden to pass servlet2
        out.print("<input type='hidden' name='username' value='"+username+"'>");
        out.print("<input type='hidden' name='emailid' value='"+emailId+"'>");

        out.print("Enter your country :<input type='text' name='country'>");

        out.print("<input type='submit' value='submit'>");
        out.print("</form>");
        out.flush();
        out.close();
    }
}
```


ShowAllData.java:

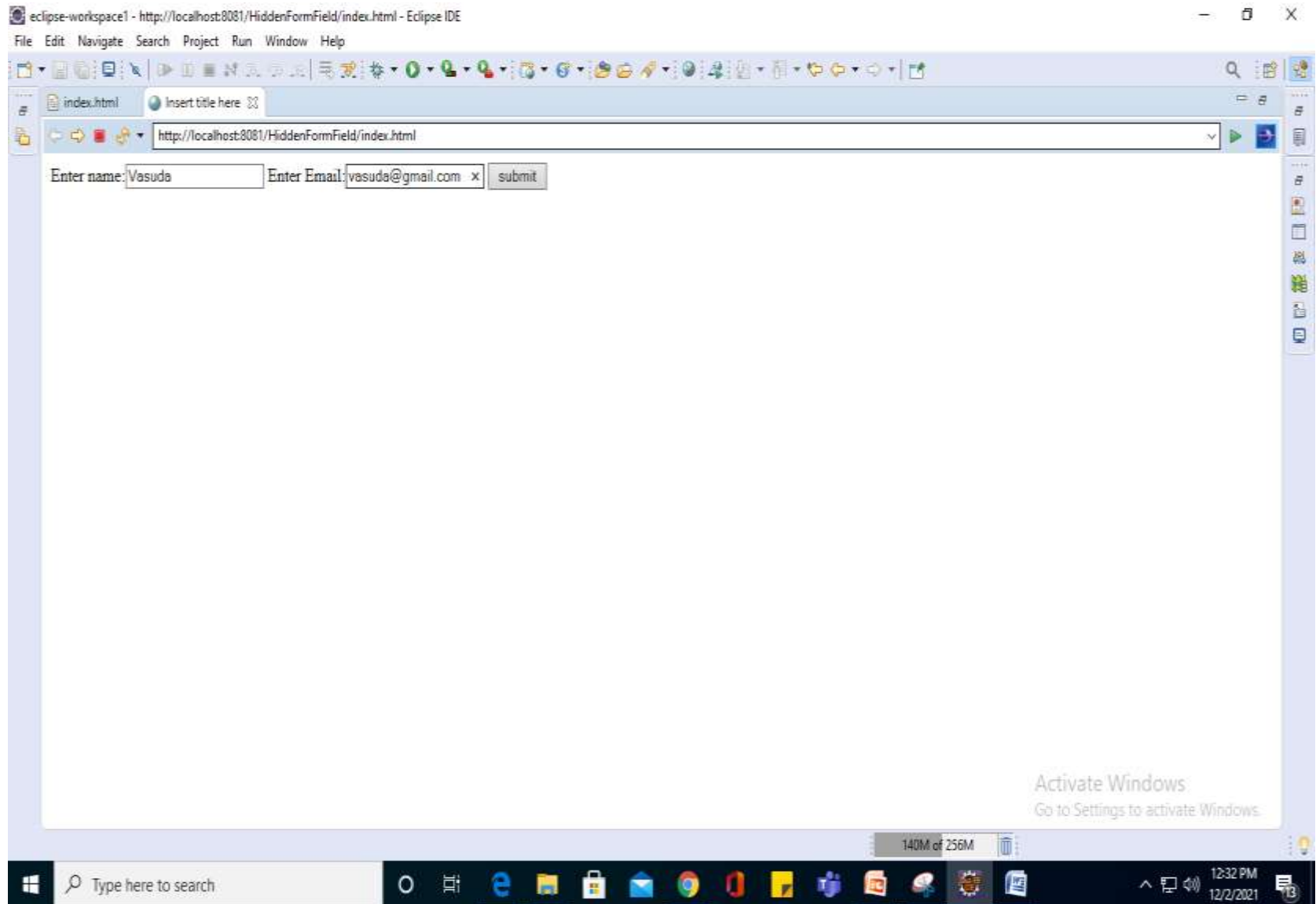
```
package servlethiddenfield;

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class ShowAllData extends HttpServlet {

    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        String username = request.getParameter("username");
        String emailId = request.getParameter("emailid");
        String country = request.getParameter("country");
        out.println("<HTML>");
        out.println("<h2>your name is display in servlet2 :"+username+"</h2>");
        out.println("<h2>your emaild is display in servlet2 :"+emailId+"</h2>");
        out.println("<h2>your country is display in servlet2 :"+country+"</h2>");
        out.println("</HTML>");
        out.flush();
        out.close();
    }
}
```



index.html http://localhost:8081/HiddenFormField/welcome?username=Vasuda&emaild=vasuda@gmail.com

http://localhost:8081/HiddenFormField/welcome?username=Vasuda&emaild=vasuda@gmail.com

your name is display in servlet1 :Vasuda

your emaild is display in servlet1 :vasuda@gmail.com

Enter your country :

Activate Windows
Go to Settings to activate Windows.

your name is display in servlet2 :Vasuda

your emaild is display in servlet2 :vasuda@gmail.com

your country is display in servlet2 :India

Activate Windows
Go to Settings to activate Windows.

Done

147M of 256M

Type here to search

12:34 PM
12/2/2021

- Hidden Form Field – Session can be defined as specific interval of time.
- If HTTP is stateless then each request will be treated as new request from the web server.
- To identify specific user session tracking will be used.

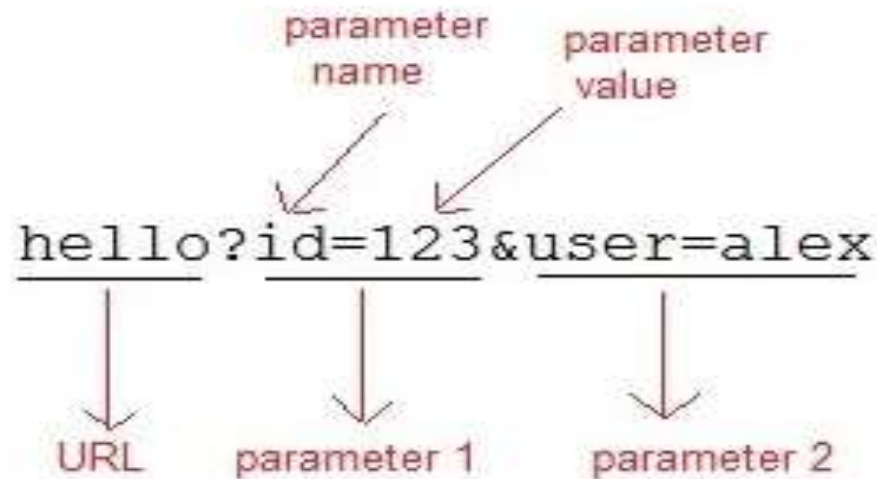
URL Rewriting

URL rewriting

If the client has disabled cookies in the browser then session management using cookie wont work. In that case **URL Rewriting** can be used as a backup. **URL rewriting** will always work.

- In URL rewriting, a token(parameter) is added at the end of the URL. The token consist of name/value pair seperated by an equal(=) sign.

For Example:



- The `getParameter()` method is used to get the parameter value at the server side.

index.html:

```
<form method="post" action="validate">
```

```
  Name:<input type="text" name="user" /><br/>
```

```
  Password:<input type="text" name="pass" ><br/>
```

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

```
</form>
```

Web.xml:

<web-app...>

<servlet>

<servlet-name>validate</servlet-name>

<servlet-class>MyServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>validate</servlet-name>

<url-pattern>/validate</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>First</servlet-name>

<servlet-class>First</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>First</servlet-name>

<url-pattern>/First</url-pattern>

</servlet-mapping>

<welcome-file-list>

<welcome-file>index.html</welcome-file>

</welcome-file-list>

</web-app>

MyServlet.java:

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class MyServlet extends HttpServlet {

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        String name = request.getParameter("user");
        String pass = request.getParameter("pass");

        if(pass.equals("1234"))
        {
            response.sendRedirect("First?user_name="+ name);
        }
    }
}
```

First.java:

```
import java.io.*;
```

```
import javax.servlet.*;
```

```
import javax.servlet.http.*;
```

```
public class First extends HttpServlet {
```

```
    protected void doGet(HttpServletRequest request,  
        HttpServletResponse response)
```

```
        throws ServletException, IOException {
```

```
        response.setContentType("text/html");
```

```
        PrintWriter out = response.getWriter();
```

```
        String user = request.getParameter("user_name");
```

```
        out.println("Welcome "+user);
```

```
    }
```

```
}
```

The screenshot shows the Eclipse IDE interface with a web browser window open. The browser address bar shows `http://localhost:8081/URLRewriting/index.html`. The browser content displays a simple web form with the following elements:

- Name:
- Password:
-

The Project Explorer on the left side of the IDE shows the following structure:

- HiddenFormField
- Servers
- ServletProgram
- ServletSendRedirect
- test
- URLRewriting
 - Deployment Descriptor: U
 - JAX-WS Web Services
 - Java Resources
 - src
 - (default package)
 - First.java
 - MyServlet.java
 - Libraries
 - JavaScript Resources
 - build
 - WebContent
 - META-INF
 - WEB-INF
 - lib
 - web.xml
 - index.html

The Servers tab at the bottom shows the following server configuration:

- Tomcat v9.0 Server at localhost [Started, Synchronized]

The status bar at the bottom indicates "Done" and "142M of 256M".

Activate Windows
Go to Settings to activate Windows.

The screenshot shows the Eclipse IDE interface. The top toolbar contains various icons for file operations, navigation, and development. The Project Explorer on the left displays the project structure for 'URLRewriting', including 'src' (with 'First.java' and 'MyServlet.java'), 'Libraries', 'JavaScript Resources', 'build', 'WebContent' (with 'META-INF', 'WEB-INF', and 'lib'), and 'index.html'. The main editor area shows a web browser window with the URL 'http://localhost:8081/URLRewriting/First?user_name=Vasuda'. The browser displays 'Welcome Vasuda'. The bottom status bar shows 'Tomcat v9.0 Server at localhost [Started, Synchronized]'.

Project Explorer

- HiddenFormField
- Servers
- ServletProgram
- ServletSendRedirect
- test
- URLRewriting
 - Deployment Descriptor: U
 - JAX-WS Web Services
 - Java Resources
 - src
 - (default package)
 - First.java
 - MyServlet.java
 - Libraries
 - JavaScript Resources
 - build
 - WebContent
 - META-INF
 - WEB-INF
 - lib
 - web.xml
 - index.html

index.html MyServlet.java First.java web.xml http://localhost:8081/URLRewriting/First?user_name=Vasuda

http://localhost:8081/URLRewriting/First?user_name=Vasuda

Welcome Vasuda

Markers Properties Servers Data Source Explorer Snippets Console

Tomcat v9.0 Server at localhost [Started, Synchronized]

Done

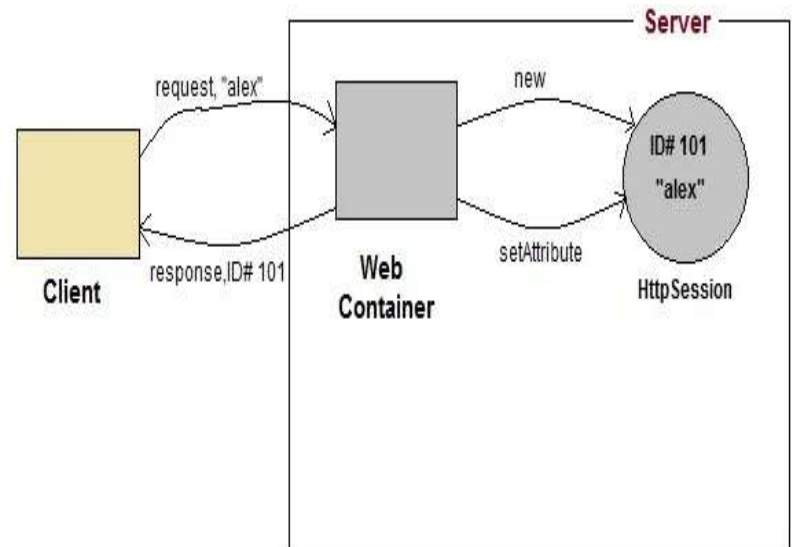
144M of 256M

HttpSession

- **HttpSession** object is used to store entire session with a specific client. We can store, retrieve and remove attribute from **HttpSession** object. Any servlet can have access to **HttpSession** object throughout the `getSession()` method of the **HttpServletRequest** object.

How HttpSession works?

- On client's first request, the **Web Container** generates a unique session ID and gives it back to the client with response. This is a temporary session created by web container.
- The client sends back the session ID with each request. Making it easier for the web container to identify where the request is coming from.
- The **Web Container** uses this ID, finds the matching session with the ID and associates the session with the request.



Index.html:

```
<!DOCTYPE html >
```

```
<HTML>
```

```
<HEAD>
```

```
</HEAD>
```

```
<BODY>
```

```
<FORM METHOD=post ACTION="serv1">
```

```
  UserName<input type=text name=username /><br>
```

```
  Age<input type=text name=age /><br>
```

```
<input type=submit name=continue value=continue />
```

```
</FORM>
```

```
</BODY>
```

```
</HTML>
```

SessionServlet.java:

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*; @WebServlet("/serv1")
public class SessionServlet extends HttpServlet {
    public void doPost(HttpServletRequest req,HttpServletResponse resp) throws
        ServletException,IOException
    {
        String uname, age;
        uname=req.getParameter("uname");
        age=req.getParameter("age");

        //Create cookies and send them as response
        HttpSession s= req.getSession(true);
        s.setAttribute("uname",uname);
        s.setAttribute("age",age);

        PrintWriter out=resp.getWriter();
        resp.setContentType("text/html");
        /*code to display other form fields*/
        out.println("<HTML><BODY>");
        out.println("<FORM METHOD=post ACTION="+resp.encodeURL("serv2")+ ">");
        out.println("qualification<input type=text name=qual /><br>");
        out.println("company<input type=text name=comp /><br>");
        out.println("<input type=submit name=submit value=submit /> ");
        out.println("</FORM></BODY></HTML>");
    }
}
```

Result.java

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
@WebServlet("/serv2")
public class Result extends HttpServlet {
    public void doPost(HttpServletRequest req,HttpServletResponse resp) throws
ServletException,IOException
    {
        //Retrieve data from form String qual,comp; qual=req.getParameter("qual");
        comp=req.getParameter("comp");
        PrintWriter out=resp.getWriter();
        resp.setContentType("text/html");
        //Retrieve data from cookies
        HttpSession s=req.getSession(true);

        /*code to display information entered*/ out.println("<HTML><BODY>");
        out.println("The information provided by you is:");
        out.println("Username"+":"+s.getAttribute("uname")+"<br>");
        out.println("Age"+":"+s.getAttribute("age")+"<br>");
        out.println("Qualification:"+qual+"<br>");
        out.println("Company:"+comp+"<br>");
        out.println("Thank you for subscribing.." +s.getAttribute("uname"));
        out.println("</BODY></HTML>");
    }
}
```

UserName kiran

Age 25

