Week-8: **Date:**

AIM:

User Authentication:

Assume four users user1, user2, user3 and user4 having the passwords pwd1, pwd2, pwd3 and pwd4 respectively. Write a servelet for doing the following.

1. Create a Cookie and add these four user id's and passwords to this Cookie.
2. Read the user id and passwords entered in the Login form (week1) and authenticate with the values (user id and passwords) available in the cookies.

If he is a valid user (i.e., user-name and password match) you should welcome him by name (user-name) else you should display "You are not an authenticated user ". Use init-parameters to do this. Store the user-names and passwords in the web.xml and access them in the servlet by using the getInitParameters() method.

DESCRIPTION:

Servlet Life cycle:

1. *Servlet class loading*
2. *Servlet Instantiation*
3. *call the init method*
4. *call the service method*
5. *call destroy method*

Class loading and instantiation

If you consider a servlet to be just like any other Java program, except that it runs within a servlet container, there has to be a process of loading the class and making it ready for requests. Servlets do not have the exact equivalent of a main method that causes them to start execution.

When a web container starts it searches for the deployment descriptor (WEB.XML) for each of its web applications. When it finds a servlet in the descriptor it will create an instance of the servlet class. At this point the class is considered to be loaded (but not initialized).

The init method

The HttpServlet class inherits the init method from GenericServlet. The init method performs a role slightly similar to a constructor in an "ordinary" Java program in that it allows initialization of an instance at start up. It is called automatically by the servlet container and as it causes the application context (WEB.XML) to be parsed and any initialization will be performed. It comes in two versions, one with a zero parameter constructor and one that takes a ServletConfig parameter.

The servlet engine creates a request object and a response object. The servlet engine invokes the servlet service() method, passing the request and response objects. Once the init method returns the servlet is said to be placed into service. The process of using init to initialize servlets means that it is

possible to change configuration details by modifying the deployment descriptor without having them hard coded in with your Java source and needing a re-compilation.

void init(ServletConfig sc)

Calling the service method

The service() method gets information about the request from the request object, processes the request, and uses methods of the response object to create the client response. The service method can invoke other methods to process the request, such as doGet(), doPost(), or methods you write. The service method is called for each request processed and is not normally overridden by the programmer.

The code that makes a servlet “go” is the. servlet

void service(ServletRequest req,ServletResponse res)

The destroy Method

Two typical reasons for the destroy method being called are if the container is shutting down or if the container is low on resources. This can happen when the container keeps a pool of instances of servlets to ensure adequate performance. If no requests have come in for a particular servlet for a while it may destroy it to ensure resources are available for the servlets that are being requested. The destroy method is called only once, before a servlet is unloaded and thus you cannot be certain when and if it is called.

void destroy()

ServletConfig Class

`ServletConfig` object is used by the Servlet Container to pass information to the Servlet during it's initialization. Servlet can obtain information regarding initialization parameters and their values using different methods of `ServletConfig` class initialization parameters are name/value pairs used to provide basic information to the Servlet during it's initialization like JDBC driver name, path to database, username, password etc.

Methods of ServletConfig class

Following are the four methods of this class :

- **getInitParameter(String paramName)** Returns value of the given parameter. If value of parameter could not be found in web.xml file then a `null` value is returned.
- **getInitParameterNames()** Returns an `Enumeration` object containing all the names of initialization parameters provided for this Servlet.
- **getServletContext()** Returns reference to the `ServletContext` object for this Servlet. It is similar to `getServletContext()` method provided by `HttpServlet` class.
- **getServletName()** Returns name of the Servlet as provided in the web.xml file or if none is provided then returns complete class path to the Servlet.

PROGRAM:

cologin.html:

```
<html>
<head>
<title> login Page </title>
<p style= "background:yellow; top:100px; left:250px; position:absolute; ">
</head>
<body>
<form  ACTION="clogin">
<label> Login </label>
<input type="text" name="usr" size="20"> <br> <br>
<label> Password </label>
<input type="password" name="pwd" size="20"> <br> <br>
<input type="submit" value="submit">
</form>
</body>
</html>
```

cologin1.html

```
<html>
<head>
<title> login Page </title>
<p style= "background:yellow; top:100px; left:250px; position:absolute; ">
</head>
<body>
<form  ACTION="clogin1">
<label> Login </label>
<input type="text" name="usr" size="20"> <br> <br>
<label> Password </label>
<input type="password" name="pwd" size="20"> <br> <br>
<input type="submit" value="submit">
</form>
</body>
</html>
```

Addcook.java:

```
import javax.servlet.* ;
import javax.servlet.http.*;
import java.io.*;
public class Addcook extends HttpServlet
```

```
{
String user,pas;
public void service(HttpServletRequest req,HttpServletResponse res) throws ServletException,IOException
{
res.setContentType("text/html");
PrintWriter out=res.getWriter();
Cookie c1=new Cookie("usr1","suni");
Cookie p1=new Cookie("pwd1","ani");
Cookie c2=new Cookie("usr2","abc");
Cookie p2=new Cookie("pwd2","123");
Cookie c3=new Cookie("usr3","def");
Cookie p3=new Cookie("pwd3","456");
Cookie c4=new Cookie("usr4","mno");
Cookie p4=new Cookie("pwd4","789");
res.addCookie(c1);
res.addCookie(p1);
res.addCookie(c2);
res.addCookie(p2);
res.addCookie(c3);
res.addCookie(p3);
res.addCookie(c4);
res.addCookie(p4);
out.println("COOKIE ADDED");
}
}
```

Clogin.java:

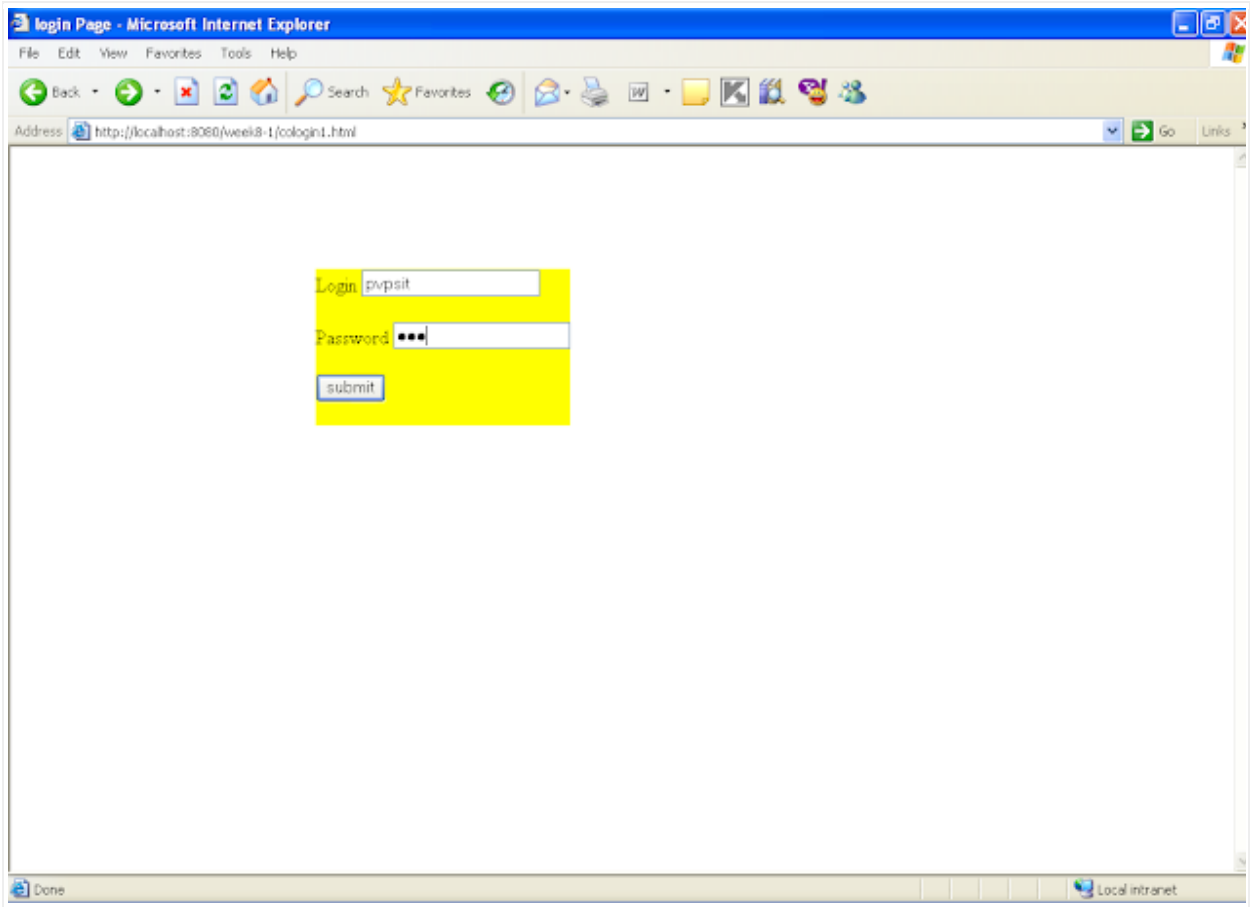
```
import javax.servlet.* ;
import javax.servlet.http.*;
import java.io.*;
public class Clogin extends HttpServlet
{
String user,pas;
public void service(HttpServletRequest req,HttpServletResponse res) throws ServletException,IOException
{
res.setContentType("text/html");
PrintWriter out=res.getWriter();
user=req.getParameter("usr");
pas=req.getParameter("pwd");
Cookie[] c=req.getCookies();
for(int i=0;i<c.length;i++)
```

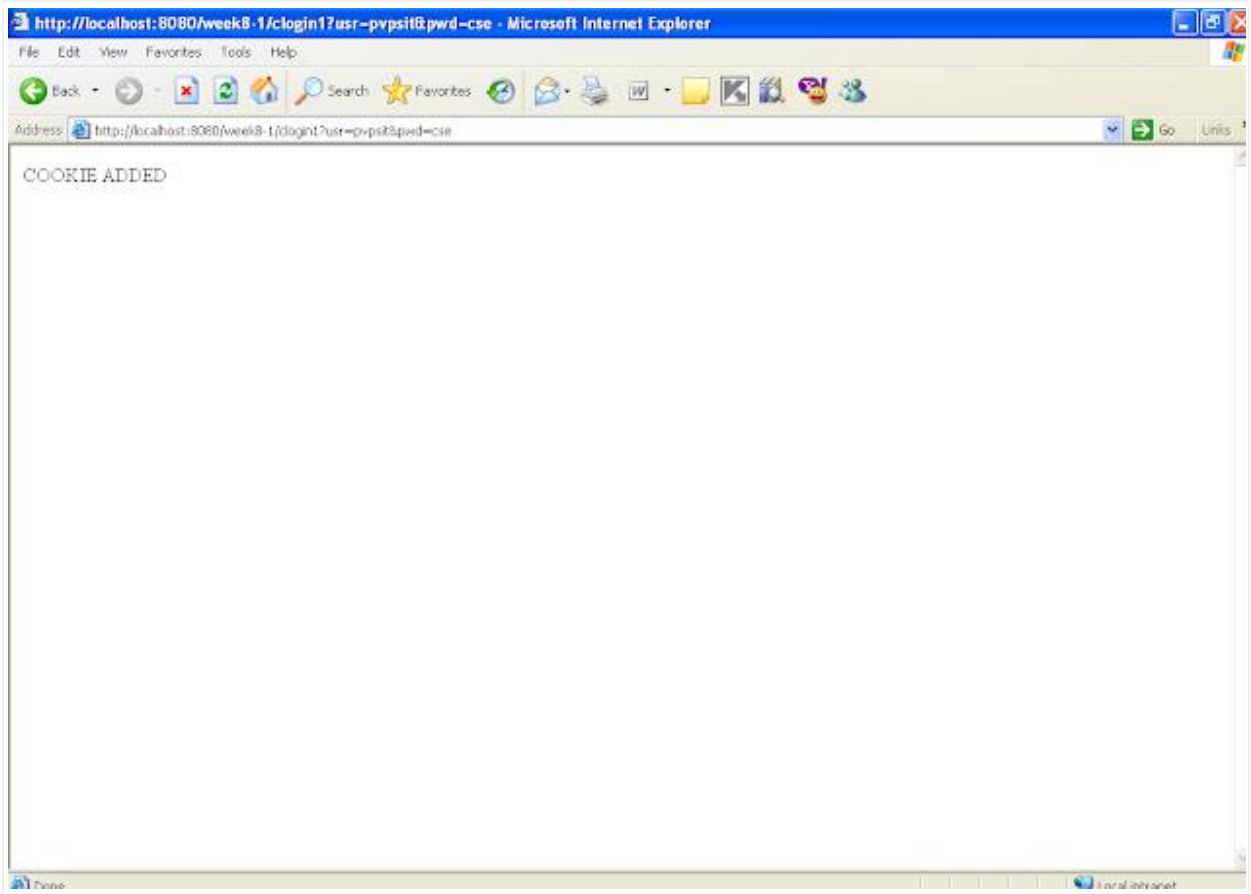
```
{
if((c[i].getName().equals("usr1")&&c[i+1].getName().equals("pwd1"))|| c[i].getName().equals("usr2")
&&c[i+1].getName().equals("pwd2"))||(c[i].getName().equals("usr3")&&
c[i+1].getName().equals("pwd3"))||(c[i].getName().equals("usr4")&& c[i+1].getName().equals("pwd4") ))
                {
                    if((user.equals(c[i].getValue()) && pas.equals(c[i+1].getValue())) )
                {
                //RequestDispatcher rd=req.getRequestDispatcher("/cart.html");
                rd.forward(req,res);
                }
                else
                {
                out.println("YOU ARE NOT AUTHORISED USER ");
                //res.sendRedirect("/cookdemo/cologin.html");
                }
                }
}
}
}
```

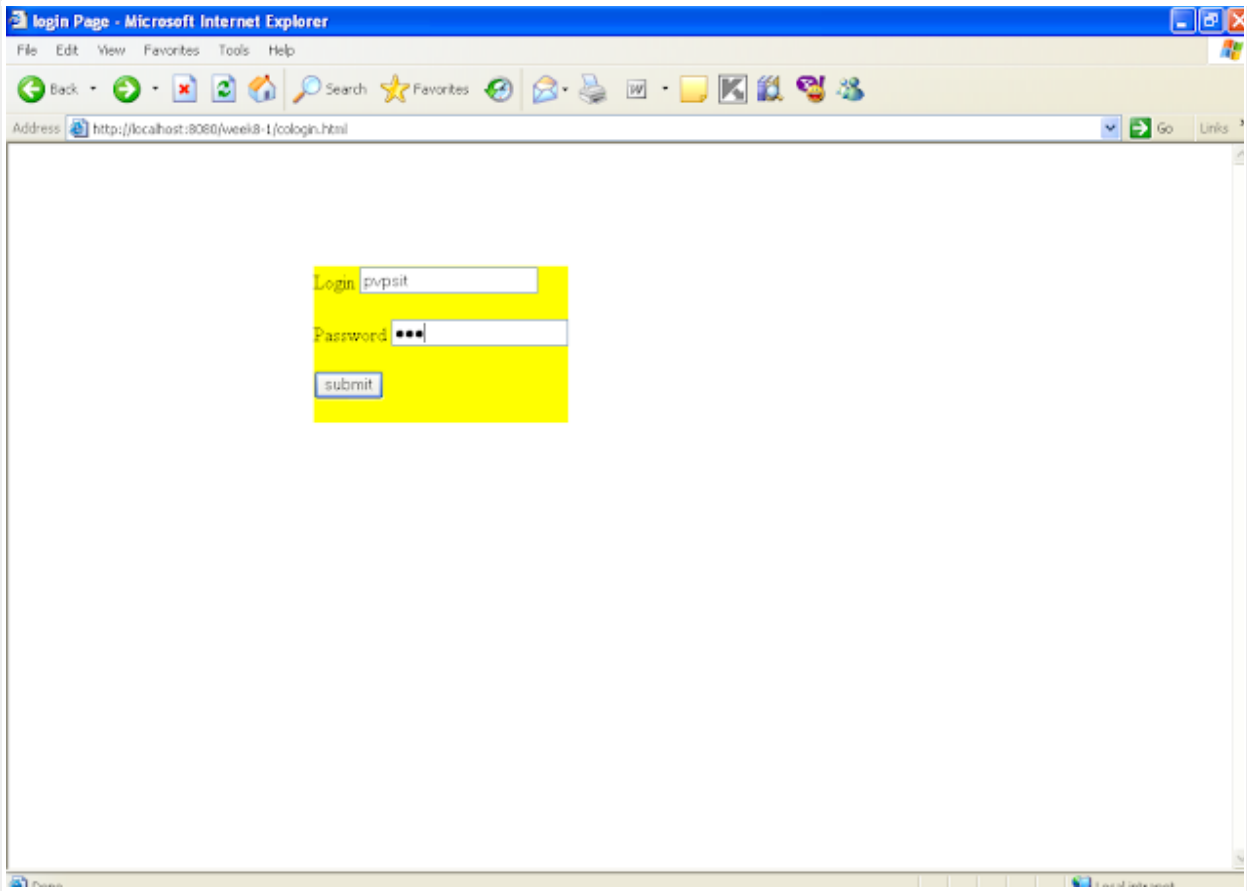
Web.xml:

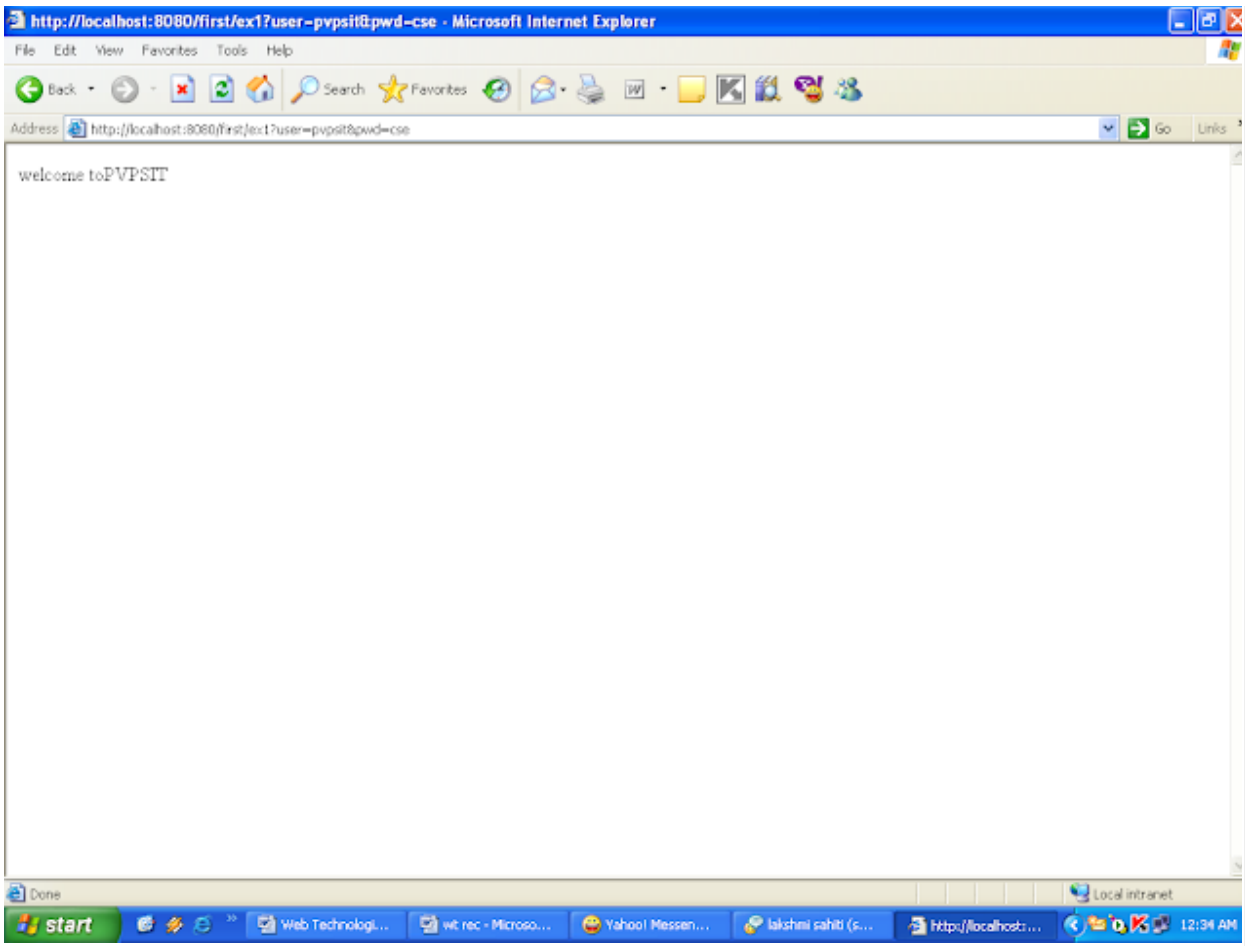
```
<web-app>
<servlet>
<servlet-name>him</servlet-name>
<servlet-class>Clogin</servlet-class>
</servlet>
<servlet>
<servlet-name>him1</servlet-name>
<servlet-class>Addcook</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>him</servlet-name>
<url-pattern>/clogin</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>him1</servlet-name>
<url-pattern>/clogin1</url-pattern>
</servlet-mapping>
</web-app>
```

OUTPUT:


login Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:8080/week8-1/cologin1.html
Go Links
Login pvpsit
Password
submit
Done
Local intranet


http://localhost:8080/week8-1/clogin1?usr=pvpsit&pwd=cse - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:8080/week8-1/clogin1?usr=pvpsit&pwd=cse
Go
Links
COOKIE ADDED


login Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:8080/week8-1/cologin.html
Go Links
Login pvpsit
Password
submit



.

2. Read the user id and passwords entered in the Login form (week1) and authenticate with the values (user id and passwords) available in the cookies.

If he is a valid user (i.e., user-name and password match) you should welcome him by name (user-name) else you should display “You are not an authenticated user “.

Use init-parameters to do this. Store the user-names and passwords in the webinf.xml and access them in the servlet by using the getInitParameters() method.

home.html:

```
<html>
<head>
  <title>Authentication</title>
</head>
<body>
<form action="ex1">
<label>Username </label>
<input type="text"size="20" name="user"><br><br>
password<input type="text" size="20" name="pwd"><br><br>
```

```
<input type="submit" value="submit">
</form>
</body>
</html>
```

Example1.java

```
import javax.servlet.*;
import java.io.*;
public class Example1 extends GenericServlet
{
  private String user1,pwd1,user2,pwd2,user3,pwd3,user4,pwd4,user5,pwd5;
  public void init(ServletConfig sc)
  {
    user1=sc.getInitParameter("username1");
    pwd1=sc.getInitParameter("password1");

    user2=sc.getInitParameter("username2");
    pwd2=sc.getInitParameter("password2");

        user3=sc.getInitParameter("username3");
    pwd3=sc.getInitParameter("password3");

        user4=sc.getInitParameter("username4");
    pwd4=sc.getInitParameter("password4");
  }
  Public void service(ServletRequest req,ServletResponse res)throws ServletException,IOException
 {
     res.setContentType("text/html");
     PrintWriter out=res.getWriter();
     user5=req.getParameter("user");
     pwd5=req.getParameter("pwd");
    if((user5.equals(user1)&&pwd5.equals(pwd1))||(user5.equals(user2)&&pwd5.equals(pwd2))||(user5.equals(user3)&&p
wd5.equals(pwd3))||(user5.equals(user4)&&pwd5.equals(pwd4)))
       out.println("<p> welcome to"+user5.toUpperCase());
     else
       out.println("You are not authorized user");
 }
}
```

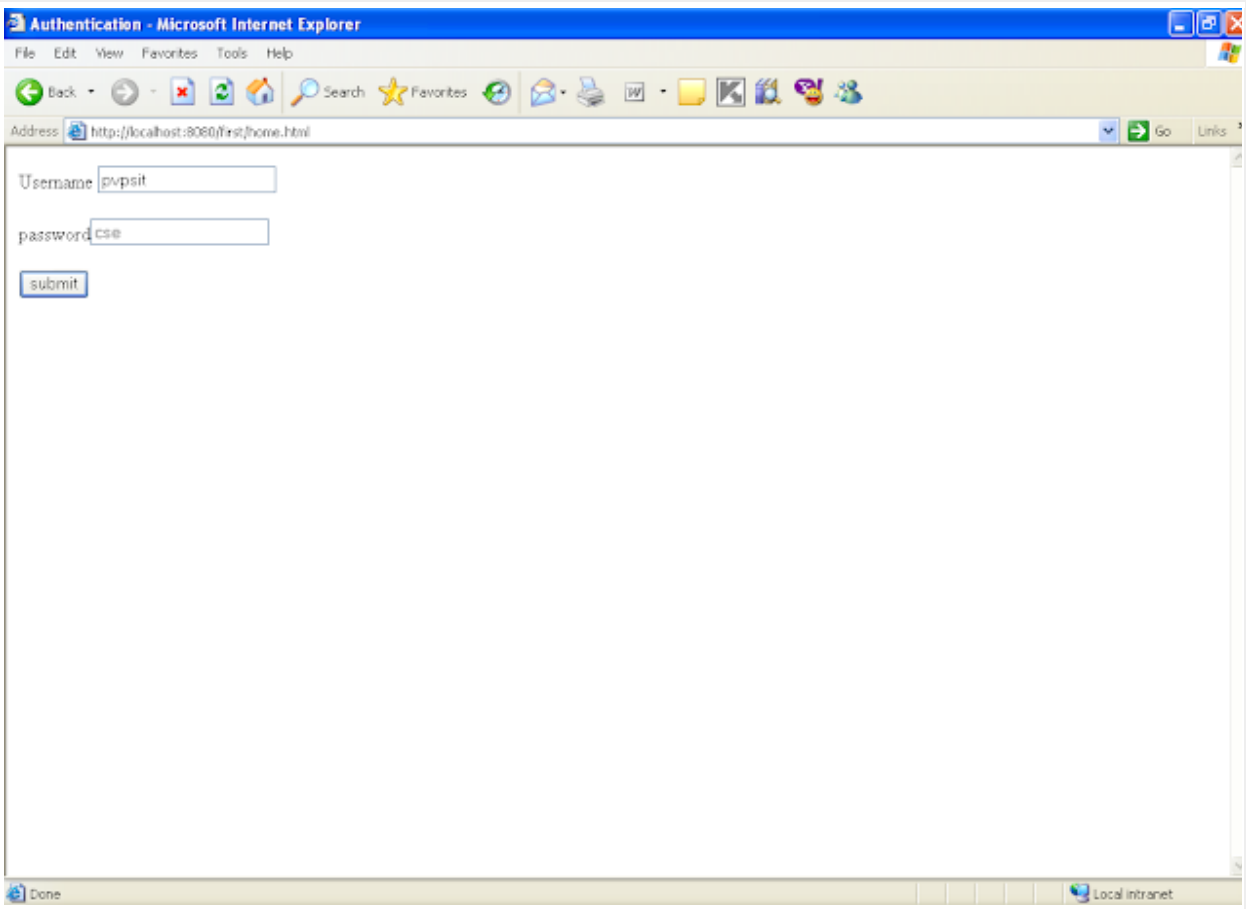
web.xml:

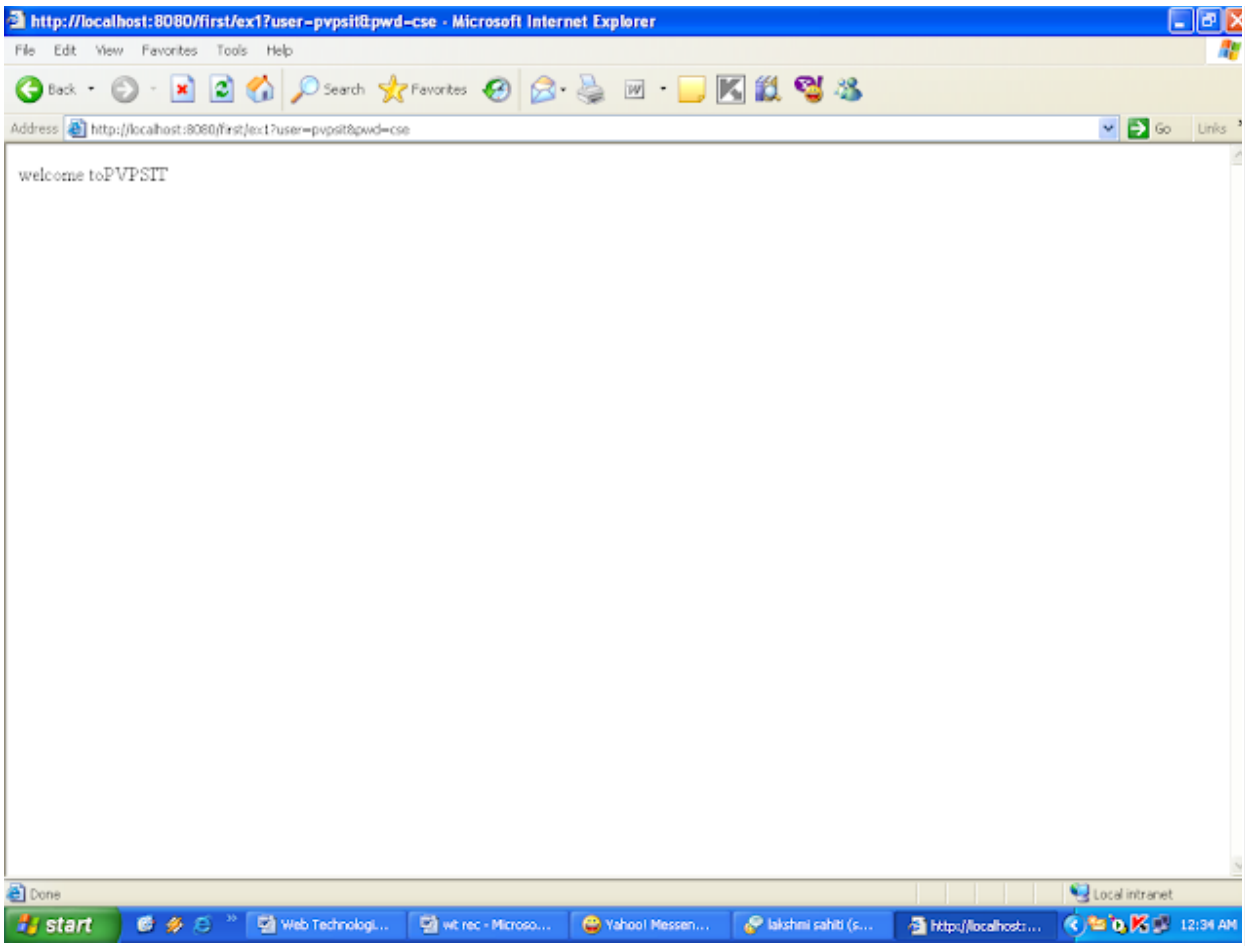
```
<web-app>
```

```
<servlet>
<servlet-name>Example</servlet-name>
<servlet-class>Example1</servlet-class>
<init-param>
<param-name>username1</param-name>
<param-value>pvpsit</param-value>
</init-param>
<init-param>
<param-name>password1</param-name>
<param-value>cse</param-value>
</init-param>
<init-param>
<param-name>username2</param-name>
<param-value>1234</param-value>
</init-param>
<init-param>
<param-name>password2</param-name>
<param-value>4567</param-value>
</init-param>
<init-param>
<param-name>username3</param-name>
<param-value>cse</param-value>
</init-param>
<init-param>
<param-name>password3</param-name>
<param-value>pvpsit</param-value>
</init-param>
<init-param>
<param-name>username4</param-name>
<param-value>wt</param-value>
</init-param>
<init-param>
<param-name>password4</param-name>
<param-value>lab</param-value>
</init-param>
</servlet>
<servlet-mapping>
<servlet-name>Example</servlet-name>
<url-pattern>/ex1</url-pattern>
</servlet-mapping>
```

</web-app>

OUTPUT:





.

RESULT:

Thus the user authentication is carried out for four users by using both cookies and getInitParameters successfully.