

Ex No: 5

**JAVA SERVLET PROGRAMS FOR INVOKING SERVLET
FROM HTML FORM AND SESSION TRACKING**

DATE: 13.09.2021

AIM

To implement the Program to invoke servlets from HTML Form

ALGORITHM

- Create a new web project.
- Create Java application logic that represents the Java servlet.
- Create the web page for the web application.
- Package the web application to a WAR file.
- Deploy the WAR file to the J2EE application server.
- Test your web application.

CONCEPTS INVOLVED

JAVA SERVLETS

Java Servlets are programs that run on a Web or Application server and act as a middle layer between a requests coming from a Web browser or other HTTP client and databases or applications on the HTTP server.

Using Servlets, you can collect input from users through web page forms, present records from a database or another source, and create web pages dynamically.

Java Servlets often serve the same purpose as programs implemented using the Common Gateway Interface (CGI). But Servlets offer several advantages in comparison with the CGI.

- Performance is significantly better.
- Servlets execute within the address space of a Web server. It is not necessary to create a separate process to handle each client request.
- Servlets are platform-independent because they are written in Java.
- Java security manager on the server enforces a set of restrictions to protect the resources on a server machine. So servlets are trusted.
- The full functionality of the Java class libraries is available to a servlet. It can communicate with applets, databases, or other software via the sockets and RMI mechanisms that you have seen already.

PROGRAM

Index.html

```
<!DOCTYPE html>
```

```
<!--
```

To change this license header, choose License Headers in Project Properties.

To change this template file, choose Tools | Templates and open the template in the editor.

```
-->
```

```
<html>
```

```
    <head>
```

```
        <title>Servlet</title>
```

```
        <meta charset="UTF-8">
```

```
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
    </head>
```

```
    <body>
```

```
        <div>
```

```
            <h1>SERVLET </h1>
```

```
        </div>
```

```
        <form name="f1" action="http://localhost:8080/WebApplication/Authentication"
        method="get">
```

```
            <label> User Name:</label>
```

```
            <input type="text" name="username" id="username"><br>
```

```
            <label>Password:</label>
```

```
            <input type="password" name="password" id="password"><br>
```

```
            <button>Login</button>
```

```
</form>

</body>

</html>
```

SERVLET PROGRAM

```
/*
 * To change this license header, choose License Headers in Project
 * Properties. * To change this template file, choose Tools | Templates * and
 * open the template in the editor.
 */
import java.io.IOException; import
java.io.PrintWriter; import
javax.servlet.ServletException; import
javax.servlet.annotation.WebServlet; import
javax.servlet.http.HttpServlet; import
javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet(urlPatterns = {"/Authentication"})
public class Authentication extends HttpServlet
{
    /**
     * Processes requests for both HTTP <code>GET</code> and <code>POST</code> *
     * methods.
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse
    response) throws ServletException, IOException
    { response.setContentType("text/html;charset=UTF-8"); try
        (PrintWriter out = response.getWriter())
        { out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet Authentication</title>");
            out.println("</head>"); out.println("<body>"); out.println("<h1>Servlet
            Authentication at " + request.getContextPath() + "</h1>");
```

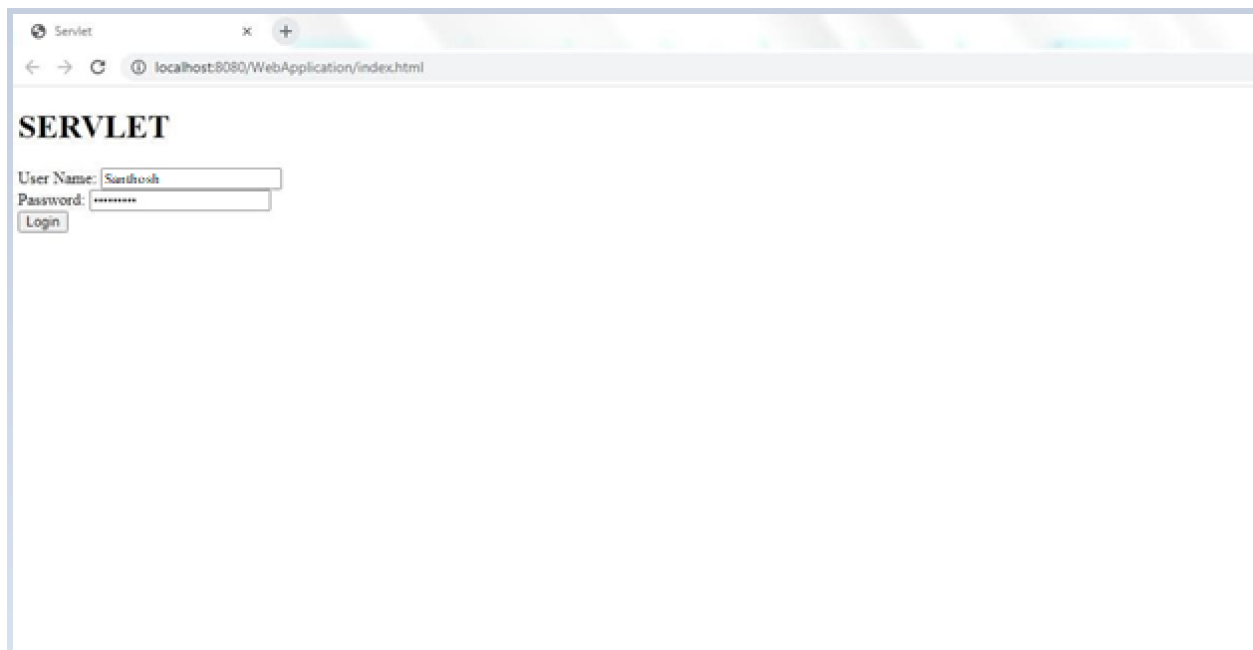
```
String username=request.getParameter("username");
String password=request.getParameter("password");
if(username==" " && password==" ")
{ out.println("User name or Password is empty!please enter");
}
else
{ out.println("Welcome "+username);
}
if(password.length()>=8)
{ out.println("Password is Strong");
} else out.println("Password is weak");
out.println("</body>");
out.println("</html>");
}
}

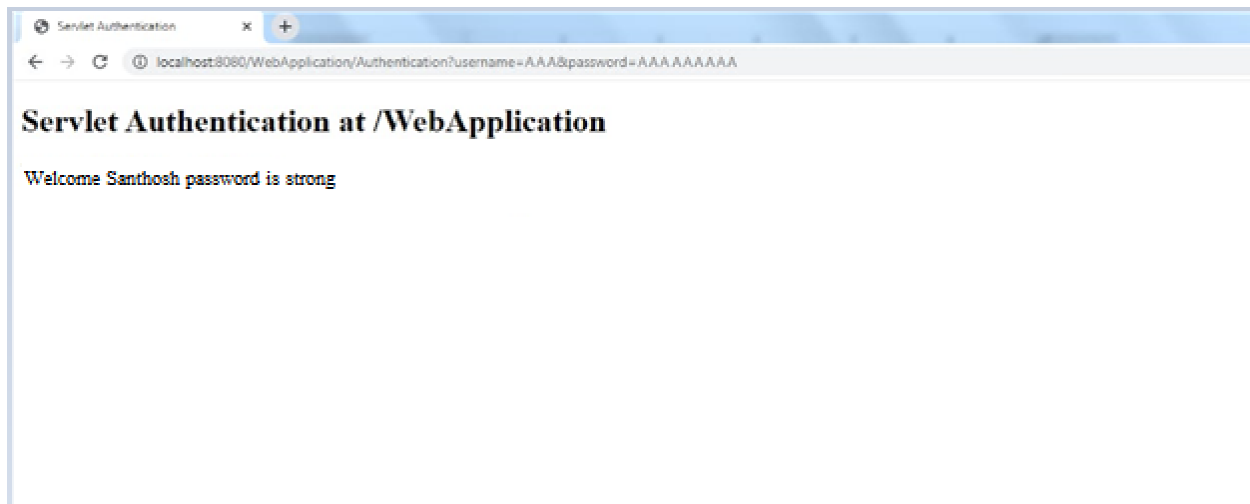
/**
 * Handles the HTTP <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException
{ processRequest(request, response);
}

/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException
{ processRequest(request, response);
```

```
    }  
    /**  
    * Returns a short description of the servlet.  
    * @return a String containing servlet description  
    */  
    @Override  
    public String getServletInfo() {  
        return "Short description";  
    } // </editor-fold>  
}
```

OUTPUT:





RESULT:

In this experiment, the invocation of servlet from HTML form has been developed and the output was verified successfully.