

ANYA M

231501018

LAB EXPERIMENT : 06

Write a Servlet to demonstrate session tracking using HttpSession.
Implement a simple login system
where the user's session is tracked.

AIM :

To develop a Java Servlet-based web application that demonstrates session tracking using HttpSession, by implementing a simple login system where user authentication status is maintained across multiple HTTP requests.

Algorithm:

Step 1: Start

Step 2: Display the login page (login.html)

- Prompt the user to enter username and password.

Step 3: Submit the form to LoginServlet using POST method.

Step 4: In LoginServlet

- Read input username and password from the request.
- Check if the credentials match predefined values (e.g., admin / 1234).
 - o If valid:
 - ❑ Create a new HttpSession object.
 - ❑ Store the username in the session.

❑ Redirect the user to WelcomeServlet.

o If invalid:

❑ Display an error message.

❑ Provide a link to retry login.

Step 5: In WelcomeServlet

- Retrieve the current HttpSession (do not create a new one).
- Check if the session exists and has a valid username attribute.

o If valid:

❑ Display a welcome message and a logout link.

o If not valid:

❑ Redirect or display a message saying the user is not logged in.

❑ Provide a login link.

Step 6: In LogoutServlet

- Retrieve the existing session.
- Invalidate the session to log the user out.
- Display a logout confirmation message and a link to login again.

Step 7: End

Code :

1. login.html

A basic HTML login form.

html

CopyEdit

```
<!DOCTYPE html>
<html>
<head>
    <title>Login Page</title>
</head>
<body>
    <h2>Login</h2>
    <form method="post" action="LoginServlet">

        Username: <input type="text" name="username"
required><br><br>
        Password: <input type="password" name="password"
required><br><br>
        <input type="submit" value="Login">
    </form>
</body>
</html>
```

2. LoginServlet.java :

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

public class LoginServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request,
        HttpServletResponse response)
```

```

throws ServletException, IOException {

String username = request.getParameter("username");
String password = request.getParameter("password");

// Hardcoded authentication for demo purposes
if ("admin".equals(username) && "1234".equals(password)) {
    // Create a session
    HttpSession session = request.getSession();
    session.setAttribute("username", username);

    // Redirect to welcome page
    response.sendRedirect("WelcomeServlet");
} else {

    // Invalid login
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    out.println("<h3>Invalid username or password!</h3>");
    out.println("<a href='login.html'>Try Again</a>");
}
}
}

```

3. WelcomeServlet.javan :

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

public class WelcomeServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {

        HttpSession session = request.getSession(false); // Get existing
        session, do not create new

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        if (session != null && session.getAttribute("username") != null) {
            String user = (String) session.getAttribute("username");
            out.println("<h2>Welcome, " + user + "!</h2>");
            out.println("<a href='LogoutServlet'>Logout</a>");
        } else {

            out.println("<h3>You are not logged in.</h3>");
            out.println("<a href='login.html'>Login Here</a>");
        }
    }
}
```

```
}  
}
```

4. LogoutServlet.java

Logs the user out by invalidating the session.

java

CopyEdit

```
import java.io.*;  
import javax.servlet.*;  
import javax.servlet.http.*;  
  
public class LogoutServlet extends HttpServlet {  
    protected void doGet(HttpServletRequest request,  
        HttpServletResponse response)  
        throws ServletException, IOException {  
  
        HttpSession session = request.getSession(false);  
        if (session != null) {  
            session.invalidate();  
        }  
  
        response.setContentType("text/html");  
        PrintWriter out = response.getWriter();
```

```
        out.println("<h3>You have been logged out.</h3>");
        out.println("<a href='login.html'>Login Again</a>");
    }

}
```

web.xml Configuration (if not using annotations)

If you're not using `@WebServlet` annotations, you need to configure it in WEB-INF/web.xml:

xml

CopyEdit

```
<web-app>
```

```
    <servlet>
```

```
        <servlet-name>LoginServlet</servlet-name>
```

```
        <servlet-class>LoginServlet</servlet-class>
```

```
    </servlet>
```

```
    <servlet-mapping>
```

```
        <servlet-name>LoginServlet</servlet-name>
```

```
        <url-pattern>/LoginServlet</url-pattern>
```

```
    </servlet-mapping>
```

```
    <servlet>
```

```
        <servlet-name>WelcomeServlet</servlet-name>
```

```
        <servlet-class>WelcomeServlet</servlet-class>
```

```
    </servlet>
```

```
    <servlet-mapping>
```



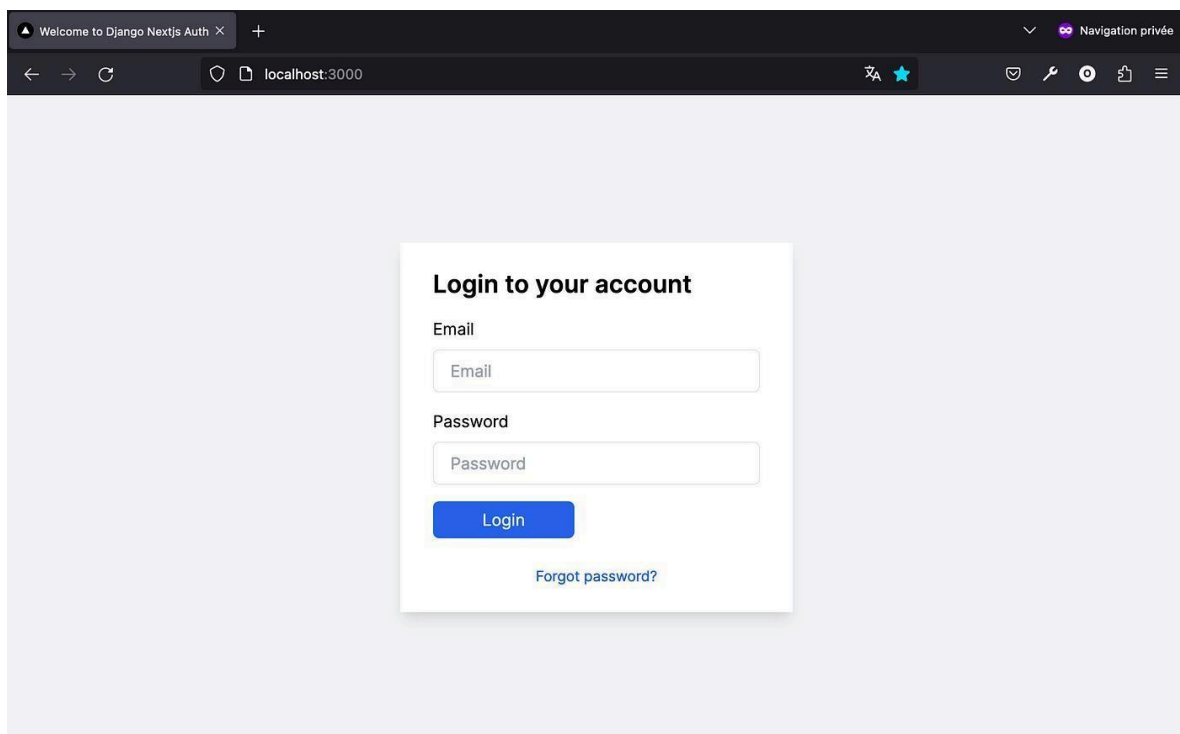
```
<servlet-name>WelcomeServlet</servlet-name>  
<url-pattern>/WelcomeServlet</url-pattern>  
</servlet-mapping>
```

```
<servlet>  
  <servlet-name>LogoutServlet</servlet-name>  
  <servlet-class>LogoutServlet</servlet-class>  
</servlet>
```

```
<servlet-mapping>  
  <servlet-name>LogoutServlet</servlet-name>  
  <url-pattern>/LogoutServlet</url-pattern>  
</servlet-mapping>
```

```
</web-app>
```

OUTPUT :



RESULT :

The program was successfully implemented and executed.

A simple login system was developed using Java Servlets, where session tracking was achieved using HttpSession.

- Upon entering valid credentials, the user is authenticated and redirected to a welcome page.
- The session stores the username and maintains the login state across multiple requests.
- When the user logs out, the session is invalidated and access to the welcome page is restricted until the user logs in again.

Thus, the program demonstrates how HttpSession can be effectively used to implement session tracking in a web application.