**EXP 6 Demonstrate Session Tracking using HttpSession**

**AIM:**

To implement a simple login system using **HttpSession** to track user sessions in a Java Servlet.

**ALGORITHM:**

1. **Initialization**
   * Load frontend interface (login form)
   * Check existing session via /check-session endpoint
2. **Authentication**
   * User submits credentials via POST /login
   * Servlet validates input & creates HttpSession
   * Store username in session attributes
3. **Session Validation**
   * Frontend receives success response
   * Switch to welcome view displaying username
   * Show session-active state
4. **Session Maintenance**
   * All subsequent requests automatically include JSESSIONID cookie
   * Server validates session for each protected action
5. **Termination**
   * User triggers POST /logout
   * Servlet invalidates session
   * Frontend reverts to login form

**PROGRAM:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Welcome Back!</title>

<style>

:root {

--primary-color: #4361ee;

--secondary-color: #3f37c9;

--accent-color: #4cc9f0;

--danger-color: #f72585;

--light-color: #f8f9fa;

--dark-color: #212529;

}

\* {

box-sizing: border-box;

margin: 0;

padding: 0;

}

body {

font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

background: linear-gradient(135deg, #f5f7fa 0%, #c3cfe2 100%);

display: flex;

justify-content: center;

align-items: center;

min-height: 100vh;

padding: 20px;

color: var(--dark-color);

}

.container {

background: white;

border-radius: 16px;

box-shadow: 0 10px 30px rgba(0, 0, 0, 0.1);

padding: 40px;

width: 100%;

max-width: 450px;

text-align: center;

position: relative;

overflow: hidden;

transition: transform 0.3s ease;

}

.container:hover {

transform: translateY(-5px);

}

.container::before {

content: '';

position: absolute;

top: 0;

left: 0;

width: 100%;

height: 8px;

background: linear-gradient(90deg, var(--primary-color), var(--accent-color));

}

h1 {

color: var(--primary-color);

margin-bottom: 30px;

font-size: 28px;

font-weight: 600;

position: relative;

display: inline-block;

}

h1::after {

content: '';

position: absolute;

bottom: -10px;

left: 50%;

transform: translateX(-50%);

width: 60px;

height: 3px;

background: var(--accent-color);

border-radius: 3px;

}

.input-group {

margin-bottom: 25px;

position: relative;

}

input {

width: 100%;

padding: 15px 20px;

border: 2px solid #e9ecef;

border-radius: 8px;

font-size: 16px;

transition: all 0.3s ease;

background-color: #f8f9fa;

}

input:focus {

border-color: var(--primary-color);

box-shadow: 0 0 0 3px rgba(67, 97, 238, 0.2);

outline: none;

background-color: white;

}

input::placeholder {

color: #adb5bd;

}

.btn {

background: linear-gradient(135deg, var(--primary-color), var(--secondary-color));

color: white;

border: none;

padding: 15px 20px;

border-radius: 8px;

cursor: pointer;

font-size: 16px;

font-weight: 600;

width: 100%;

transition: all 0.3s ease;

box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);

}

.btn:hover {

transform: translateY(-2px);

box-shadow: 0 6px 12px rgba(0, 0, 0, 0.15);

}

.logout-btn {

background: linear-gradient(135deg, var(--danger-color), #b5179e);

margin-top: 20px;

}

.welcome-message {

margin: 25px 0;

color: #495057;

font-size: 18px;

line-height: 1.6;

}

.user-avatar {

width: 80px;

height: 80px;

border-radius: 50%;

background: linear-gradient(135deg, var(--accent-color), var(--primary-color));

margin: 0 auto 20px;

display: flex;

align-items: center;

justify-content: center;

color: white;

font-size: 32px;

font-weight: bold;

box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);

}

.hidden {

display: none;

}

.error-message {

color: var(--danger-color);

margin-top: 10px;

font-size: 14px;

}

@media (max-width: 480px) {

.container {

padding: 30px 20px;

}

h1 {

font-size: 24px;

}

}

</style>

</head>

<body>

<div class="container">

<!-- Login Form -->

<div id="login-section">

<h1>Welcome Back!</h1>

<form id="login-form" action="login" method="post">

<div class="input-group">

<input type="text" id="username" name="username" placeholder="Enter your name" required>

</div>

<div class="input-group">

<input type="password" id="password" name="password" placeholder="Enter your password" required>

</div>

<button type="submit" class="btn">Login</button>

<div id="error-message" class="error-message"></div>

</form>

</div>

<!-- Welcome Section -->

<div id="welcome-section" class="hidden">

<div class="user-avatar" id="user-avatar"></div>

<h1>Welcome, <span id="display-name"></span>!</h1>

<div class="welcome-message">

We are happy to see you! Enjoy your session.

</div>

<button id="logout-btn" class="btn logout-btn">Logout</button>

</div>

</div>

<script>

document.addEventListener('DOMContentLoaded', function() {

const loginSection = document.getElementById('login-section');

const welcomeSection = document.getElementById('welcome-section');

const loginForm = document.getElementById('login-form');

const logoutBtn = document.getElementById('logout-btn');

const displayName = document.getElementById('display-name');

const userAvatar = document.getElementById('user-avatar');

const errorMessage = document.getElementById('error-message');

// Check session status on page load

checkSession();

// Login form submission

loginForm.addEventListener('submit', async function(e) {

e.preventDefault();

const formData = new FormData(loginForm);

try {

const response = await fetch('login', {

method: 'POST',

body: formData

});

const result = await response.json();

if (result.success) {

showWelcomeSection(result.username);

} else {

showError(result.message);

}

} catch (error) {

showError('An error occurred. Please try again.');

}

});

// Logout button click

logoutBtn.addEventListener('click', async function() {

try {

const response = await fetch('logout', {

method: 'POST'

});

const result = await response.json();

if (result.success) {

loginSection.style.display = 'block';

welcomeSection.classList.add('hidden');

loginForm.reset();

}

} catch (error) {

console.error('Logout error:', error);

}

});

// Check session status

async function checkSession() {

try {

const response = await fetch('check-session');

const result = await response.json();

if (result.loggedIn) {

showWelcomeSection(result.username);

}

} catch (error) {

console.error('Session check error:', error);

}

}

// Show welcome section

function showWelcomeSection(username) {

displayName.textContent = username;

userAvatar.textContent = username.charAt(0).toUpperCase();

loginSection.style.display = 'none';

welcomeSection.classList.remove('hidden');

errorMessage.textContent = '';

}

// Show error message

function showError(message) {

errorMessage.textContent = message;

}

});

</script>

</body>

</html>

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import org.json.JSONObject;

@WebServlet(name = "LoginServlet", urlPatterns = {"/login", "/logout", "/check-session"})

public class LoginServlet extends HttpServlet {

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

String path = request.getServletPath();

response.setContentType("application/json");

JSONObject jsonResponse = new JSONObject();

if (path.equals("/login")) {

// Handle login

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

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

// Simple authentication (in real app, use proper authentication)

if (username != null && !username.trim().isEmpty() &&

password != null && !password.trim().isEmpty()) {

HttpSession session = request.getSession();

session.setAttribute("username", username);

session.setAttribute("loggedIn", true);

jsonResponse.put("success", true);

jsonResponse.put("username", username);

} else {

jsonResponse.put("success", false);

jsonResponse.put("message", "Username and password are required");

}

}

else if (path.equals("/logout")) {

// Handle logout

HttpSession session = request.getSession(false);

if (session != null) {

session.invalidate();

}

jsonResponse.put("success", true);

}

response.getWriter().write(jsonResponse.toString());

}

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

if (request.getServletPath().equals("/check-session")) {

response.setContentType("application/json");

JSONObject jsonResponse = new JSONObject();

HttpSession session = request.getSession(false);

if (session != null && session.getAttribute("loggedIn") != null) {

jsonResponse.put("loggedIn", true);

jsonResponse.put("username", session.getAttribute("username"));

} else {

jsonResponse.put("loggedIn", false);

}

response.getWriter().write(jsonResponse.toString());

} else {

response.sendError(HttpServletResponse.SC\_NOT\_FOUND);

}

}

}

<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee

http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd"

version="4.0">

<session-config>

<session-timeout>30</session-timeout>

</session-config>

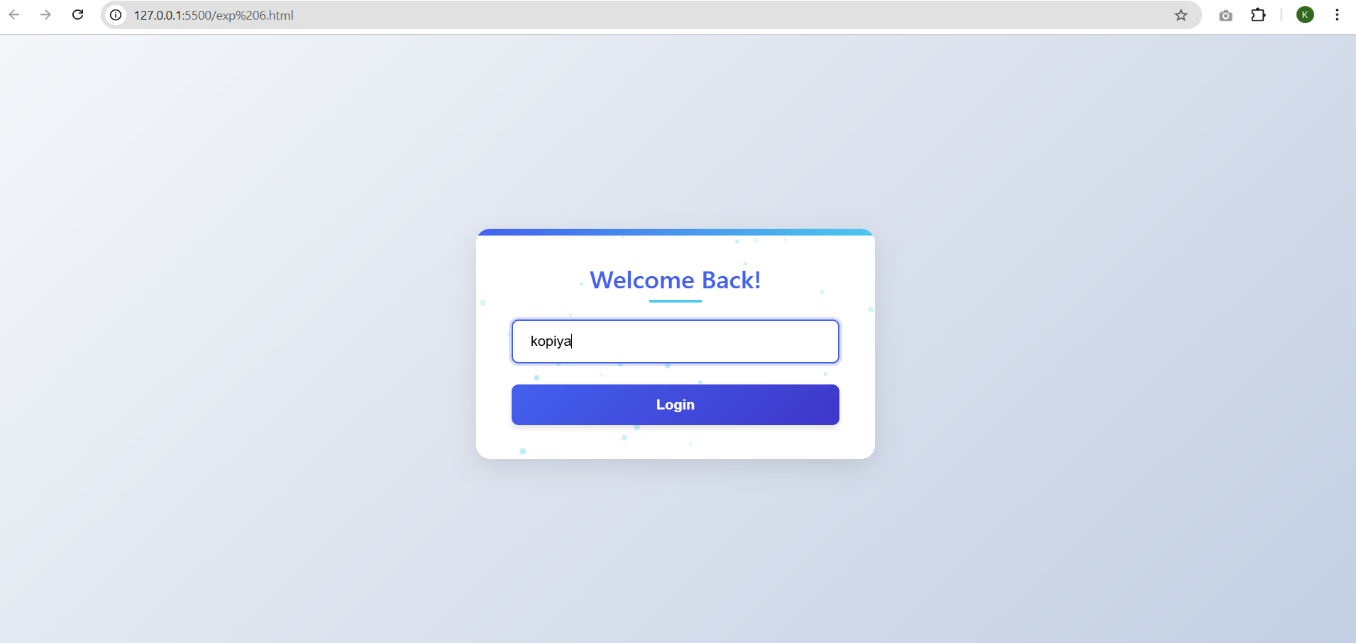
<welcome-file-list>

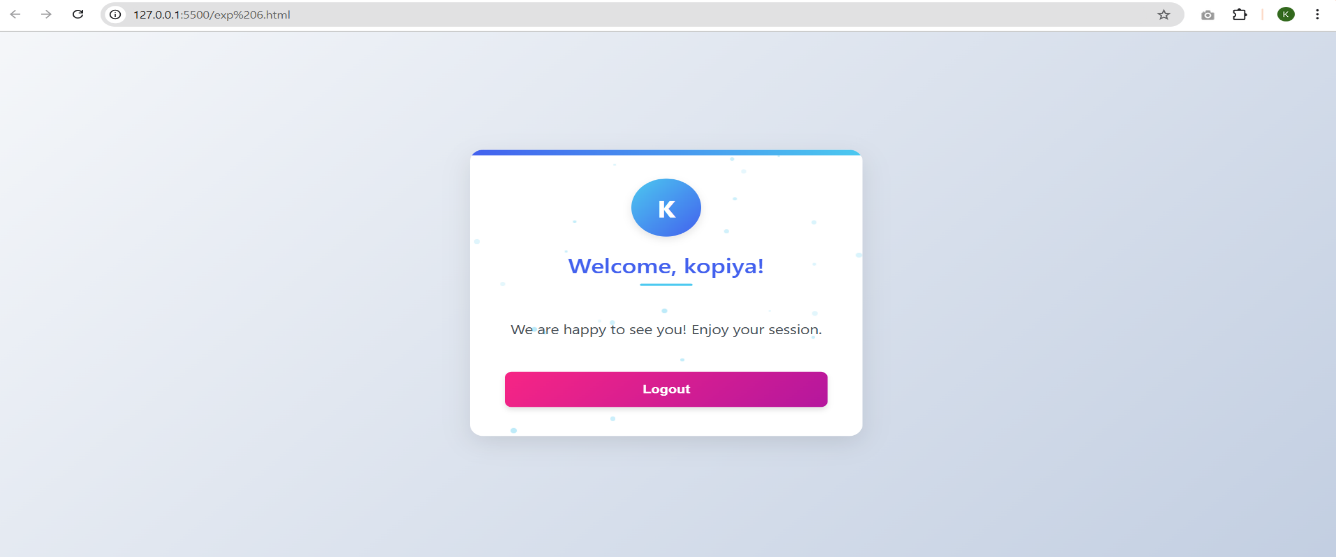
<welcome-file>index.html</welcome-file>

</welcome-file-list>

</web-app>

**OUTPUT:**





**RESULT:** The Session Tracking using HttpSession were executed successfully.