

Group assignment

1. Describe the Role of Sessions (1 mark)

Answer: Sessions in web applications are used to maintain state and store user-specific data across multiple HTTP requests. They allow a server to remember information about a user's interactions with a web application, such as login status, user preferences, or shopping cart contents. This helps in providing a consistent and personalized experience as users navigate through different pages of the application.

2. Differentiate HttpSession getSession() with and without the true parameter (2 marks)

Answer:

HttpSession getSession(): This method retrieves the current session associated with the request. If no session exists, it creates a new one.

- **HttpSession getSession(boolean create):**

- **getSession(true):** This method retrieves the current session if it exists. If no session exists, it creates a new one.
- **getSession(false):** This method retrieves the current session if it exists. If no session exists, it returns null.

the true parameter ensures that a session will always be created if it doesn't already exist, while the false parameter ensures that no new session will be created if one does not exist.

3. Describe the Methods That Can Be Used by the HttpSession Interface (3 marks)

Answer:

The HttpSession interface provides several methods to manage session data:

- **getAttribute(String name):** Retrieves an object bound to this session under the specified name.
- **setAttribute(String name, Object value):** Binds an object to this session, using the specified name.
- **removeAttribute(String name):** Removes the object bound to this session under the specified name.
- **invalidate():** Invalidates this session and unbinds any objects bound to it.
- **getId():** Returns a string representing the unique identifier assigned to this session.
- **getCreationTime():** Returns the time when this session was created, measured in milliseconds since midnight January 1, 1970 GMT.
- **getLastAccessedTime():** Returns the last time the client sent a request associated with this session, measured in milliseconds since midnight January 1, 1970 GMT.
- **getMaxInactiveInterval():** Returns the maximum time interval, in seconds, that the servlet container will keep this session open between client requests.
- **setMaxInactiveInterval(int interval):** Sets the maximum time interval, in seconds, that the servlet container will keep this session open between client requests.

4. Implement a Session-Based Authentication System (4 marks)

a. create a form login.html

```
Project Explorer x login.html x welcome.html error.html LoginServlet.java WelcomeServlet.java LogoutServlet.java
> LOGIN
> Servers
> work
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Login</title>
5 </head>
6 <body>
7   <h2>Login</h2>
8   <form action="login" method="post">
9     Username: <input type="text" name="username" required><br>
10    Password: <input type="password" name="password" required><br>
11    <input type="submit" value="Login">
12  </form>
13 </body>
14 </html>
15
```

b. then welcome.html page:

```
Project Explorer x login.html x welcome.html error.html LoginServlet.java WelcomeServlet.java LogoutServlet.java
> LOGIN
> Servers
> work
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Welcome</title>
5 </head>
6 <body>
7   <h2>Welcome, <span id="username"></span>!</h2>
8   <a href="logout">Logout</a>
9   <script>
10     document.getElementById('username').textContent = sessionStorage.getItem('username') || 'Guest';
11   </script>
12 </body>
13 </html>
14
```

C. Error.html page

```
Project Explorer x login.html welcome.html error.html x LoginServlet.java WelcomeServlet.java LogoutServlet.java
> LOGIN
> Servers
> work
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Error</title>
5 </head>
6 <body>
7   <h2>Login failed. Please check your username and password.</h2>
8   <a href="login.html">Try again</a>
9 </body>
10 </html>
11
```

d. LoginServlet.java

```
Project Explorer x login.html welcome.html error.html LoginServlet.java x WelcomeServlet.java LogoutServlet.java
> LOGIN
> Servers
> work
1 package group;
2
3 import javax.servlet.ServletException;
10
11 @WebServlet("/login")
12 public class LoginServlet extends HttpServlet {
13   @Override
14   protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
15     String username = req.getParameter("username");
16     String password = req.getParameter("password");
17     if ("group".equals(username) && "pass@123".equals(password)) {
18       HttpSession session = req.getSession();
19       session.setAttribute("username", username);
20       resp.sendRedirect("welcome");
21     } else {
22       resp.sendRedirect("error.html");
23     }
24   }
25 }
```

e. WelcomeServlet.java

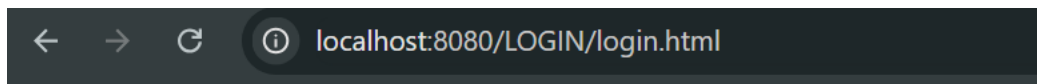
```
Project Explorer x login.html welcome.html error.html LoginServlet.java WelcomeServlet.java LogoutServlet.java
> LOGIN
> Servers
> work
1 package group;
2
3 import javax.servlet.ServletException;
10
11 @WebServlet("/welcome")
12 public class WelcomeServlet extends HttpServlet {
13     @Override
14     protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
15         HttpSession session = req.getSession(false);
16         if (session != null && session.getAttribute("username") != null) {
17             req.getRequestDispatcher("/welcome.html").forward(req, resp);
18         } else {
19             resp.sendRedirect("login.html");
20         }
21     }
22 }
23
```

f. LogoutServlet.java

```
Project Explorer x login.html welcome.html error.html LoginServlet.java WelcomeServlet.java LogoutServlet.java x
> LOGIN
> Servers
> work
1 package group;
2
3 import javax.servlet.ServletException;
10
11 @WebServlet("/logout")
12 public class LogoutServlet extends HttpServlet {
13     @Override
14     protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
15         HttpSession session = req.getSession(false);
16         if (session != null) {
17             session.invalidate();
18         }
19         resp.sendRedirect("login.html");
20     }
21 }
22
```

output:

1. before entering valid credentials:

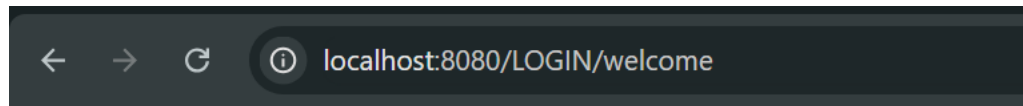


Login

Username:

Password:

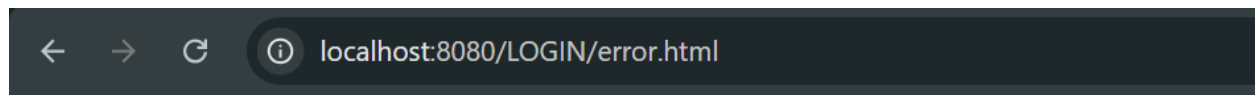
Result:



Welcome, Guest!

[Logout](#)

output: of invalid credentials:



Login failed. Please check your username and password.

[Try again](#)