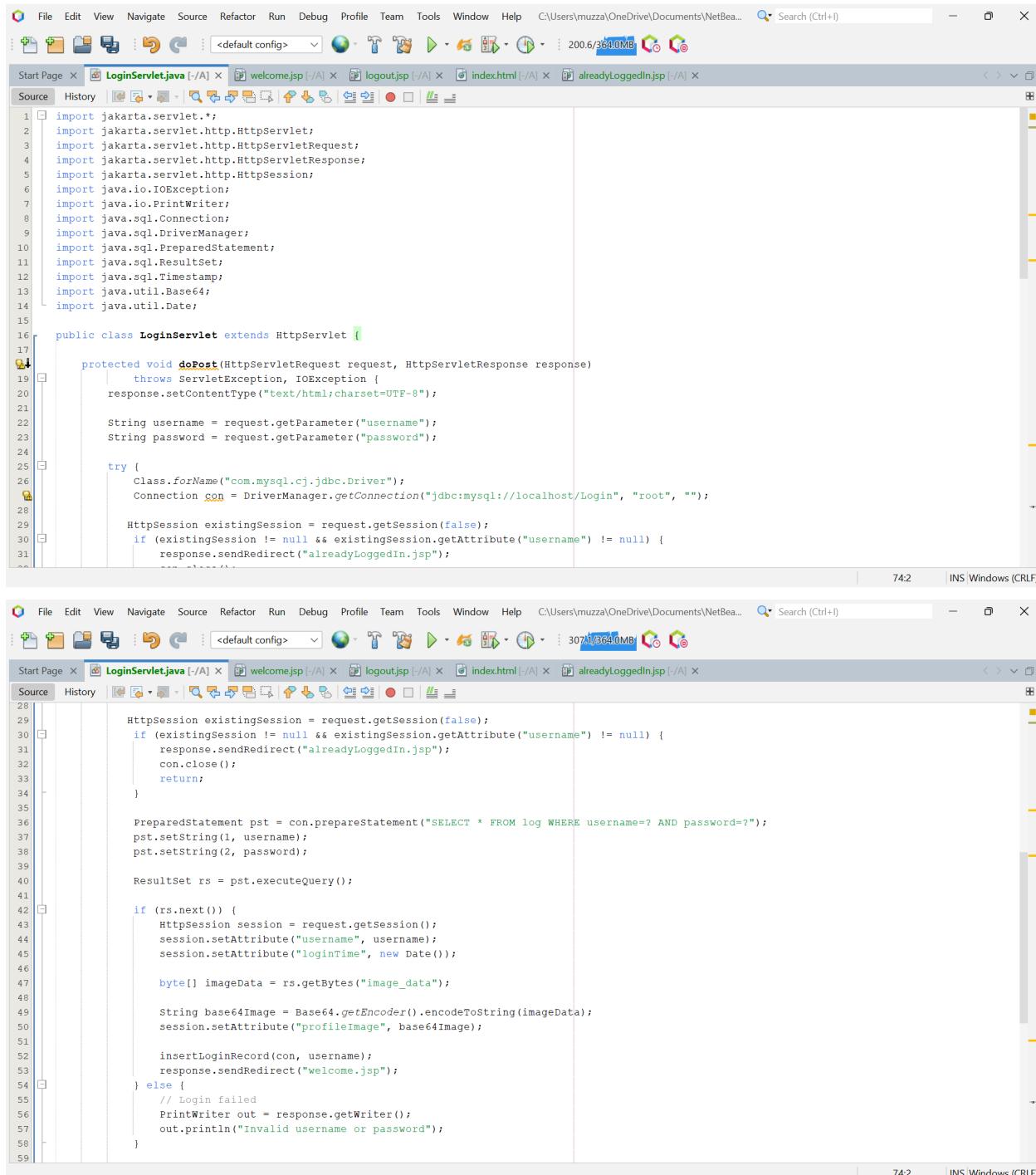


CODE:

LOGINSERVLET



The screenshot shows the NetBeans IDE interface with the LoginServlet.java file open. The code implements a HttpServlet that handles POST requests. It attempts to log in a user by connecting to a MySQL database and checking if the provided username and password match. If successful, it sets session attributes and sends a redirect to welcome.jsp. If unsuccessful, it prints an error message.

```
1 import jakarta.servlet.*;
2 import jakarta.servlet.http.HttpServlet;
3 import jakarta.servlet.http.HttpServletRequest;
4 import jakarta.servlet.http.HttpServletResponse;
5 import jakarta.servlet.http.HttpSession;
6 import java.io.IOException;
7 import java.io.PrintWriter;
8 import java.sql.Connection;
9 import java.sql.DriverManager;
10 import java.sql.PreparedStatement;
11 import java.sql.ResultSet;
12 import java.sql.Timestamp;
13 import java.util.Base64;
14 import java.util.Date;
15
16 public class LoginServlet extends HttpServlet {
17
18     protected void doPost(HttpServletRequest request, HttpServletResponse response)
19             throws ServletException, IOException {
20         response.setContentType("text/html;charset=UTF-8");
21
22         String username = request.getParameter("username");
23         String password = request.getParameter("password");
24
25         try {
26             Class.forName("com.mysql.cj.jdbc.Driver");
27             Connection con = DriverManager.getConnection("jdbc:mysql://localhost/Login", "root", "");
28
29             HttpSession existingSession = request.getSession(false);
30             if (existingSession != null && existingSession.getAttribute("username") != null) {
31                 response.sendRedirect("alreadyLoggedIn.jsp");
32             }
33
34             PreparedStatement pst = con.prepareStatement("SELECT * FROM log WHERE username=? AND password=?");
35             pst.setString(1, username);
36             pst.setString(2, password);
37
38             ResultSet rs = pst.executeQuery();
39
40             if (rs.next()) {
41                 HttpSession session = request.getSession();
42                 session.setAttribute("username", username);
43                 session.setAttribute("loginTime", new Date());
44
45                 byte[] imageData = rs.getBytes("image_data");
46
47                 String base64Image = Base64.getEncoder().encodeToString(imageData);
48                 session.setAttribute("profileImage", base64Image);
49
50                 insertLoginRecord(con, username);
51                 response.sendRedirect("welcome.jsp");
52             } else {
53                 // Login failed
54                 PrintWriter out = response.getWriter();
55                 out.println("Invalid username or password");
56             }
57         }
58     }
59 }
```

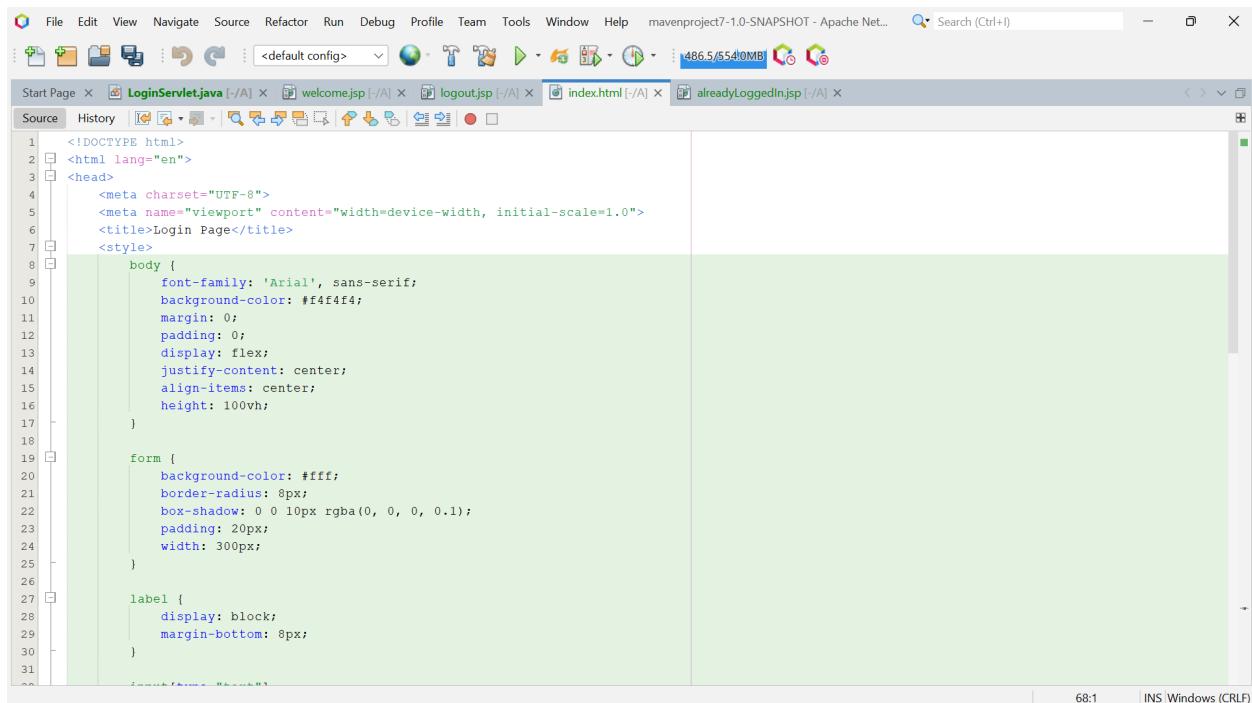
The screenshot shows the NetBeans IDE interface with the following details:

- Title Bar:** File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help C:\Users\muzza\OneDrive\Documents\NetBea... Search (Ctrl+I)
- Toolbar:** Standard NetBeans icons for file operations, search, and project management.
- Project Navigator:** Shows files like <default config>, welcome.jsp, logout.jsp, index.html, and alreadyLoggedIn.jsp.
- Code Editor:** The main window displays the Java code for `LoginServlet.java`. The code handles user login, retrieves profile images from a database, and inserts login records into a session.

```
45     SESSION.setAttribute("loginTime", new Date());
46
47     byte[] imageData = rs.getBytes("image_data");
48
49     String base64Image = Base64.getEncoder().encodeToString(imageData);
50     session.setAttribute("profileImage", base64Image);
51
52     insertLoginRecord(con, username);
53     response.sendRedirect("welcome.jsp");
54 } else {
55     // Login failed
56     PrintWriter out = response.getWriter();
57     out.println("Invalid username or password");
58 }
59
60 con.close();
61 } catch (Exception e) {
62     e.printStackTrace();
63 }
64 }
65
66 private void insertLoginRecord(Connection con, String username) throws Exception {
67     String query = "INSERT INTO user_sessions (username, login_time) VALUES (?, ?)";
68     try (PreparedStatement pst = con.prepareStatement(query)) {
69         pst.setString(1, username);
70         pst.setTimestamp(2, new Timestamp(System.currentTimeMillis()));
71         pst.executeUpdate();
72     }
73 }
```

- Status Bar:** Shows line 74:2 and character 74:2, indicating the current position in the code.

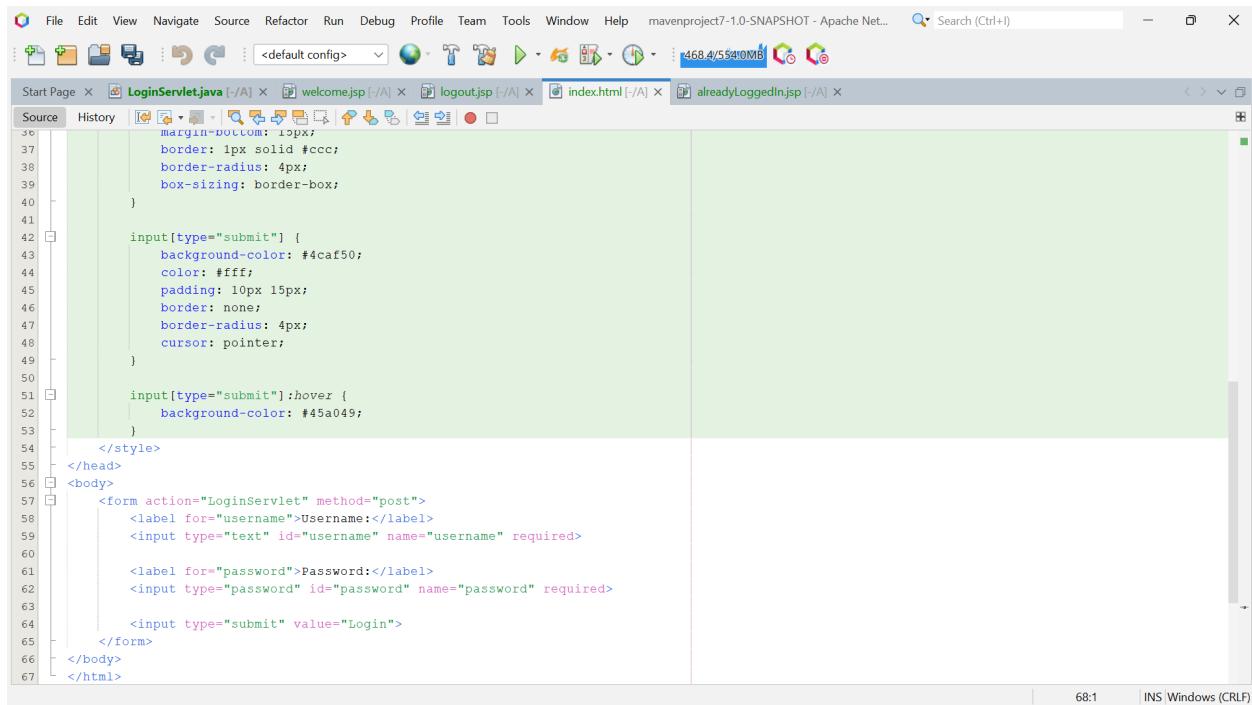
INDEX PAGE



IDE screenshot showing the source code for `index.html`. The code defines a simple login form with a centered body and a form container.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login Page</title>
    <style>
        body {
            font-family: 'Arial', sans-serif;
            background-color: #f4f4f4;
            margin: 0;
            padding: 0;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
        }
        form {
            background-color: #fff;
            border-radius: 8px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
            padding: 20px;
            width: 300px;
        }
        label {
            display: block;
            margin-bottom: 8px;
        }
    </style>

```



IDE screenshot showing the source code for `index.html` with additional CSS and form elements. The code includes a style block with margin and border rules, and a form with input fields and a submit button.

```
        margin-bottom: 15px;
        border: 1px solid #ccc;
        border-radius: 4px;
        box-sizing: border-box;
    }

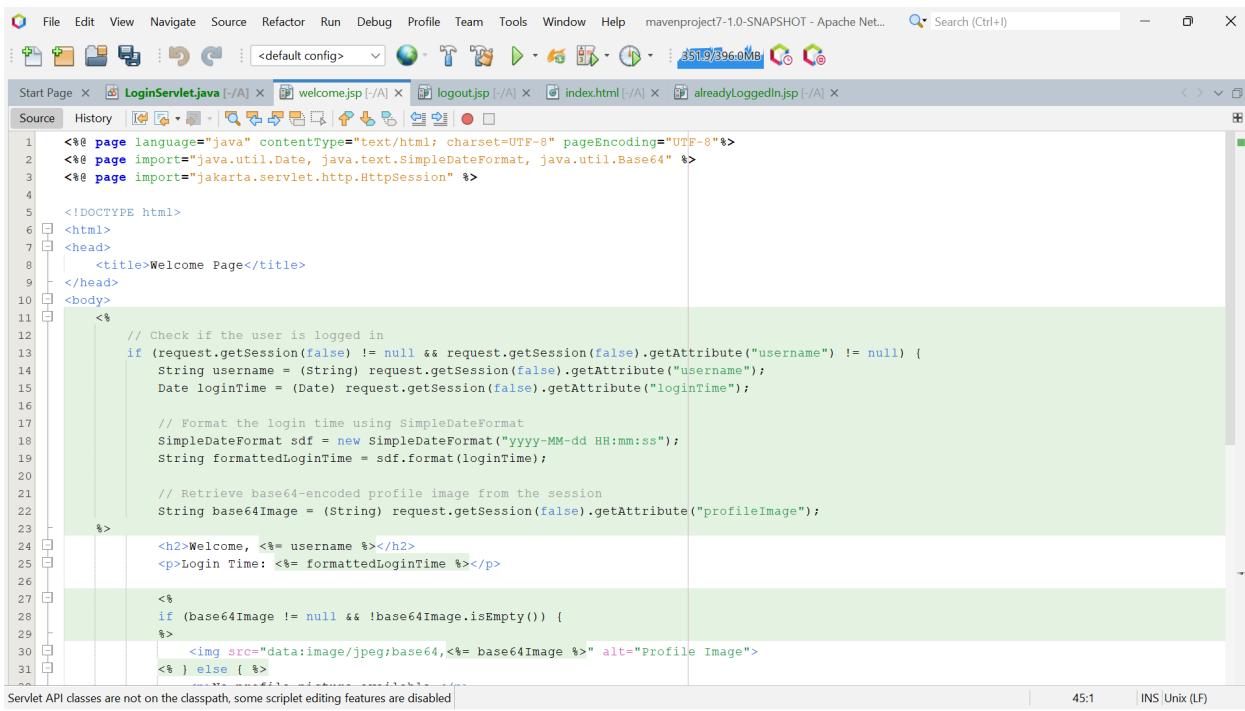
    input[type="submit"] {
        background-color: #4caf50;
        color: #fff;
        padding: 10px 15px;
        border: none;
        border-radius: 4px;
        cursor: pointer;
    }

    input[type="submit"]:hover {
        background-color: #45a049;
    }
</style>
</head>
<body>
    <form action="LoginServlet" method="post">
        <label for="username">Username:</label>
        <input type="text" id="username" name="username" required>

        <label for="password">Password:</label>
        <input type="password" id="password" name="password" required>

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

WELCOME PAGE

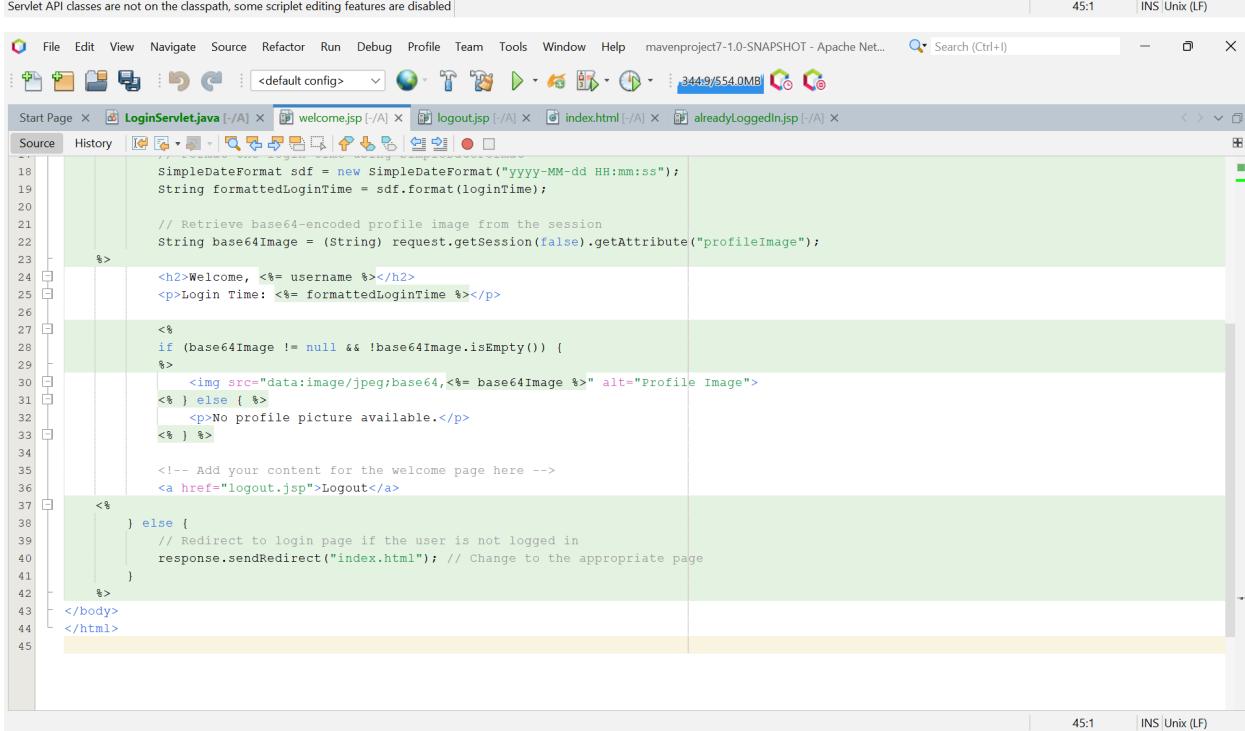


```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ page import="java.util.Date, java.text.SimpleDateFormat, java.util.Base64" %>
<%@ page import="jakarta.servlet.http.HttpSession" %>

<!DOCTYPE html>
<html>
<head>
    <title>Welcome Page</title>
</head>
<body>
    <%
        // Check if the user is logged in
        if (request.getSession(false) != null && request.getSession(false).getAttribute("username") != null) {
            String username = (String) request.getSession(false).getAttribute("username");
            Date loginTime = (Date) request.getSession(false).getAttribute("loginTime");

            // Format the login time using SimpleDateFormat
            SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
            String formattedLoginTime = sdf.format(loginTime);

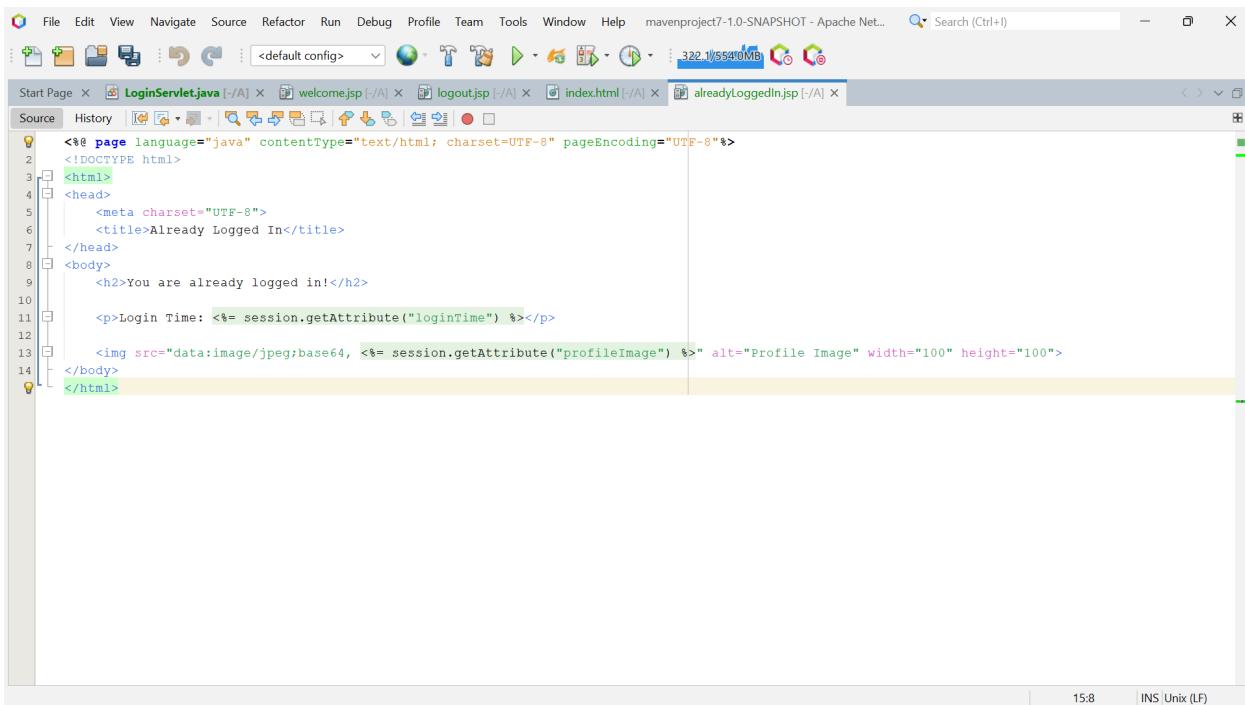
            // Retrieve base64-encoded profile image from the session
            String base64Image = (String) request.getSession(false).getAttribute("profileImage");
    %>
        <h2>Welcome, <%= username %></h2>
        <p>Login Time: <%= formattedLoginTime %></p>
    <%
        if (base64Image != null && !base64Image.isEmpty()) {
    %>
        
    <% } %>
    <!-- Add your content for the welcome page here -->
    <a href="logout.jsp">Logout</a>
<% } else {
        // Redirect to login page if the user is not logged in
        response.sendRedirect("index.html"); // Change to the appropriate page
    }
    <%>
</body>
</html>
```



```
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
    String formattedLoginTime = sdf.format(loginTime);

    // Retrieve base64-encoded profile image from the session
    String base64Image = (String) request.getSession(false).getAttribute("profileImage");
    <%
        if (base64Image != null && !base64Image.isEmpty()) {
    %>
        
        <p>No profile picture available.</p>
    <% } %>
    <!-- Add your content for the welcome page here -->
    <a href="logout.jsp">Logout</a>
<% } else {
        // Redirect to login page if the user is not logged in
        response.sendRedirect("index.html"); // Change to the appropriate page
    }
    <%>
</body>
</html>
```

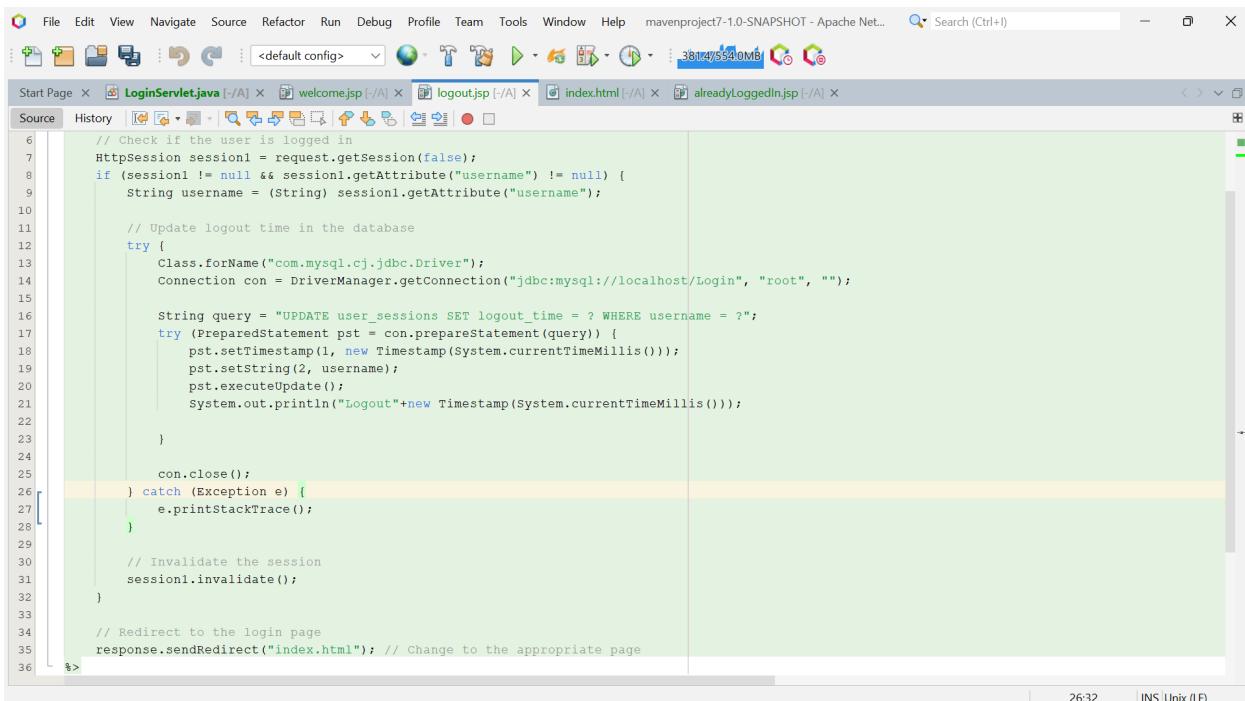
ALREADYLOGGEDIN PAGE



The screenshot shows the Apache NetBeans IDE interface with the file `alreadyLoggedIn.jsp` open in the editor. The code displays a simple JSP page with an embedded Java scriptlet to output session attributes and a base64-encoded profile image.

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Already Logged In</title>
</head>
<body>
<h2>You are already logged in!</h2>
<p>Login Time: <%= session.getAttribute("loginTime") %></p>
" alt="Profile Image" width="100" height="100">
</body>
</html>
```

LOGOUT PAGE



The screenshot shows the Apache NetBeans IDE interface with the file `logout.jsp` open in the editor. The code implements a logout functionality by updating the `user_sessions` database table with a new logout timestamp for the user's session and then invalidating the session.

```
// Check if the user is logged in
HttpSession session1 = request.getSession(false);
if (session1 != null && session1.getAttribute("username") != null) {
    String username = (String) session1.getAttribute("username");

    // Update logout time in the database
    try {
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost/Login", "root", "");

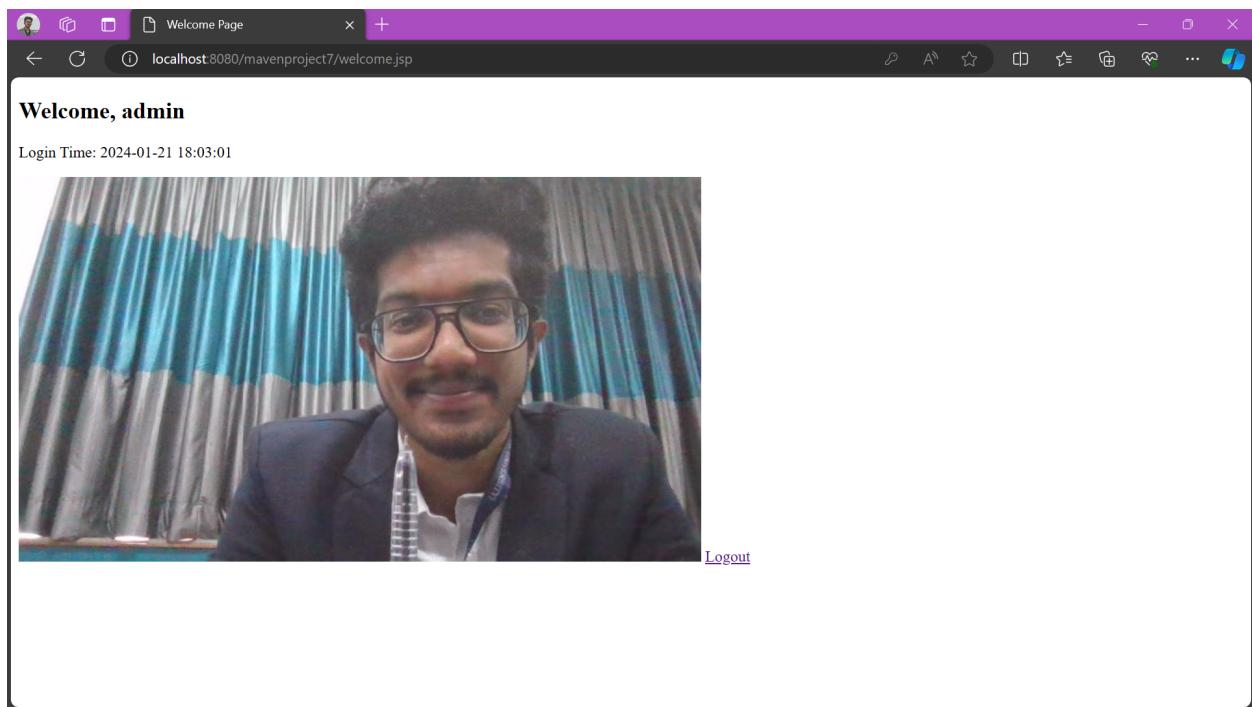
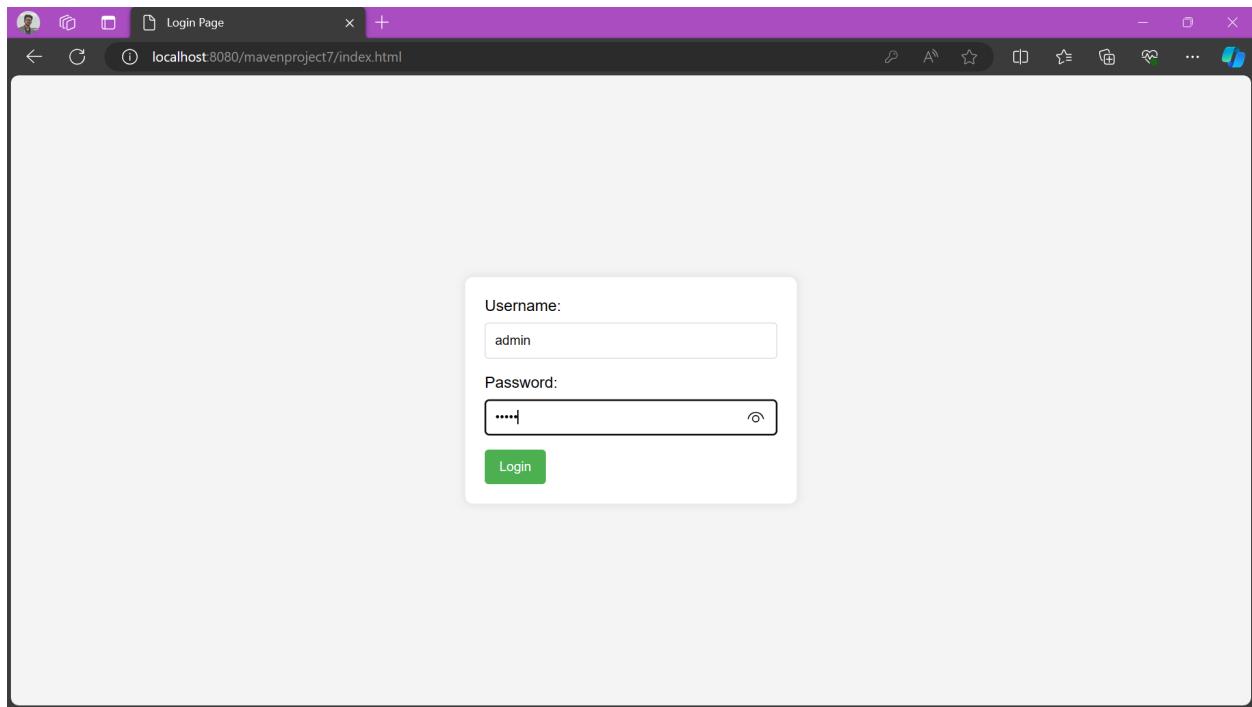
        String query = "UPDATE user_sessions SET logout_time = ? WHERE username = ?";
        try (PreparedStatement pst = con.prepareStatement(query)) {
            pst.setTimestamp(1, new Timestamp(System.currentTimeMillis()));
            pst.setString(2, username);
            pst.executeUpdate();
            System.out.println("Logout"+new Timestamp(System.currentTimeMillis()));
        }
    }

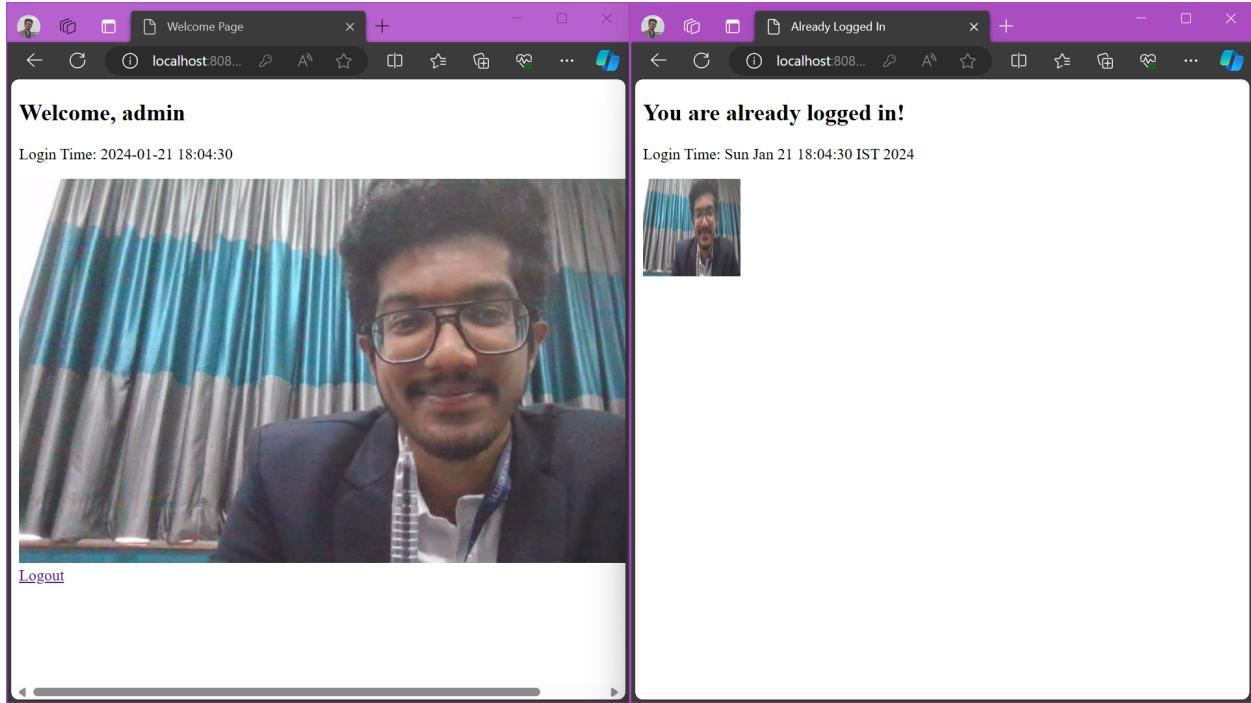
    con.close();
} catch (Exception e) {
    e.printStackTrace();
}

// Invalidate the session
session1.invalidate();

// Redirect to the login page
response.sendRedirect("index.html"); // Change to the appropriate page
%>
```

OUTPUT:





The screenshot shows the phpMyAdmin interface for a MySQL database. The left sidebar lists databases: mysql, performance_schema, phpmyadmin, stu, student, and test. The 'information_schema' database is selected. The main area shows the 'user_sessions' table under the 'login' database. The table has columns: id, username, login_time, and logout_time. There are 18 rows, all with 'admin' as the username and various dates/times between 2024-01-21 and 2024-01-22.

	Edit	Copy	Delete	id	username	login_time	logout_time
<input type="checkbox"/>	Edit	Copy	Delete	1	admin	2024-01-21 17:39:46	2024-01-21 17:39:46
<input type="checkbox"/>	Edit	Copy	Delete	2	admin	2024-01-21 17:39:46	2024-01-21 17:39:46
<input type="checkbox"/>	Edit	Copy	Delete	3	admin	2024-01-21 17:39:46	2024-01-21 17:39:46
<input type="checkbox"/>	Edit	Copy	Delete	4	admin	2024-01-21 17:39:46	2024-01-21 17:39:46
<input type="checkbox"/>	Edit	Copy	Delete	5	admin	2024-01-21 17:39:46	2024-01-21 17:39:46
<input type="checkbox"/>	Edit	Copy	Delete	6	admin	2024-01-21 17:39:46	2024-01-21 17:39:46
<input type="checkbox"/>	Edit	Copy	Delete	7	admin	2024-01-21 17:39:46	2024-01-21 17:39:46
<input type="checkbox"/>	Edit	Copy	Delete	8	admin	2024-01-21 17:39:46	2024-01-21 17:39:46
<input type="checkbox"/>	Edit	Copy	Delete	9	admin	2024-01-21 17:39:46	2024-01-21 17:39:46
<input type="checkbox"/>	Edit	Copy	Delete	10	admin	2024-01-21 17:39:46	2024-01-21 17:39:46
<input checked="" type="checkbox"/>	Console	Copy	Delete	11	admin	2024-01-22 17:39:46	2024-01-22 17:39:46

