

Advanced Server Side Programming (Lab)

Assessment 2 **Servlet web application**

Name : Brian E Shilo

Reg No. : 21MIC0131

Lab Slot : L21 + L22

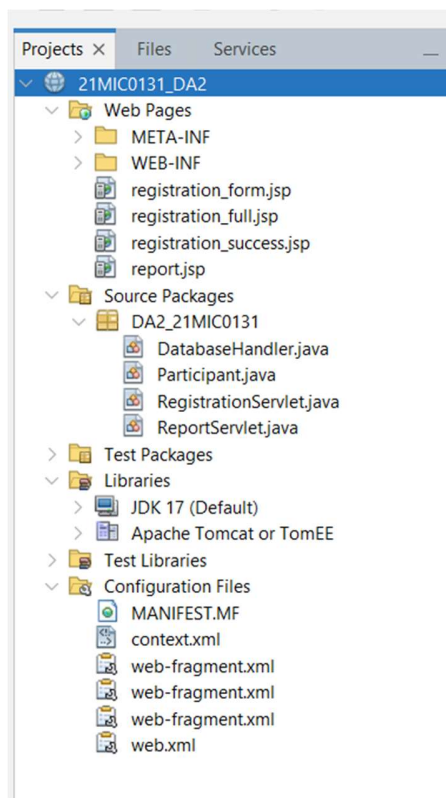
Course code : CSI3023

Problem Statement : Registration process of an expert talk on "Web Services"

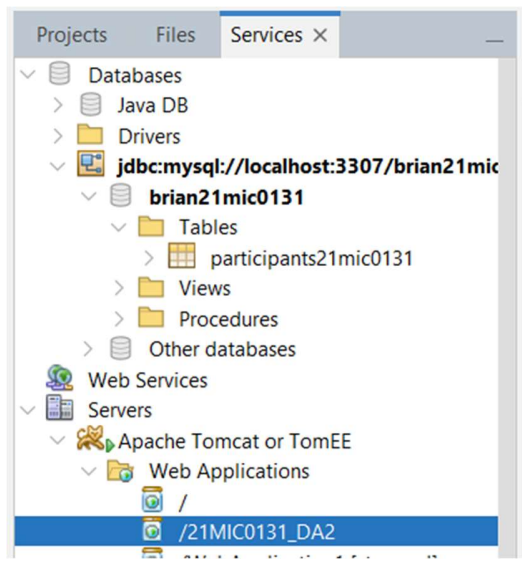
You are asked to carry out the registration process of an expert talk on "Web Services". First, you need to provide the form, which includes the participant personal details (Name, Id, Designation, Affiliation, Fee). Expected participants are students, researchers and software engineer. The registration fee is fixed depends on their designation that means students (500/-), researchers (1000/-) and software engineer (2000/-). The event is organized for the fixed number of participations to 15 each category with 5 participation. You need to display the message as "Registration Full" while it reaches the maximum registration. Once the registration is completed, generate the participation report. The report should contain the list of participant details on each category.

IMPLEMENTATION

Project Structure



Services – Database(MySQL and Server integration(TomCat)



Programs - Codes

Client side

1. registration_form.jsp

```
<!-- registration_form.jsp -->
<%@ page contentType="text/html;charset=UTF-8" language="java"
%>
<html>
<head>
    <title>Web Services Registration</title>
</head>
<body>
    <h2>Web Services Registration Form</h2>
    <form action="${pageContext.request.contextPath}/register"
method="post">
```

```
Name: <input type="text" name="name" required><br>
Designation:
<select name="designation" onchange="calculateFee(this.value)">
  <option value="student">Student</option>
  <option value="researcher">Researcher</option>
  <option value="software_engineer">Software
Engineer</option>
</select><br>
Affiliation: <input type="text" name="affiliation" required><br>
Fee: <span id="feeDisplay">0</span><br>
<input type="submit" value="Register">
</form>
```

```
<script>
function calculateFee(designation) {
  var feeDisplay = document.getElementById("feeDisplay");
  switch (designation) {
    case "student":
      feeDisplay.textContent = "500";
      break;
    case "researcher":
      feeDisplay.textContent = "1000";
      break;
    case "software_engineer":
      feeDisplay.textContent = "2000";
      break;
  }
}
```

```
        default:
            feeDisplay.textContent = "0";
        }
    }
</script>
</body>
</html>
```

2. registration_full.jsp

```
<!-- registration_full.jsp -->
<%@ page contentType="text/html; charset=UTF-8"
language="java" %>
<html>
<head>
    <title>Registration Full</title>
</head>
<body>
    <h2>Registration Full</h2>
    <p>${message}</p>
</body>
</html>
```

3. registration_success.jsp

```
<!-- registration_success.jsp -->
```

```
<%@ page contentType="text/html; charset=UTF-8"
language="java" %>
```

```
<html>
```

```
<head>
```

```
    <title>Registration Success</title>
```

```
</head>
```

```
<body>
```

```
    <h2>Registration Successful</h2>
```

```
    <p>Thank you for registering for the Web Services event!</p>
```

```
    <p><a
```

```
href="{pageContext.request.contextPath}/report.jsp">View
Participant Report</a></p>
```

```
    <!-- Add the link to view the participant report -->
```

```
    <p><a
```

```
href="{pageContext.request.contextPath}/registration_form.js
p">Register Another Person</a></p>
```

```
    <!-- Add the link to register another person -->
```

```
</body>
```

```
</html>
```

4. report.jsp

```
<!-- report.jsp -->
```

```
<%@ page contentType="text/html; charset=UTF-8"
language="java" %>
```

```
<html>
```

```
<head>
```

```
    <title>Participant Report</title>
```

```
</head>
```

```
<body>
```

```
    <h2>Participant Report</h2>
```

```
    <a
```

```
href="${pageContext.request.contextPath}/registration_form.js
p">Go Back to Registration</a>
```

```
    <h3>Students</h3>
```

```
    <ul>
```

```
        <c:forEach var="student" items="${students}">
```

```
            <li>${student.name}, ${student.affiliation}, Fee:
${student.fee}</li>
```

```
        </c:forEach>
```

```
    </ul>
```

```
    <h3>Researchers</h3>
```

```
<ul>
  <c:forEach var="researcher" items="${researchers}">
    <li>${researcher.name}, ${researcher.affiliation}, Fee:
${researcher.fee}</li>
  </c:forEach>
</ul>

<h3>Software Engineers</h3>
<ul>
  <c:forEach var="engineer" items="${softwareEngineers}">
    <li>${engineer.name}, ${engineer.affiliation}, Fee:
${engineer.fee}</li>
  </c:forEach>
</ul>
</body>
</html>
```

Server Side

1. Participant.java

```
// Participant.java
```

```
package DA2_21MIC0131;
```



```
public class Participant {  
    private String name;  
    private String id; // Add ID attribute  
    private String designation;  
    private String affiliation;  
    private int fee;  
  
    // Default constructor  
    public Participant() {  
    }  
  
    // Parameterized constructor with ID  
    public Participant(String name, String id, String designation,  
String affiliation, int fee) {  
        this.name = name;  
        this.id = id;  
        this.designation = designation;  
        this.affiliation = affiliation;  
        this.fee = fee;  
    }  
  
    // Getters and setters for each attribute  
  
    public String getName() {
```

```
    return name;
}
```

```
public void setName(String name) {
    this.name = name;
}
```

```
public String getId() {
    return id;
}
```

```
public void setId(String id) {
    this.id = id;
}
```

```
public String getDesignation() {
    return designation;
}
```

```
public void setDesignation(String designation) {
    this.designation = designation;
}
```

```
public String getAffiliation() {
```

```
    return affiliation;
}
```

```
public void setAffiliation(String affiliation) {
    this.affiliation = affiliation;
}
```

```
public int getFee() {
    return fee;
}
```

```
public void setFee(int fee) {
    this.fee = fee;
}
```

```
// toString method for easy debugging
```

```
@Override
```

```
public String toString() {
    return "Participant{" +
        "name='" + name + '\'' +
        ", id='" + id + '\'' +
        ", designation='" + designation + '\'' +
        ", affiliation='" + affiliation + '\'' +
```

```
        ", fee=" + fee +  
        '}';  
    }  
}
```

2. RegistrationServlet.java

// RegistrationServlet.java

```
package DA2_21MIC0131;
```

```
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 java.util.HashMap;  
import java.util.Map;
```

```
@WebServlet("/register")
```

```
public class RegistrationServlet extends HttpServlet {  
    private static final int MAX_PARTICIPANTS_PER_CATEGORY =  
    5;
```

```
private static final Map<String, Integer> registrationCount =  
new HashMap<>();
```

```
protected void doGet(HttpServletRequest request,  
HttpServletResponse response)  
throws ServletException, IOException {  
String designation = request.getParameter("designation");
```

```
int fee;  
switch (designation) {  
    case "student":  
        fee = 500;  
        break;  
    case "researcher":  
        fee = 1000;  
        break;  
    case "software_engineer":  
        fee = 2000;  
        break;  
    default:  
        fee = 0;  
}
```

```
request.setAttribute("fee", fee);
```

```
// Check registration count for the category
int categoryCount =
registrationCount.getOrDefault(designation, 0);
if (categoryCount < MAX_PARTICIPANTS_PER_CATEGORY) {

request.getRequestDispatcher("registration_form.jsp").forward(
request, response);
} else {
    request.setAttribute("message", "Registration Full for " +
designation + " category");

request.getRequestDispatcher("registration_full.jsp").forward(r
equest, response);
}
}
```

```
protected void doPost(HttpServletRequest request,
HttpServletRequest response)
    throws ServletException, IOException {
    String name = request.getParameter("name");
    String id = request.getParameter("id"); // Add ID
parameter
    String designation = request.getParameter("designation");
    String affiliation = request.getParameter("affiliation");
```

```
int fee;
switch (designation) {
    case "student":
        fee = 500;
        break;
    case "researcher":
        fee = 1000;
        break;
    case "software_engineer":
        fee = 2000;
        break;
    default:
        fee = 0;
}
```

```
Participant participant = new Participant();
participant.setName(name);
participant.setId(id); // Set ID
participant.setDesignation(designation);
participant.setAffiliation(affiliation);
participant.setFee(fee);
```

```
DatabaseHandler.registerParticipant(participant);
```

```
        // Update registration count for the category
        registrationCount.put(designation,
registrationCount.getOrDefault(designation, 0) + 1);

        response.sendRedirect("registration_success.jsp"); //
Redirect to a success page
    }
}
```

3. ReportServlet.java

```
package DA2_21MIC0131;

import java.io.IOException;
import java.io.PrintWriter;
import java.util.ArrayList;
import java.util.List;
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 java.sql.Connection;
```



```
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
```

```
@WebServlet("/report")
```

```
public class ReportServlet extends HttpServlet {
```

```
    protected void doGet(HttpServletRequest request,
        HttpServletResponse response)
```

```
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
```

```
        // Initialize lists for each category
```

```
        List<Participant> students = new ArrayList<>();
```

```
        List<Participant> researchers = new ArrayList<>();
```

```
        List<Participant> softwareEngineers = new ArrayList<>();
```

```
        try (Connection connection =
            DatabaseHandler.getConnection());
```

```
            PreparedStatement preparedStatement =
            connection.prepareStatement("SELECT * FROM
            participants21mic0131");
```

```
            ResultSet resultSet =
            preparedStatement.executeQuery()) {
```

```
// Loop through the result set and categorize participants
while (resultSet.next()) {
    String designation = resultSet.getString("designation");
    Participant participant = new Participant();
    participant.setName(resultSet.getString("name"));
    participant.setId(resultSet.getString("id"));
    participant.setDesignation(designation);

    participant.setAffiliation(resultSet.getString("affiliation"));
    participant.setFee(resultSet.getInt("fee"));

    // Add participant to the respective category list
    switch (designation) {
        case "student":
            students.add(participant);
            break;
        case "researcher":
            researchers.add(participant);
            break;
        case "software_engineer":
            softwareEngineers.add(participant);
            break;
        default:
```

```

        break;
    }
}

// Generate HTML report
out.println("<html><head><title>Participant
Report</title></head><body>");
out.println("<h2>Participant Report</h2>");
out.println("<h3>Students</h3>");
printParticipantList(out, students);
out.println("<h3>Researchers</h3>");
printParticipantList(out, researchers);
out.println("<h3>Software Engineers</h3>");
printParticipantList(out, softwareEngineers);
out.println("</body></html>");

} catch (SQLException e) {
    out.println("Error generating report: " + e.getMessage());
}
}

private void printParticipantList(PrintWriter out,
List<Participant> participants) {
    if (participants.isEmpty()) {

```

```

        out.println("<p>No participants in this category.</p>");
        return;
    }
    out.println("<ul>");
    for (Participant participant : participants) {
        out.println("<li>" + participant.toString() + "</li>");
    }
    out.println("</ul>");
}
}

```

Database side

1. DatabaseHandler.java

// DatabaseHandler.java

```

package DA2_21MIC0131;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;

```

```
public class DatabaseHandler {  
    private static final String JDBC_URL =  
"jdbc:mysql://localhost:3307/brian21mic0131";  
    private static final String USERNAME = "root"; // Update with  
your MySQL username  
    private static final String PASSWORD = "0704101798"; //  
Update with your MySQL password  
  
    private static final Logger LOGGER =  
Logger.getLogger(DatabaseHandler.class.getName());  
  
    public static Connection getConnection() throws  
SQLException {  
        return DriverManager.getConnection(JDBC_URL,  
USERNAME, PASSWORD);  
    }  
  
    public static void registerParticipant(Participant participant) {  
        String sql = "INSERT INTO participants21mic0131 (name,  
designation, affiliation, fee) VALUES (?, ?, ?, ?)";  
  
        try (Connection connection = getConnection();  
            PreparedStatement preparedStatement =  
connection.prepareStatement(sql)) {  
            preparedStatement.setString(1, participant.getName());  

```

```

        preparedStatement.setString(2,
participant.getDesignation());
        preparedStatement.setString(3,
participant.getAffiliation());
        preparedStatement.setInt(4, participant.getFee());

        preparedStatement.executeUpdate();

        // Log information about the registered participant
        LOGGER.log(Level.INFO, "Participant registered: {0}",
participant);

        // Log information about all participants in the database
        logAllParticipants(connection);
    } catch (SQLException e) {
        LOGGER.log(Level.SEVERE, "Error registering participant",
e);
    }
}

private static void logAllParticipants(Connection connection) {
    String selectQuery = "SELECT * FROM
participants21mic0131";
    try (PreparedStatement preparedStatement =
connection.prepareStatement(selectQuery);

```

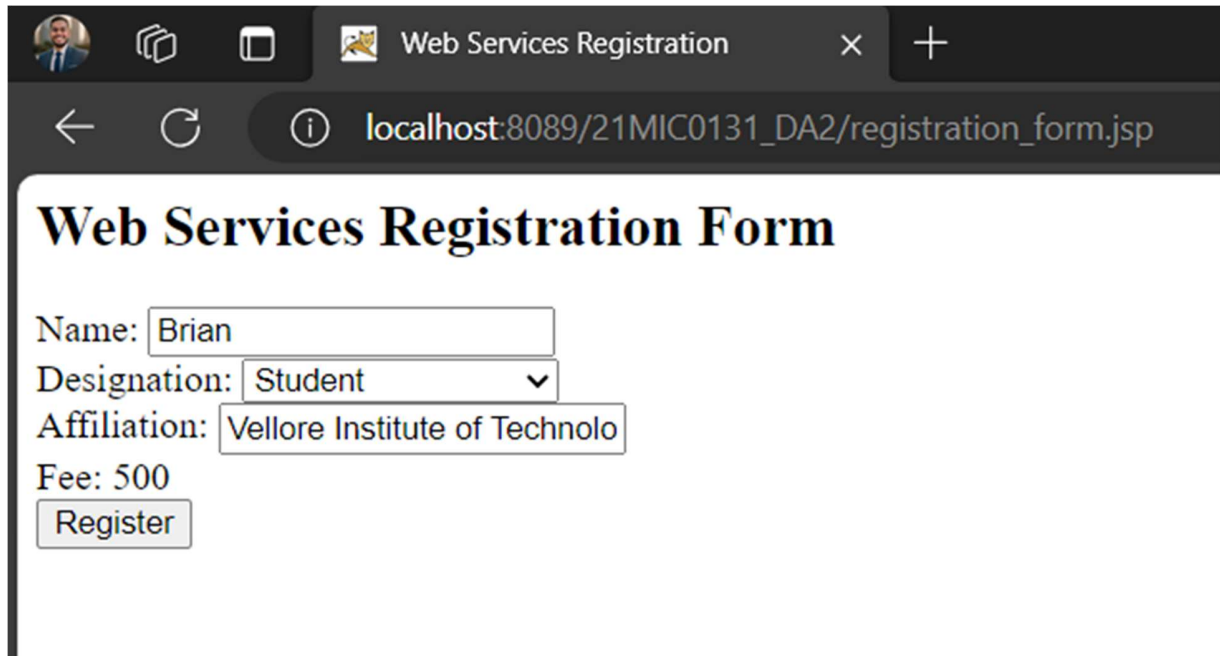
```
var resultSet = preparedStatement.executeQuery() {

    LOGGER.log(Level.INFO, "All Participants:");
    while (resultSet.next()) {
        String name = resultSet.getString("name");
        String designation = resultSet.getString("designation");
        String affiliation = resultSet.getString("affiliation");
        int fee = resultSet.getInt("fee");

        LOGGER.log(Level.INFO, "Name: {0}, Designation: {1},
Affiliation: {2}, Fee: {3}",
            new Object[]{name, designation, affiliation, fee});
    }
} catch (SQLException e) {
    LOGGER.log(Level.SEVERE, "Error retrieving all
participants", e);
}
}
```

Output – Screenshots

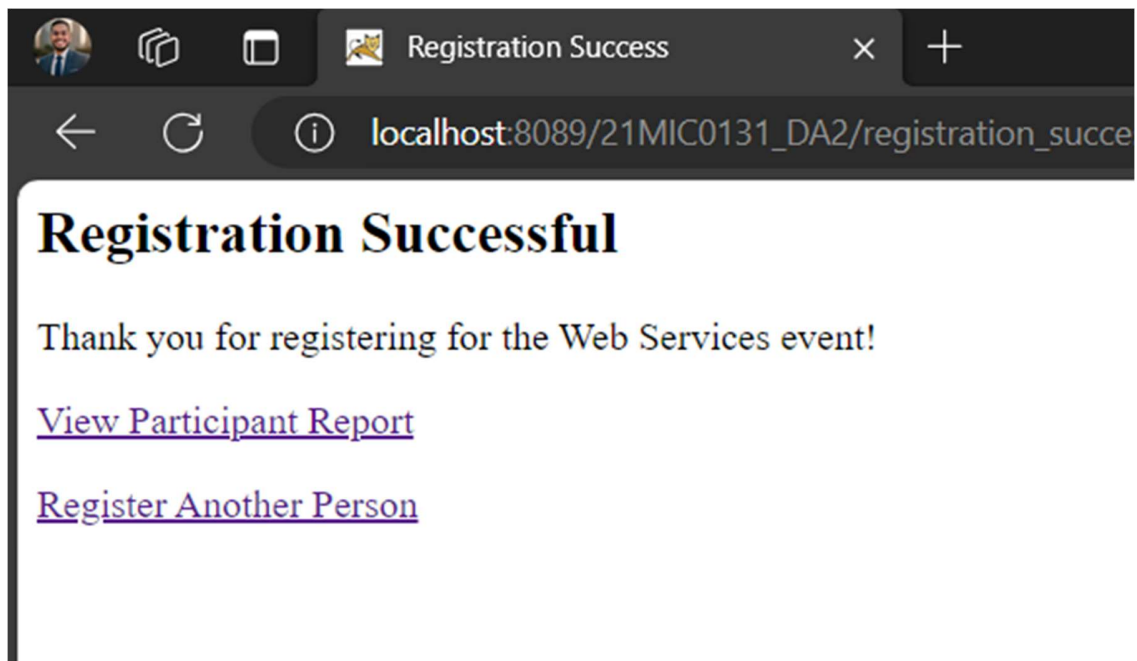
Registration form



A screenshot of a web browser displaying a registration form. The browser's address bar shows the URL `localhost:8089/21MIC0131_DA2/registration_form.jsp`. The page title is "Web Services Registration". The form contains the following fields and values:

- Name:
- Designation: (with a dropdown arrow)
- Affiliation:
- Fee: 500
- A "Register" button.

Registration Successful



A screenshot of a web browser displaying a "Registration Successful" page. The browser's address bar shows the URL `localhost:8089/21MIC0131_DA2/registration_succe`. The page title is "Registration Success". The content of the page includes:

- A heading "Registration Successful".
- A message: "Thank you for registering for the Web Services event!".
- Two links: [View Participant Report](#) and [Register Another Person](#).

Max 5 allowed

The screenshot shows a web browser window with the title 'Web Services Registration'. The address bar shows 'localhost:8089/21MIC0131_DA2/'. The page displays a registration form with the following fields: 'Name' (filled with 'Rajesh'), 'Designation' (a dropdown menu showing 'Student'), 'Affiliation' (filled with 'VIT'), and 'Fee' (filled with '500'). There is a 'Register' button. A dark grey error message box is overlaid on the right side of the form, stating 'localhost:8089 says Registration Full' with an 'OK' button.

Table

The screenshot shows a database table viewer with the query 'SELECT * FROM participant...'. The table has 10 rows and 6 columns: '#', 'id', 'name', 'designation', 'affiliation', and 'fee'. The data is as follows:

#	id	name	designation	affiliation	fee
1	2	Brian	student	Vellore Institute of Technology, Vellore	500
2	3	Vidhisha	student	Vellore Institute of Technology, Vellore	500
3	4	Nandini	student	Vellore Institute of Technology, Vellore	500
4	5	Malhar	student	Vellore Institute of Technology, Vellore	500
5	6	Abhishek	student	Vellore Institute of Technology, Vellore	500
6	7	Aryan	researcher	IIT, Madras	1000
7	8	Surya	researcher	IIT, Bombay	1000
8	9	Rachel	researcher	NIT, Trichy	1000
9	10	Ramini	researcher	Vellore Institute of Technology, Vellore	1000
10	11	Mary	software_engineer	IBM	2000

Report

The screenshot shows a web browser window with the title 'Registration Report'. The address bar shows 'localhost:8089/21MIC0131_DA2/report.jsp'. The page displays a 'Registration Report' with the following sections:

- Category-wise Participants**
- Students:**
 - Brian
 - Vidhisha
 - Nandini
 - Malhar
 - Abhishek
- Researchers:**
 - Aryan
 - Surya
 - Rachel
 - Ramini
- Software Engineers:**
 - Mary

At the bottom, there is a link: [Back to Registration](#)
