



Case Study: Online Feedback Collection System

Objective:

An educational institution needs an online system to collect **student feedback** on courses or instructors. Students will fill out a form, and the data will be processed by a backend Java servlet hosted on a Tomcat 10+ server using Jakarta EE 9+ APIs.



Scenario Description:

The institute offers various technical and non-technical courses. They want to capture feedback from students after course completion. The feedback form includes fields like:

- Student Name
- Email Address
- Course Attended
- Feedback or Suggestions

The institution also wants to:

- Display a confirmation page showing submitted data.
- Optionally store this feedback in a database for future reference.
- Track if a student has already submitted feedback using session/cookie-based mechanisms.



System Design Overview:

1. Frontend (User Interface):

- A JSP (or HTML) page with a form that asks students to enter their feedback.
- The form uses HTTP POST to submit data to the server.

2. Servlet Processing (Backend):

- A Java servlet receives the form data through a POST request.
- The servlet:
 - Extracts parameters (e.g., name, email, feedback).
 - Validates the input (e.g., non-empty fields).

- Generates a response page thanking the student and echoing back the submitted data.
- Optionally, stores the feedback in a database or writes it to a file.

3. Optional Enhancements:

- Use **cookies** to track if a student has already submitted feedback.
- Use **HTTP sessions** to temporarily store user data across requests.
- Redirect to different pages based on whether the feedback was already submitted.

Tools & Technologies:

- **Jakarta EE 9+ (Servlet 5.0)** — For writing the HttpServlet
- **Apache Tomcat 10+** — Web server and servlet container
- **JSP/HTML** — Frontend form
- **Eclipse IDE / IntelliJ** — Development environment
- **Maven** or manual `.war` deployment — For packaging and deployment

Workflow:

1. **Student accesses** the feedback form via a browser (`index.jsp` or `feedback.html`).
2. **Student submits** the form after filling out the details.
3. **Servlet receives** the POST request, processes data, and optionally stores it.
4. **Response page** is generated by the servlet, confirming submission.
5. **Cookie or session tracking** prevents duplicate submissions.

Key Servlet Concepts Demonstrated:

- HttpServlet usage (`doPost()` method)
- Request parameter extraction (`getParameter`)
- Response generation using `PrintWriter` or JSP forwarding
- `@WebServlet` annotation (alternative to `web.xml` mapping)

- Session and cookie handling (optional advanced part)
- Deployment on Tomcat 10+ with Jakarta namespace

SOLUTION

//index.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <title>Feedback Form</title>
</head>
<body>
    <h2>Course Feedback Form</h2>
    <form action="submitFeedback" method="post">
        Name: <input type="text" name="studentName" required><br><br>
        Email: <input type="email" name="email" required><br><br>
        Course: <input type="text" name="course" required><br><br>
        Feedback:<br>
        <textarea name="feedback" rows="5" cols="30" required></textarea><br><br>
        <input type="submit" value="Submit Feedback">
    </form>
</body>
</html>
```

//FeedbackServlet.java

package com.feedback;

```
import java.io.IOException;
import java.io.PrintWriter;

import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
```

```
@WebServlet("/submitFeedback")
public class FeedbackServlet extends HttpServlet {
```

```
/*
 *
 */
private static final long serialVersionUID = 1L;
```

@Override

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
```

```
// Set encoding and content type
response.setContentType("text/html;charset=UTF-8");
PrintWriter out = response.getWriter();

// Extract form data
String name = request.getParameter("studentName");
String email = request.getParameter("email");
String course = request.getParameter("course");
String feedback = request.getParameter("feedback");

// Simple response
out.println("<html><body>");
out.println("<h2>Thank you for your feedback!</h2>");
out.println("<p>Name: " + name + "</p>");
out.println("<p>Email: " + email + "</p>");
out.println("<p>Course: " + course + "</p>");
out.println("<p>Feedback: " + feedback + "</p>");
out.println("</body></html>");

}
```

Course Feedback Form

Name:

Email:

Course:

Feedback:

Thank you for your feedback!

Name: Spoorthi B

Email: spoorthib708@gmail.com

Course: Web Development

Feedback: Had a great Learning !!