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:

1. Student Name
2. Email Address
3. Course Attended
4. Feedback or Suggestions

The institution also wants to:

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

System Design Overview:

1. Frontend (User Interface):

1. A JSP (or HTML) page with a form that asks students to enter their feedback.
2. 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:

1.Extracts parameters (e.g., name, email, feedback).

2.Validates the input (e.g., non-empty fields).

3.Generates a response page thanking the student and echoing back the submitted data

4.Optionally, stores the feedback in a database or writes it to a file

3.Optional Enhancements:

1.Use cookies to track if a student has already submitted feedback.

2.Use HTTP sessions to temporarily store user data across requests.

3.Redirect to different pages based on whether the feedback was already submitted.

4.Tools & Technologies:

1.Jakarta EE 9+ (Servlet 5.0) — For writing the HttpServlet

2.Apache Tomcat 10+ — Web server and servlet container

3. JSP/HTML — Frontend form

4.Eclipse IDE / IntelliJ — Development environment

5. Maven or manual .war deployment — For packaging and deployment

5.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.

6.Key Servlet Concepts Demonstrated:

1.HttpServlet usage (doPost() method)

2. Request parameter extraction (getParameter)

3.Response generation using PrintWriter or JSP forwarding

4.@WebServlet annotation (alternative to web.xml mapping)

5.Session and cookie handling (optional advanced part)

6.Deployment on Tomcat 10+ with Jakarta namespace

index.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
   pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Feedback Form</title>
</head>
<body>
<h2>Course Feedback Form</h2>
<form action="feedback" method="post">
Name: <input type="text" name="name" required><br><br>
Email: <input type="email" name="email" required><br><br>
Course Attended:
<select name="course">
<option>Java</option>
<option>Python</option>
<option>Web Development</option>
<option>Data Science</option>
</select><br><br>
Feedback: <br>
<textarea name="feedback" rows="5" cols="40"
required></textarea><br><br>
<input type="submit" value="Submit Feedback">
</form>
</body>
</html>
```

```
}
```

FeedbackServlet.java

```
package com.casestudy.servlet;
import java.io.IOException;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.*;
@WebServlet("/feedback")
public class FeedbackServlet extends HttpServlet {

    /**
     *
     */
    private static final long serialVersionUID = 1L;
    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        boolean alreadySubmitted = false;
        Cookie[] cookies = req.getCookies();
        if (cookies != null) {
            for (Cookie c : cookies) {
                if ("feedback_submitted".equals(c.getName())) {
                    alreadySubmitted = true;
                    break;
                }
            }
        }

        resp.setContentType("text/html");
        if (alreadySubmitted) {
            resp.getWriter().println("<h3>You have already
            submitted feedback!</h3>");
            return;
        }
        String name = req.getParameter("name");
        String email = req.getParameter("email");
        String course = req.getParameter("course");
        String feedback = req.getParameter("feedback");
        if (name == null || email == null || feedback == null ||
        name.isEmpty() || email.isEmpty() || feedback.isEmpty()) {
            resp.getWriter().println("<h3>All fields are
            required!</h3>");
            return;
        }
        Cookie submittedCookie = new
        Cookie("feedback_submitted", "true");
        submittedCookie.setMaxAge(24 * 60 * 60);
        resp.addCookie(submittedCookie);
        resp.getWriter().println("<h2>Thank you for your
        feedback!</h2>");
        resp.getWriter().println("<p><b>Name:</b> " + name +
        "</p>");
        resp.getWriter().println("<p><b>Email:</b> " + email +
        "</p>");
```

```
                resp.getWriter().println("<p><b>Course:</b> " + course +
                  "</p>");
                resp.getWriter().println("<p><b>Feedback:</b> " + feedback +
                  "</p>");
                  }
}
```