🧾 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 `Print Writer` or JSP forwarding
- `@WebServlet` annotation (alternative to web.xml mapping)

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

Code :

index.jsp (Feedback Form)

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

FeedbackServlet.java

```
package co. Feedback;

import jakarta. servlet. ServletException;
```

```
import Servlet Exception;
import jakarta. servlet. http. *;
import http. *;
import http. *;

@WebServlet("/feedback")
public class Feedback Servlet extends HttpServlet {
   protected void doPost (Feedback Servlet request, Feedback Servlet response)
         throws ServletException, IO Exception {

      String name = request. GetParameter("name");
      String email = request. GetParameter("email");
      String course = request. GetParameter("course");
      String feedback = request. GetParameter("feedback");

      request. GetParameter("text/html");
      Print Writer out = Print Writer ();

      // Check cookie if already submitted
      Cookie [] cookies = Writer ((();
      Boolean already Submitted = false;

      if (cookies! = null) {
         for (Cookie c: cookies) {
            if (cetane (). equals("submitted") && cogitable (). equals(email)) {
               already Submitted = true;
               break;
```

```
            }
        }
    }

    if (already Submitted) {
        outsprinting("<h3>You have already submitted feedback. Thank you! </h3>");
    } else {
        // Set cookie to avoid future submissions
        Cookie submitted Cookie = new Cookie ("submitted", email);
        Cookie ((24 * 60 * 60); // 1 day
        Cookie ((submitted Cookie);

        // Display response
        outsprinting("<h2>Thank you for your feedback! </h2>");
        outsprinting("<p><strong>Name:</strong> " + name + "</p>");
        outsprinting("<p><strong>Email:</strong> " + email + "</p>");
        outsprinting("<p><strong>Course:</strong> " + course + "</p>");
        outsprinting("<p><strong>Feedback:</strong> " + feedback + "</p>");
    }
  }
}
```