

Mini Project Assignment: Online Feedback System

Assigned By: Java Trainer, Archer InfoTech, Pune

Date of Assignment: March 26, 2025

Submission Deadline: March 27, 2025

Technologies Allowed: HTML, CSS, Java Servlets, MySQL, JDBC

Restrictions: No JavaScript, No JSP

Project Overview

The "Online Feedback System" is a web-based application designed to streamline feedback collection and analysis. The system starts with an introduction page that provides an overview and prompts users to submit feedback or admins to log in. Users can register with their name, email, and password, log in, and answer feedback questions set by the admin. Admins can log in to add, update, or delete questions, assign subjects and marks to questions, view feedback results, and calculate an overall rating based on user responses. The application leverages HTML and CSS for the frontend, Java Servlets for backend logic, and MySQL with JDBC for persistent data storage.

Objectives

1. To create a multi-page web application with distinct user and admin interfaces.
 2. To implement user registration, login, and feedback submission using Servlets.
 3. To enable admin management of feedback questions and result analysis with MySQL and JDBC.
 4. To apply concepts of Servlet lifecycle, HTTP request handling, and database operations.
-

Functional Requirements

1. Introduction Page

- A static page (index.html) displaying:
 - A brief introduction (e.g., "Welcome to the Online Feedback System – Share your thoughts or manage feedback!").
 - Two options: "Submit Feedback" (for users) and "Admin Login" (for admins).

2. User Workflow

- **Registration Page:**
 - Form to collect Name, Email, and Password.
 - Submit to store user details in the database.
- **User Login Page:**
 - Form to enter Email and Password.
 - Validate credentials and redirect to the feedback form.
- **Feedback Form Page:**
 - Display a list of questions set by the admin (e.g., "Rate the course content" with a 1–5 scale).
 - User submits answers, which are saved with their profile.

3. Admin Workflow

- **Admin Login Page:**
 - Form to enter admin credentials (e.g., hardcoded or stored in DB).

- **Question Management Page:**
 - Add a new question with text, subject, and maximum marks.
 - Update or delete existing questions.
 - View all questions in a table format.
 - **Results Page:**
 - Display feedback results per user and question.
 - Calculate and show the overall rating (e.g., total marks scored / total possible marks).
4. **Servlet Processing**
- Handle user registration, login, and feedback submission with validation.
 - Manage admin login, question CRUD operations, and result computation.
 - Provide dynamic HTML responses for success or error scenarios.
-

Technical Requirements

1. **Frontend (HTML & CSS):**
 - Use HTML5 for page structure (introduction, registration, login, feedback, admin pages).
 - Use an external CSS file (styles.css) for consistent styling (e.g., forms, tables, buttons).
 - Ensure a clean, professional design (e.g., centered layouts, color scheme).
2. **Backend (Servlets):**
 - **IntroServlet.java:** Redirects from the introduction page to user or admin flows.
 - **UserRegisterServlet.java:** Registers users in the database.
 - **UserLoginServlet.java:** Validates user login and redirects to the feedback form.
 - **FeedbackServlet.java:** Displays admin-set questions and processes user answers.
 - **AdminLoginServlet.java:** Validates admin credentials.
 - **QuestionServlet.java:** Manages question CRUD operations.
 - **ResultServlet.java:** Displays feedback results and calculates the overall rating.
3. **Database (MySQL):**
 - **Database:** feedback_db
 - **Tables:**

```
CREATE TABLE users (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(50) NOT NULL,  
    email VARCHAR(50) UNIQUE NOT NULL,  
    password VARCHAR(50) NOT NULL  
);  
  
CREATE TABLE questions (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    question_text VARCHAR(200) NOT NULL,  
    subject VARCHAR(50) NOT NULL,  
    max_marks INT NOT NULL  
);  
  
CREATE TABLE feedback (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    user_id INT,  
    question_id INT,  
    answer INT NOT NULL,  
    submission_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
```

```
FOREIGN KEY (user_id) REFERENCES users(id),
FOREIGN KEY (question_id) REFERENCES questions(id)
);
```

4. Deployment:

- Deploy on Apache Tomcat.
- Configure web.xml to map Servlets to URL patterns.

Project Structure

```
OnlineFeedbackSystem/
├── src/
│   ├── IntroServlet.java           // Handles intro page navigation
│   ├── UserRegisterServlet.java    // User registration
│   ├── UserLoginServlet.java       // User login
│   ├── FeedbackServlet.java        // Feedback form and submission
│   ├── AdminLoginServlet.java      // Admin login
│   ├── QuestionServlet.java        // Question management
│   └── ResultServlet.java          // Results and rating calculation
├── web/
│   ├── index.html                 // Introduction page
│   ├── register.html              // User registration page
│   ├── user_login.html            // User login page
│   ├── admin_login.html           // Admin login page
│   ├── styles.css                 // CSS styling
│   └── WEB-INF/
│       └── web.xml                 // Servlet configuration
├── lib/
│   └── mysql-connector-java-x.x.xx.jar // MySQL JDBC driver
└── sql/
    └── setup.sql                  // SQL script for database setup
```

5. Detailed Task Breakdown

1. Step 1: Set Up the Database

- Install MySQL, create feedback_db, and execute setup.sql to set up tables.
- Insert a sample admin (e.g., email: "admin@archer.com", password: "admin123").

2. Step 2: Design the Introduction Page (index.html)

- Content:


```
<h1>Welcome to the Online Feedback System</h1>
<p>Share your feedback or manage the system as an admin.</p>
<form action="/userRegister" method="get"><input type="submit" value="Submit Feedback"></form>
<form action="/adminLogin" method="get"><input type="submit" value="Admin Login"></form>
```
- Style with CSS (e.g., center content, button hover effects).

3. Step 3: User Workflow

- Registration (register.html):

- Form: `<input type="text" name="name">`, `<input type="email" name="email">`, `<input type="password" name="password">`.
 - Action: `/register`, Method: `post`.
 - **Login (user_login.html):**
 - Form: `<input type="email" name="email">`, `<input type="password" name="password">`.
 - Action: `/userLogin`, Method: `post`.
 - **Feedback Form (FeedbackServlet):**
 - Fetch questions from questions table.
 - Generate a form dynamically (e.g., `<input type="number" name="q1" min="1" max="5">`).
 - Action: `/submitFeedback`, Method: `post`.
- 4. Step 4: Admin Workflow**
- **Login (admin_login.html):**
 - Form: `<input type="email" name="email">`, `<input type="password" name="password">`.
 - Action: `/adminLogin`, Method: `post`.
 - **Question Management (QuestionServlet):**
 - Add form: `<input type="text" name="question_text">`, `<input type="text" name="subject">`, `<input type="number" name="max_marks">`.
 - Table with "Edit" and "Delete" links (e.g., `Delete`).
 - **Results (ResultServlet):**
 - Fetch data from feedback table, join with users and questions.
 - Calculate overall rating: $(\text{SUM}(\text{answer}) / \text{SUM}(\text{max_marks})) * 100$.

6. Deliverables

1. Complete source code (HTML, CSS, Servlets, web.xml).
2. MySQL database script (setup.sql).
3. A report (2–3 pages) explaining:
 - Implementation details (user/admin flows, database design).
 - Rating calculation logic.
 - Challenges faced and resolutions.

Evaluation Criteria

Criteria	Marks
Page Design & Navigation	20
User Functionality	20
Admin Functionality	25
Database Integration	25
Code Quality & Report	10
Total	100

Guidelines

- Use exception handling for JDBC and Servlet operations.
- Submit as a ZIP file: [YourName]_FeedbackSystem.zip.
- Avoid hardcoding sensitive data (e.g., DB credentials) in production code.

Sample Workflow

1. **User:** Visits index.html, clicks "Submit Feedback," registers, logs in, answers questions, submits.
2. **Admin:** Visits index.html, logs in, adds questions, views results, sees overall rating.

Optional Enhancements (Bonus Points)

- Add a "Logout" link for users and admins.
- Display subject-wise ratings on the results page.
- Add input validation (e.g., email format) in Servlets.