**TEACHERS FEEDBACK FORM**

**1. Problem Statement**

The Teacher Feedback Form application allows students to submit feedback about teachers. The system collects ratings for categories such as interaction, teaching, knowledge, and clarity, along with comments. The feedback is stored in a MySQL database and can be retrieved for analysis.

**2. Dataset Description**

This project uses a relational database (e.g., MySQL, SQLite) to store contact details. The table structure for storing contacts includes the following fields:

| **ColumnName** | **Data Type** | **Description** |
| --- | --- | --- |
| id | INT | Auto-increment primary key for uniquely identifying each feedback. |
| student\_name | VARCHAR(255) | Name of the student submitting the feedback. |
| teacher name | VARCHAR(255) | Name of the teacher receiving the feedback. |
| interaction | INT | Rating for interaction skills (1 to 5). |
| teaching | INT | Rating for teaching skills (1 to 5). |
| knowledge | INT | Rating for subject knowledge (1 to 5). |
| clarity | INT | Rating for clarity of explanation (1 to 5). |
| punctuality | INT | Rating for punctuality (1 to 5). |
| comments | TEXT | Additional comments or feedback from the student. |

Each field represents a specific attribute of a contact, enabling efficient and organized data storage for retrieval and updates.

**3. Language and Concepts Used**

**Programming Language**: Java

**Database**: SQL (structured in a relational database)

**Key Concepts** :

**Object-Oriented Programming (OOP):**

* **Classes and Objects:**
  + **Feedback class to represent each feedback entry.**
  + **Feedback Manager class to manage operations on feedback data.**
  + **Encapsulation: Private fields in classes with public getter and setter methods to access and modify them.**
  + **Abstraction: Database operations are encapsulated in the Feedback Manager class.**

**Java Database Connectivity (JDBC):**

* Used to connect the Java application to the MySQL database.
* Handles SQL queries for CRUD operations:
  + **Create**: Insert new feedback records.
  + **Read**: Retrieve and display feedback records.
  + **Update**: Modify existing feedback entries (can be extended).
  + **Delete**: Remove feedback records (can be extended).

**Swing Framework:**

* Provides the graphical user interface (GUI) for user interaction.
* Includes components like JTextField, JComboBox, JTextArea, and JButton to create an interactive form.

**Exception Handling:**

* Managed using try-catch blocks to handle SQL exceptions and runtime errors gracefully.

**Database Design:**

* Designed a relational database with proper data types and constraints.
* **Primary Key**: Ensures each feedback entry is uniquely identifiable.
* **Constraints**: Ratings are validated using the CHECK constraint to allow values only between 1 and 5.

**Data Validation:**

* Input fields in the form are validated to ensure meaningful and accurate data is stored in the database.

**Input and Output Handling:**

* **Scanner**: For capturing input in console-based interfaces (if used).
* **Swing Components**: For capturing and displaying user input in the GUI.

**4. Code**

import javax.swing.\*;

import javax.swing.SwingUtilities;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.SQLException;

public class TeacherFeedbackForm extends JFrame {

private static final long serialVersionUID = 1L;

private JTextField teacherNameField, studentNameField;

private JComboBox<Integer> interactionBox, teachingBox, knowledgeBox, clarityBox, punctualityBox;

private JTextArea commentsArea;

public TeacherFeedbackForm() {

setTitle("Teacher Feedback Form");

setSize(450, 600);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(new BorderLayout());

// College Name Label

JLabel collegeLabel = new JLabel("KONGU ENGINEERING COLLEGE", JLabel.CENTER);

collegeLabel.setFont(new Font("Serif", Font.BOLD, 18));

collegeLabel.setBorder(BorderFactory.createEmptyBorder(10, 0, 20, 0));

// Form Panel

JPanel formPanel = new JPanel();

formPanel.setLayout(new GridLayout(9, 2, 10, 10));

formPanel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

teacherNameField = new JTextField();

studentNameField = new JTextField();

interactionBox = createRatingComboBox();

teachingBox = createRatingComboBox();

knowledgeBox = createRatingComboBox();

clarityBox = createRatingComboBox();

punctualityBox = createRatingComboBox();

commentsArea = new JTextArea(3, 20);

// Submit Button

JButton submitButton = new JButton("Submit");

formPanel.add(new JLabel("Student Name:"));

formPanel.add(studentNameField);

formPanel.add(new JLabel("Teacher Name:"));

formPanel.add(teacherNameField);

formPanel.add(new JLabel("Interaction:"));

formPanel.add(interactionBox);

formPanel.add(new JLabel("Teaching:"));

formPanel.add(teachingBox);

formPanel.add(new JLabel("Subject Knowledge:"));

formPanel.add(knowledgeBox);

formPanel.add(new JLabel("Clarity:"));

formPanel.add(clarityBox);

formPanel.add(new JLabel("Punctuality:"));

formPanel.add(punctualityBox);

formPanel.add(new JLabel("Comments:"));

formPanel.add(new JScrollPane(commentsArea));

// Adding Components

add(collegeLabel, BorderLayout.NORTH);

add(formPanel, BorderLayout.CENTER);

add(submitButton, BorderLayout.SOUTH);

// Action Listener for Submit Button

submitButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

submitFeedback();

}

});

}

private JComboBox<Integer> createRatingComboBox() {

Integer[] ratings = {1, 2, 3, 4, 5};

return new JComboBox<>(ratings);

}

private void submitFeedback() {

String studentName = studentNameField.getText();

String teacherName = teacherNameField.getText();

int interaction = (int) interactionBox.getSelectedItem();

int teaching = (int) teachingBox.getSelectedItem();

int knowledge = (int) knowledgeBox.getSelectedItem();

int clarity = (int) clarityBox.getSelectedItem();

int punctuality = (int) punctualityBox.getSelectedItem();

String comments = commentsArea.getText();

String query = "INSERT INTO teacher\_feedback (student\_name, teacher\_name, interaction, teaching, knowledge, clarity, punctuality, comments) VALUES (?, ?, ?, ?, ?, ?, ?, ?)";

try (Connection connection = DatabaseConnection.getConnection();

PreparedStatement preparedStatement = connection.prepareStatement(query)) {

preparedStatement.setString(1, studentName);

preparedStatement.setString(2, teacherName);

preparedStatement.setInt(3, interaction);

preparedStatement.setInt(4, teaching);

preparedStatement.setInt(5, knowledge);

preparedStatement.setInt(6, clarity);

preparedStatement.setInt(7, punctuality);

preparedStatement.setString(8, comments);

int rowsAffected = preparedStatement.executeUpdate();

if (rowsAffected > 0) {

JOptionPane.showMessageDialog(this, "Feedback submitted successfully!");

clearForm();

} else {

JOptionPane.showMessageDialog(this, "Failed to submit feedback.");

}

} catch (SQLException ex) {

ex.printStackTrace();

}

}

private void clearForm() {

studentNameField.setText("");

teacherNameField.setText("");

interactionBox.setSelectedIndex(0);

teachingBox.setSelectedIndex(0);

knowledgeBox.setSelectedIndex(0);

clarityBox.setSelectedIndex(0);

punctualityBox.setSelectedIndex(0);

commentsArea.setText("");

}

public static void main(String[] args) {

SwingUtilities.invokeLater(() -> {

TeacherFeedbackForm form = new TeacherFeedbackForm();

form.setVisible(true);

});

}

}

**5. Result**

**1. Feedback Submission**

* Students can submit feedback for teachers through the GUI form.
* Each feedback entry includes:
  + Student Name
  + Teacher Name
  + Ratings for :
    - Interaction
    - Teaching
    - Subject Knowledge
    - Clarity
    - Punctuality
  + Additional comments.
* Feedback is successfully stored in the MySQL database table teacher feedback.

**Example Output**:

+----+--------------+--------------+-------------+----------+-----------+--------+------------+--------------------+

| id | student\_name | teacher name | interaction | teaching | knowledge | clarity| punctuality| comments |

+----+--------------+--------------+-------------+----------+-----------+--------+------------+--------------------+

| 1 | John Doe | Mr. Smith | 5 | 4 | 5 | 4 | 5 | Excellent teacher! |

+----+--------------+--------------+-------------+----------+-----------+--------+------------+--------------------+

**6. Conclusion**

The **Teacher Feedback Form Project** is a comprehensive application that successfully addresses the need for structured feedback collection and analysis. By integrating a user-friendly GUI with robust backend support using Java and MySQL, the project offers a reliable system for gathering valuable insights into teacher performance. It is a step towards improving the quality of education by fostering a feedback-driven culture. It empowers students to share their opinions and provides educators with constructive insights, ultimately contributing to better teaching and learning experiences.