OBE IMPLEMENTATION: COURSE OUTCOME SETTING

by

Srija.CH[AP22110010418]

Vaishnovi[AP22110010420]

Sruthi.P[AP22110010425]

Jahnavi.K[AP22110010434]

Lalitha.A[AP22110010435]

Chaitali.L[AP22110010437]

A report for the CS307: Mobile Application Development using JAVA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

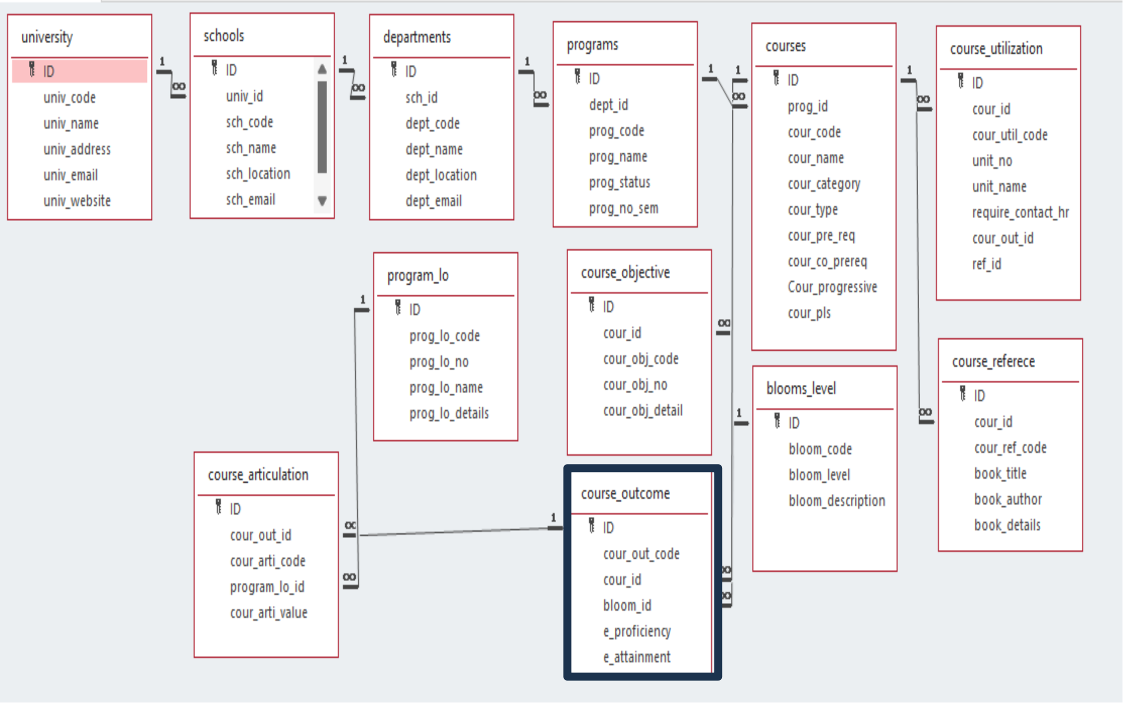
SRM UNIVERSITY AP: AMARAVATHI

INDEX

Introduction:

The Course Outcome Setting module is a critical component of academic management systems that allows administrators to define, track, and assess the learning outcomes associated with each course. This module enables the mapping of course outcomes to Bloom's taxonomy levels and facilitates the measurement of student proficiency and attainment levels. By systematically organizing course outcomes, institutions can ensure alignment with curriculum goals and accreditation requirements.

Architecture Diagram



Module Description

Module Name: course_outcome_setting
Module Description:
This module is used to **Create**, **Update**, **Retrieve**, and **Delete** (hereafter known as **CURD**) the course ouetcome details, such as the course outcome code, proficiency level, and attainment level. These details are stored in the database table course_outcome in MySQL. The module plays a crucial role in mapping expected learning outcomes to individual courses, supporting academic planning, and outcome-based education (OBE) evaluations.

Programming Details –

- **Class name / Activity name:**
  Team11_course_outcome

- **Function / Method names:**

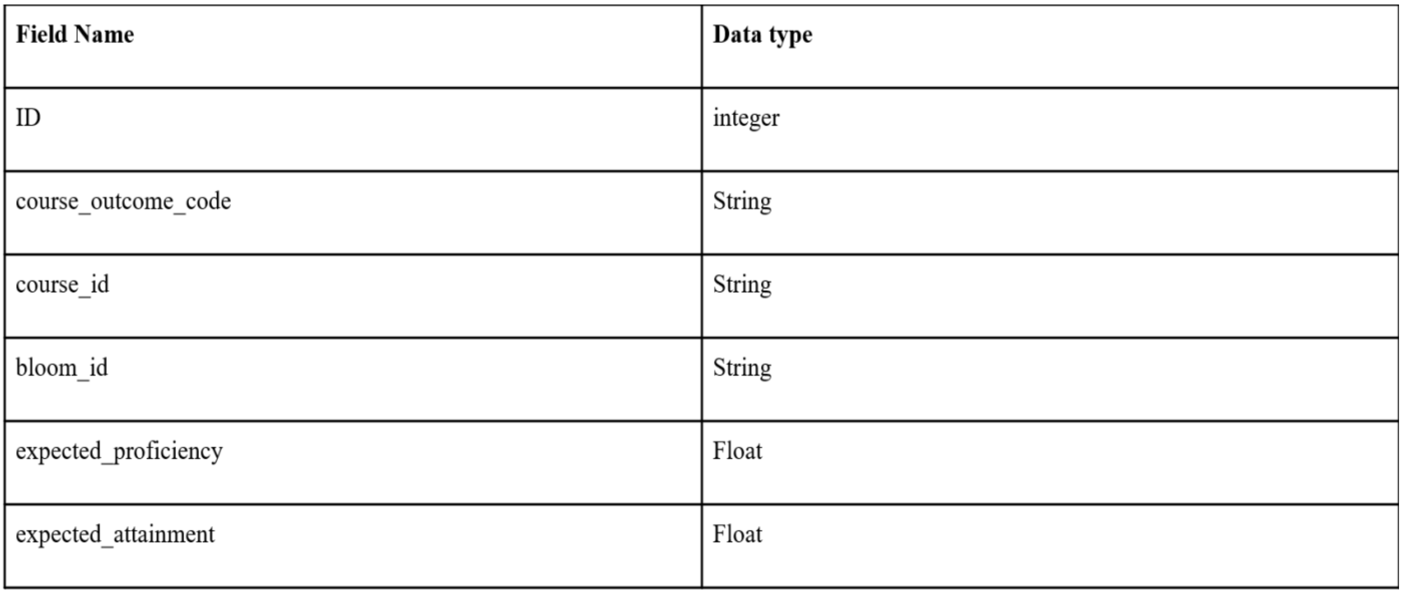
  - **Create:** Team11_course_outcome_create

  - **Update:** Team11_course_outcome_update

  - **Retrieve:** Team11_course_outcome_retrive

  - **Delete:** Team11_course_outcome_delete

Table details: Course Outcome



Field Name	**Data type**
ID	integer
course_outcome_code	String
course_id	String
bloom_id	String
expected_proficiency	Float
expected_attainment	Float

Source Code

Team11_course_outcome.java

```
package courseoutcome;
import java.awt.*;
import java.sql.*;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
public class Team11_course_outcome {
    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            try {

UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
                new CourseOutcomeGUI().setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        });
    }
}
class CourseOutcomeGUI extends JFrame {
    private JTextField tfCode, tfCourseID, tfBloomID, tfProficiency, tfAttainment,
tfUpdateID, tfDeleteID;
    private JScrollPane retrieveScrollPane;
    private static final Font LABEL_FONT = new Font("Segoe UI", Font.BOLD,
16);
    private static final Font FIELD_FONT = new Font("Segoe UI", Font.PLAIN,
16);
    public CourseOutcomeGUI() {
        setTitle("Course Outcome Management System - Team 11");
        setSize(1000, 700);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        JPanel mainPanel = new JPanel(new BorderLayout(10, 10));
        mainPanel.setBorder(BorderFactory.createEmptyBorder(15, 15, 15, 15));
        mainPanel.setBackground(new Color(240, 240, 240));
        mainPanel.add(createHeaderPanel(), BorderLayout.NORTH);
        JTabbedPane tabbedPane = new JTabbedPane();
        tabbedPane.setFont(new Font("Segoe UI", Font.BOLD, 14));
        tabbedPane.addTab("Create", createCreatePanel());
        tabbedPane.addTab("Retrieve", createRetrievePanel());
        tabbedPane.addTab("Update", createUpdatePanel());
        tabbedPane.addTab("Delete", createDeletePanel());
        mainPanel.add(tabbedPane, BorderLayout.CENTER);
        mainPanel.add(createFooterPanel(), BorderLayout.SOUTH);
        add(mainPanel);
    }
    private JPanel createHeaderPanel() {
        JPanel panel = new JPanel();
        panel.setBackground(new Color(70, 130, 180));
        JLabel title = new JLabel("COURSE OUTCOME MANAGEMENT
SYSTEM");
        title.setFont(new Font("Segoe UI", Font.BOLD, 30));
        title.setForeground(Color.WHITE);
        panel.add(title);
        return panel;
    }
    private JPanel createCreatePanel() {
        JPanel panel = new JPanel(new GridBagLayout());
        panel.setBackground(Color.WHITE);
        panel.setBorder(BorderFactory.createEmptyBorder(20, 20, 20, 20));
        GridBagConstraints gbc = new GridBagConstraints();
        gbc.insets = new Insets(10, 10, 10, 10);
        gbc.fill = GridBagConstraints.HORIZONTAL;
        String[] labels = {"Course Outcome Code:", "Course ID:", "Bloom ID:",
                          "Expected Proficiency:", "Expected Attainment:"};
        JTextField[] fields = {tfCode = new JTextField(20), tfCourseID = new
JTextField(20),
                              tfBloomID = new JTextField(20), tfProficiency = new
JTextField(20),
                              tfAttainment = new JTextField(20)};
        for (int i = 0; i < labels.length; i++) {
            JLabel label = new JLabel(labels[i]);
            label.setFont(LABEL_FONT);
            gbc.gridx = 0;
            gbc.gridy = i;
            panel.add(label, gbc);
            fields[i].setFont(FIELD_FONT);
            gbc.gridx = 1;
            panel.add(fields[i], gbc);
        }
        gbc.gridx = 0;
        gbc.gridy = labels.length;
        gbc.gridwidth = 2;
        gbc.anchor = GridBagConstraints.CENTER;
        JButton btnCreate = createStyledButton("Create Record", new Color(34,
139, 34));
        btnCreate.addActionListener(e -> Team11_course_outcome_create());
        panel.add(btnCreate, gbc);
        return panel;
    }
    private void Team11_course_outcome_create() {
        try {
            if (!isValidIdFormat(tfCode.getText()) ||
!isValidIdFormat(tfCourseID.getText()) ||
                !isValidIdFormat(tfBloomID.getText())) {
                JOptionPane.showMessageDialog(this, "ID fields must start with letters
and can include numbers", "Input Error", JOptionPane.ERROR_MESSAGE);
                return;
            }
            if (!isCourseOutcomeCodeUnique(tfCode.getText())) {
                JOptionPane.showMessageDialog(this, "Course Outcome Code must be
unique", "Duplicate Code", JOptionPane.ERROR_MESSAGE);
                return;
            }
            Team11_course_outcome_update("INSERT INTO
team11_course_outcome (course_outcome_code, course_id, bloom_id,
expected_proficiency, expected_attainment) VALUES (?, ?, ?, ?, ?)",
                tfCode.getText(), tfCourseID.getText(), tfBloomID.getText(),
                Float.parseFloat(tfProficiency.getText()),
Float.parseFloat(tfAttainment.getText()));
            tfCode.setText("");
            tfCourseID.setText("");
            tfBloomID.setText("");
            tfProficiency.setText("");
            tfAttainment.setText("");
        } catch (NumberFormatException ex) {
            JOptionPane.showMessageDialog(this, "Invalid number format", "Input
Error", JOptionPane.ERROR_MESSAGE);
        }
    }
    private JPanel createRetrievePanel() {
        JPanel panel = new JPanel(new BorderLayout(10, 10));
        panel.setBackground(Color.WHITE);
        panel.setBorder(BorderFactory.createEmptyBorder(20, 20, 20, 20));
        JPanel buttonPanel = new JPanel();
        buttonPanel.setBackground(Color.WHITE);
        JButton btnRetrieve = createStyledButton("Retrieve All Records", new
Color(70, 130, 180));
        btnRetrieve.addActionListener(e -> Team11_course_outcome_retrieve());
        buttonPanel.add(btnRetrieve);
        panel.add(buttonPanel, BorderLayout.NORTH);
        retrieveScrollPane = new JScrollPane();
        retrieveScrollPane.setBorder(BorderFactory.createTitledBorder("Course
Outcome Records"));
```

```
        panel.add(retrieveScrollPane, BorderLayout.CENTER);
        return panel;
    }
    private void Team11_course_outcome_retrieve() {
        try (Connection conn = DBHelper.getConnection();
             Statement stmt = conn.createStatement();
             ResultSet rs = stmt.executeQuery("SELECT * FROM
team11_course_outcome")) {
            DefaultTableModel model = new DefaultTableModel();
            ResultSetMetaData metaData = rs.getMetaData();
            for (int i = 1; i <= metaData.getColumnCount(); i++) {
                model.addColumn(metaData.getColumnName(i));
            }
            while (rs.next()) {
                Object[] row = new Object[metaData.getColumnCount()];
                for (int i = 1; i <= metaData.getColumnCount(); i++) {
                    row[i - 1] = rs.getObject(i);
                }
                model.addRow(row);
            }
            JTable table = new JTable(model);
            table.setFont(FIELD_FONT);
            table.setRowHeight(30);
            retrieveScrollPane.setViewportView(table);
        } catch (SQLException ex) {
            showError("Database Error: " + ex.getMessage());
        }
    }
    private JPanel createUpdatePanel() {
        JPanel panel = new JPanel(new GridBagLayout());
        panel.setBackground(Color.WHITE);
        panel.setBorder(BorderFactory.createEmptyBorder(20, 20, 20, 20));
        GridBagConstraints gbc = new GridBagConstraints();
        gbc.insets = new Insets(10, 10, 10, 10);
        gbc.fill = GridBagConstraints.HORIZONTAL;
        JLabel updateLabel = new JLabel("Outcome Code to Update:");
        updateLabel.setFont(LABEL_FONT);
        gbc.gridx = 0;
        gbc.gridy = 0;
        panel.add(updateLabel, gbc);
        tfUpdateID = new JTextField(20);
        tfUpdateID.setFont(FIELD_FONT);
        gbc.gridx = 1;
        panel.add(tfUpdateID, gbc);
        String[] labels = {"New Course ID:", "New Bloom ID:", "New Expected
Proficiency:", "New Expected Attainment:"};
```

```
        JTextField[] fields = new JTextField[labels.length];
        for (int i = 0; i < labels.length; i++) {
            JLabel label = new JLabel(labels[i]);
            label.setFont(LABEL_FONT);
            gbc.gridx = 0;
            gbc.gridy = i+1;
            panel.add(label, gbc);
            fields[i] = new JTextField(20);
            fields[i].setFont(FIELD_FONT);
            gbc.gridx = 1;
            panel.add(fields[i], gbc);
        }
        gbc.gridx = 0;
        gbc.gridy = labels.length+1;
        gbc.gridwidth = 2;
        gbc.anchor = GridBagConstraints.CENTER;
        JButton btnUpdate = createStyledButton("Update Record", new Color(218,
165, 32));
        btnUpdate.addActionListener(e -> {
            if (tfUpdateID.getText().trim().isEmpty()) {
                showError("Outcome Code is required");
                return;
            }
            try {
                Team11_course_outcome_update("UPDATE team11_course_outcome
SET course_id=?, bloom_id=?, expected_proficiency=?, expected_attainment=?
WHERE course_outcome_code=?",
                    fields[0].getText(), fields[1].getText(),
Float.parseFloat(fields[2].getText()),
                    Float.parseFloat(fields[3].getText()), tfUpdateID.getText());
                tfUpdateID.setText("");
                for (JTextField field : fields) field.setText("");
            } catch (NumberFormatException ex) {
                showError("Invalid number format");
            }
        });
        panel.add(btnUpdate, gbc);

        return panel;
    }
    private JPanel createDeletePanel() {
        JPanel panel = new JPanel(new GridBagLayout());
        panel.setBackground(Color.WHITE);
        panel.setBorder(BorderFactory.createEmptyBorder(20, 20, 20, 20));
        GridBagConstraints gbc = new GridBagConstraints();
        gbc.insets = new Insets(10, 10, 10, 10);
        JLabel deleteLabel = new JLabel("Record ID to Delete:");
        deleteLabel.setFont(LABEL_FONT);
        gbc.gridx = 0;
        gbc.gridy = 0;
        panel.add(deleteLabel, gbc);
        tfDeleteID = new JTextField(20);
        tfDeleteID.setFont(FIELD_FONT);
        gbc.gridx = 1;
        panel.add(tfDeleteID, gbc);
        gbc.gridx = 0;
        gbc.gridy = 1;
        gbc.gridwidth = 2;
        gbc.anchor = GridBagConstraints.CENTER;
        JButton btnDelete = createStyledButton("Delete Record", new Color(178,
34, 34));
        btnDelete.addActionListener(e -> {
            try {
                Team11_course_outcome_delete("DELETE FROM
team11_course_outcome WHERE id=?", Integer.parseInt(tfDeleteID.getText()));
                tfDeleteID.setText("");
            } catch (NumberFormatException ex) {
                showError("Invalid ID format");
            }
        });
        panel.add(btnDelete, gbc);
        return panel;
    }
    private void Team11_course_outcome_update(String sql, Object... params) {
        try (Connection conn = DBHelper.getConnection();
             PreparedStatement pstmt = conn.prepareStatement(sql)) {
            for (int i = 0; i < params.length; i++) {
                pstmt.setObject(i+1, params[i]);
            }
            int rows = pstmt.executeUpdate();
            JOptionPane.showMessageDialog(this, rows + " record(s) affected",
"Success", JOptionPane.INFORMATION_MESSAGE);
        } catch (SQLException e) {
            showError("Database Error: " + e.getMessage());
        }
    }
    private void Team11_course_outcome_delete(String sql, Object... params) {
        try (Connection conn = DBHelper.getConnection();
             PreparedStatement pstmt = conn.prepareStatement(sql)) {
            for (int i = 0; i < params.length; i++) {
                pstmt.setObject(i+1, params[i]);
            }
            int rows = pstmt.executeUpdate();
            JOptionPane.showMessageDialog(this, rows + " record(s) deleted",
"Success", JOptionPane.INFORMATION_MESSAGE);
        } catch (SQLException e) {
            showError("Database Error: " + e.getMessage());
        }
    }
    private JPanel createFooterPanel() {
        JPanel panel = new JPanel();
        panel.setBackground(new Color(70, 130, 180));
        JLabel footer = new JLabel("Team 11 - Course Outcome Management
System");
        footer.setForeground(Color.WHITE);
        panel.add(footer);
        return panel;
    }
    private JButton createStyledButton(String text, Color bgColor) {
        JButton button = new JButton(text);
        button.setFont(new Font("Segoe UI", Font.BOLD, 16));
        button.setBackground(bgColor);
        button.setForeground(Color.BLACK);
        button.setFocusPainted(false);
        button.setBorder(BorderFactory.createEmptyBorder(10, 25, 10, 25));
        button.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseEntered(java.awt.event.MouseEvent evt) {
                button.setBackground(bgColor.darker());
            }
            public void mouseExited(java.awt.event.MouseEvent evt) {
                button.setBackground(bgColor);
            }
        });
        return button;
    }
    private boolean isCourseOutcomeCodeUnique(String code) {
        try (Connection conn = DBHelper.getConnection();
             PreparedStatement pstmt = conn.prepareStatement("SELECT COUNT(*)
FROM team11_course_outcome WHERE course_outcome_code=?")) {
            pstmt.setString(1, code);
            ResultSet rs = pstmt.executeQuery();
            return rs.next() && rs.getInt(1) == 0;
        } catch (SQLException e) {
            showError("Database Error: " + e.getMessage());
            return false;
        }
    }
    private boolean isValidIdFormat(String input) {
        return input != null && !input.trim().isEmpty() && input.matches("^[a-zA-
Z]+[a-zA-Z0-9]*$");
    }
    private void showError(String message) {
        JOptionPane.showMessageDialog(this, message, "Error",
JOptionPane.ERROR_MESSAGE);
    }
}
```

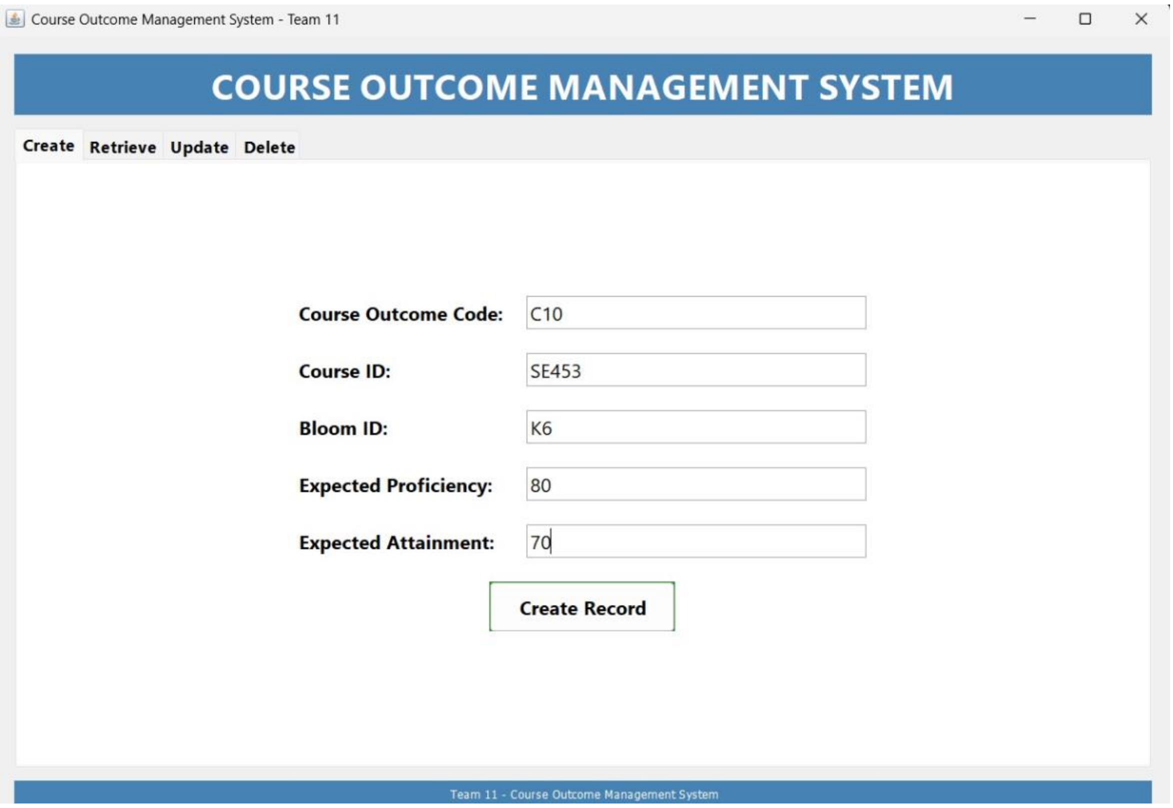
DBHelper.java

```
package courseoutcome;
import java.sql.*;
public class DBHelper {
    private static final String DB_URL = "jdbc:sqlite:course_outcome.db";
    static {
        try {
            Class.forName("org.sqlite.JDBC");
            Connection conn = getConnection();
            Statement stmt = conn.createStatement();
            String sql = "CREATE TABLE IF NOT EXISTS
Team11_course_outcome (" +
                    "id INTEGER PRIMARY KEY AUTOINCREMENT," +
                    "course_outcome_code TEXT UNIQUE," +
                    "course_id TEXT," +
                    "bloom_id TEXT," +
                    "expected_proficiency REAL," +
                    "expected_attainment REAL)";
            stmt.execute(sql);
            stmt.close();
            conn.close();
        } catch (ClassNotFoundException e) {
            System.err.println("SQLite JDBC driver not found");
            e.printStackTrace();
        } catch (SQLException e) {
            System.err.println("Error creating database table");
            e.printStackTrace();
        }
    }

    public static Connection getConnection() throws SQLException {
        return DriverManager.getConnection(DB_URL);
    }
```

Screenshots

Creation of data:-


Course Outcome Management System - Team 11
COURSE OUTCOME MANAGEMENT SYSTEM
Create
Retrieve
Update
Delete
Create Record
Team 11 - Course Outcome Management System

Course Outcome Code: C10

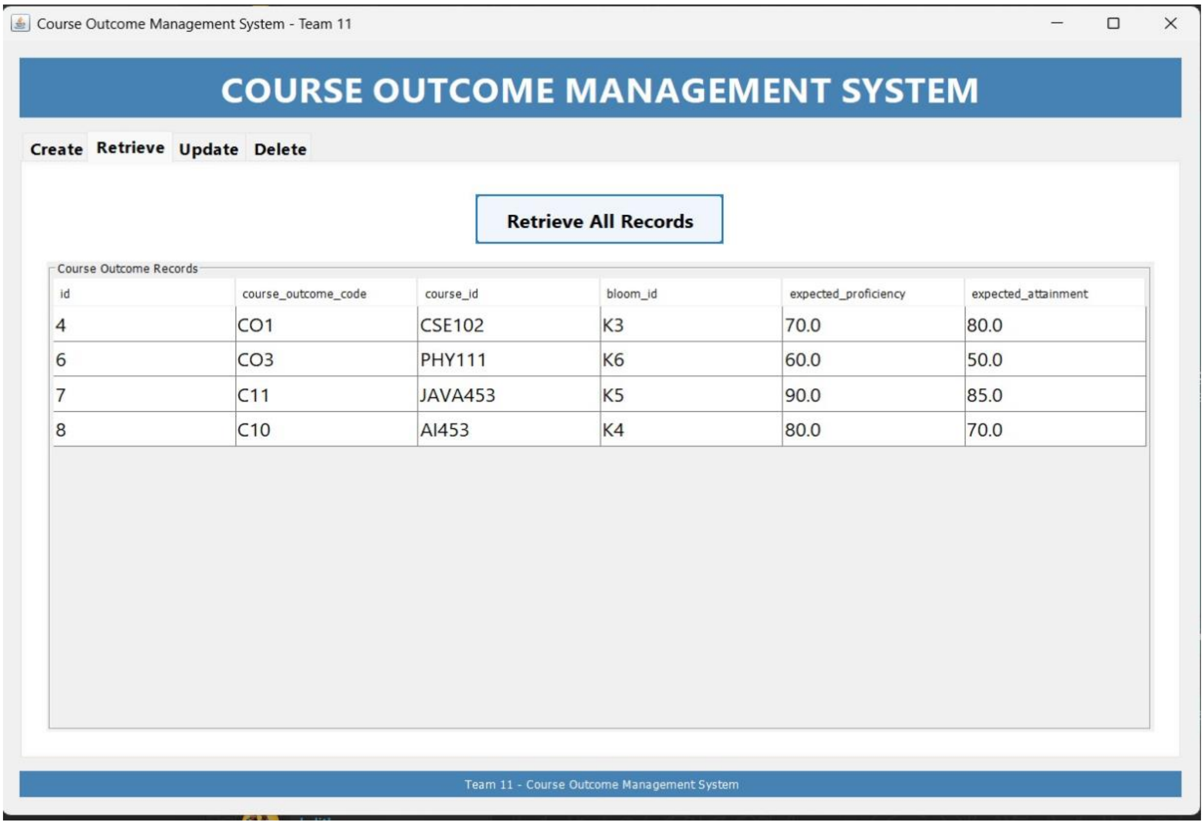
Course ID: SE453

Bloom ID: K6

Expected Proficiency: 80

Expected Attainment: 70

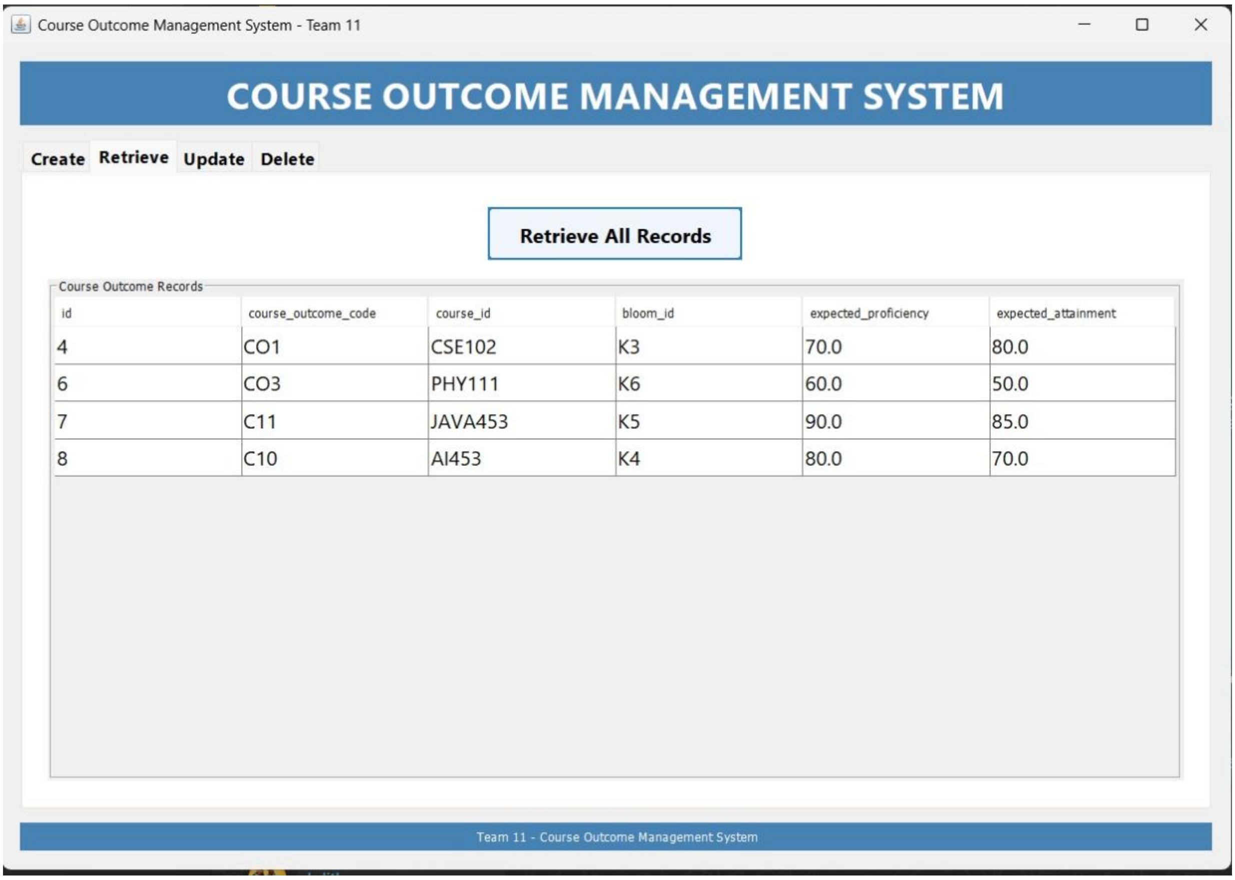
Retrieval of Data:-


Course Outcome Management System - Team 11
COURSE OUTCOME MANAGEMENT SYSTEM
Create
Retrieve
Update
Delete
Retrieve All Records
Team 11 - Course Outcome Management System

Course Outcome Records

id	course_outcome_code	course_id	bloom_id	expected_proficiency	expected_attainment
4	CO1	CSE102	K3	70.0	80.0
6	CO3	PHY111	K6	60.0	50.0
7	C11	JAVA453	K5	90.0	85.0
8	C10	AI453	K4	80.0	70.0

Before Updating:-



Course Outcome Management System - Team 11

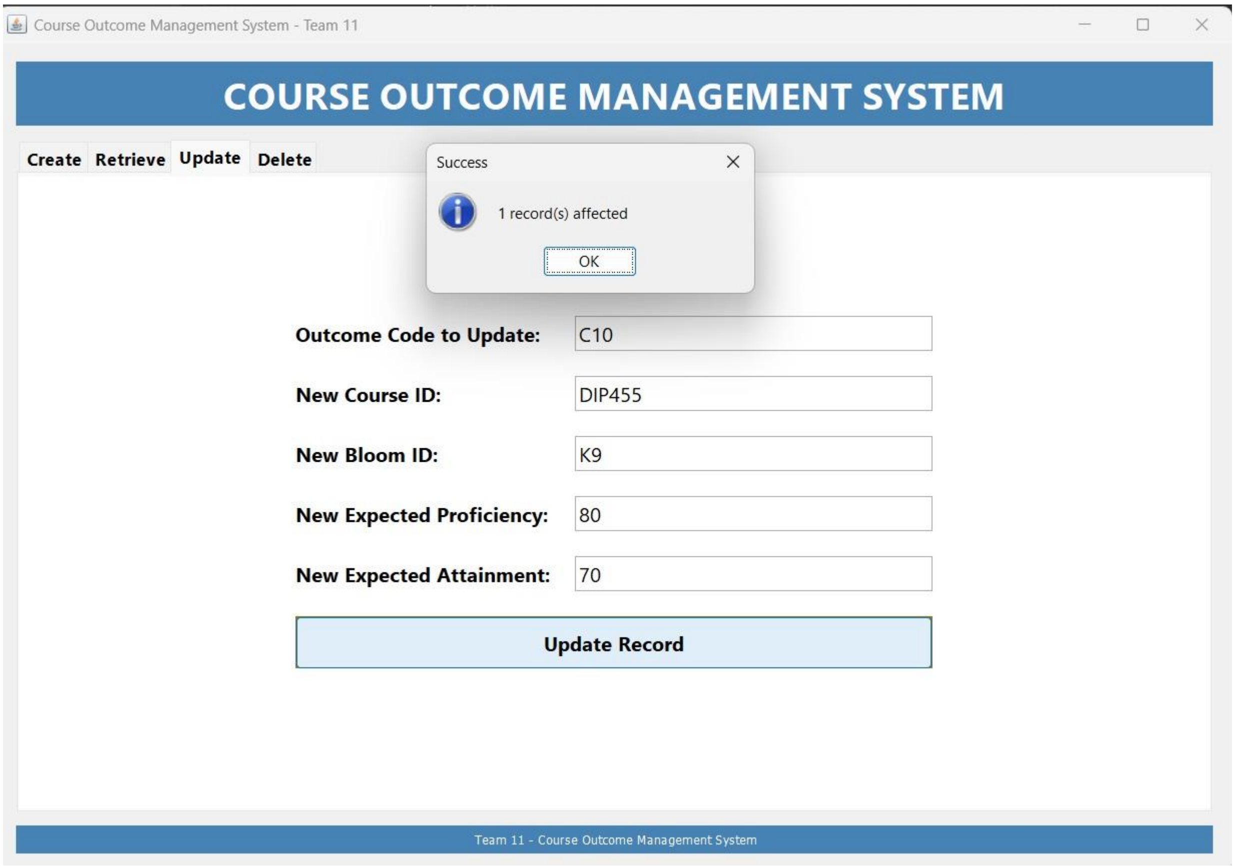
COURSE OUTCOME MANAGEMENT SYSTEM

Course Outcome Records

id	course_outcome_code	course_id	bloom_id	expected_proficiency	expected_attainment
4	CO1	CSE102	K3	70.0	80.0
6	CO3	PHY111	K6	60.0	50.0
7	C11	JAVA453	K5	90.0	85.0
8	C10	AI453	K4	80.0	70.0

Team 11 - Course Outcome Management System

Updating Process:-



Course Outcome Management System - Team 11

COURSE OUTCOME MANAGEMENT SYSTEM

Success

1 record(s) affected

Outcome Code to Update: C10

New Course ID: DIP455

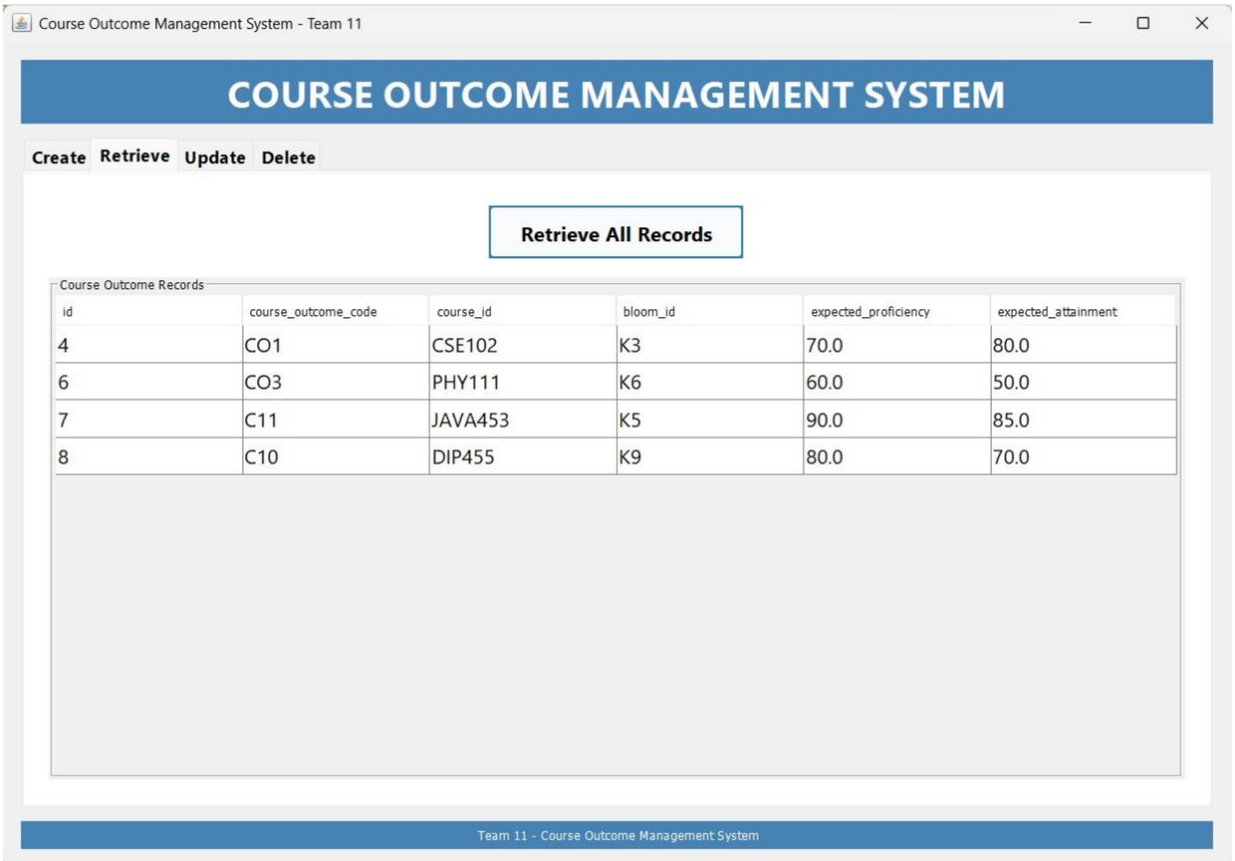
New Bloom ID: K9

New Expected Proficiency: 80

New Expected Attainment: 70

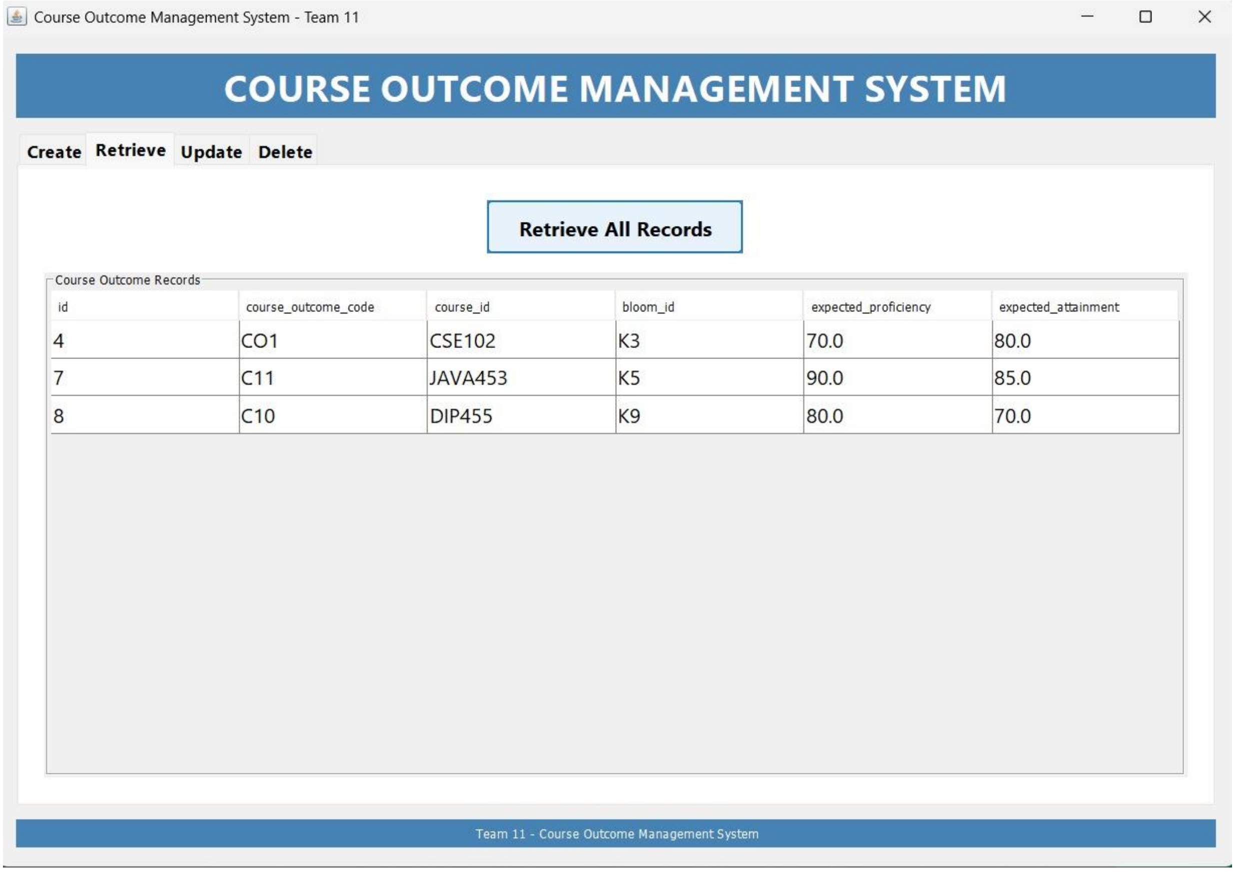
Team 11 - Course Outcome Management System

After Updating:-


Course Outcome Management System - Team 11
COURSE OUTCOME MANAGEMENT SYSTEM
Create
Retrieve
Update
Delete
Retrieve All Records
Course Outcome Records
Team 11 - Course Outcome Management System

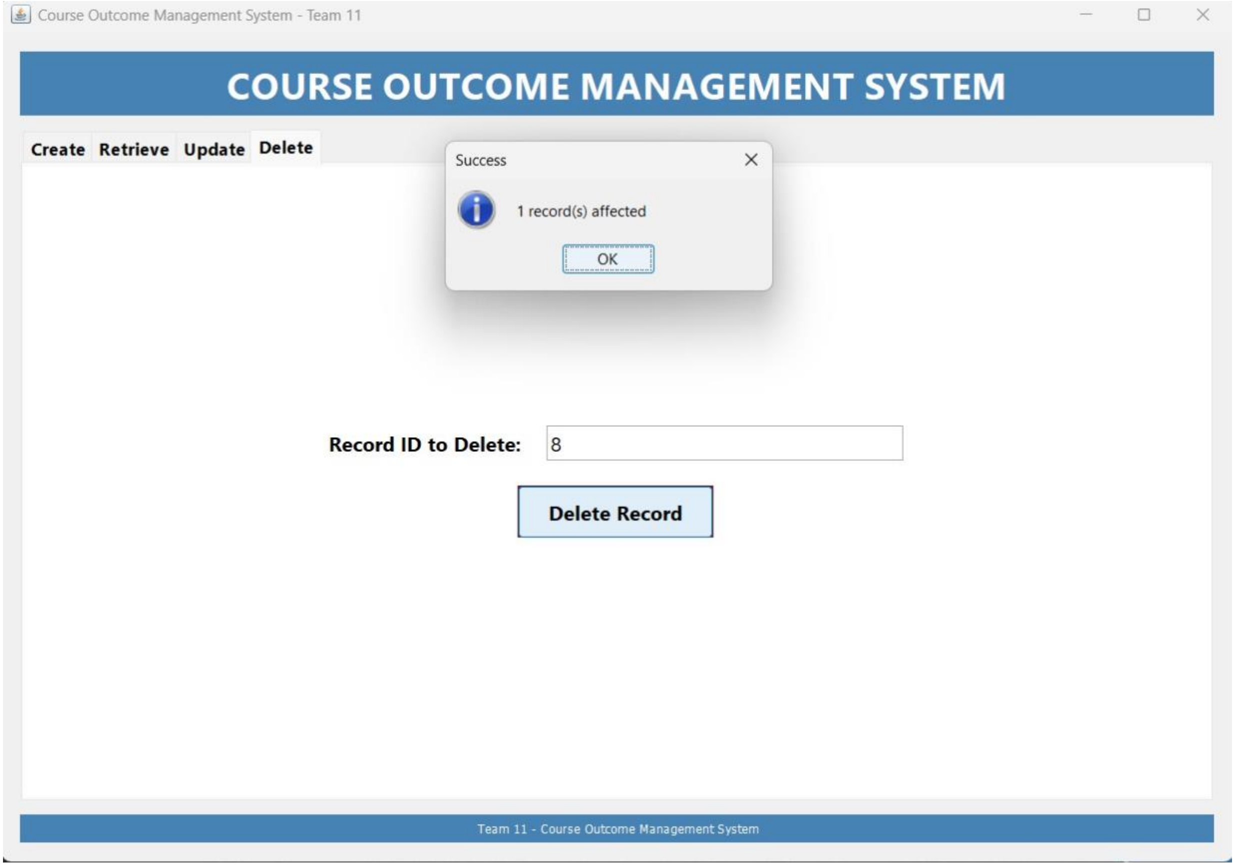
id	course_outcome_code	course_id	bloom_id	expected_proficiency	expected_attainment
4	CO1	CSE102	K3	70.0	80.0
6	CO3	PHY111	K6	60.0	50.0
7	C11	JAVA453	K5	90.0	85.0
8	C10	DIP455	K9	80.0	70.0

Before Deletion:-


Course Outcome Management System - Team 11
COURSE OUTCOME MANAGEMENT SYSTEM
Create
Retrieve
Update
Delete
Retrieve All Records
Course Outcome Records
Team 11 - Course Outcome Management System

id	course_outcome_code	course_id	bloom_id	expected_proficiency	expected_attainment
4	CO1	CSE102	K3	70.0	80.0
7	C11	JAVA453	K5	90.0	85.0
8	C10	DIP455	K9	80.0	70.0

Deleting Process:-



Course Outcome Management System - Team 11

COURSE OUTCOME MANAGEMENT SYSTEM

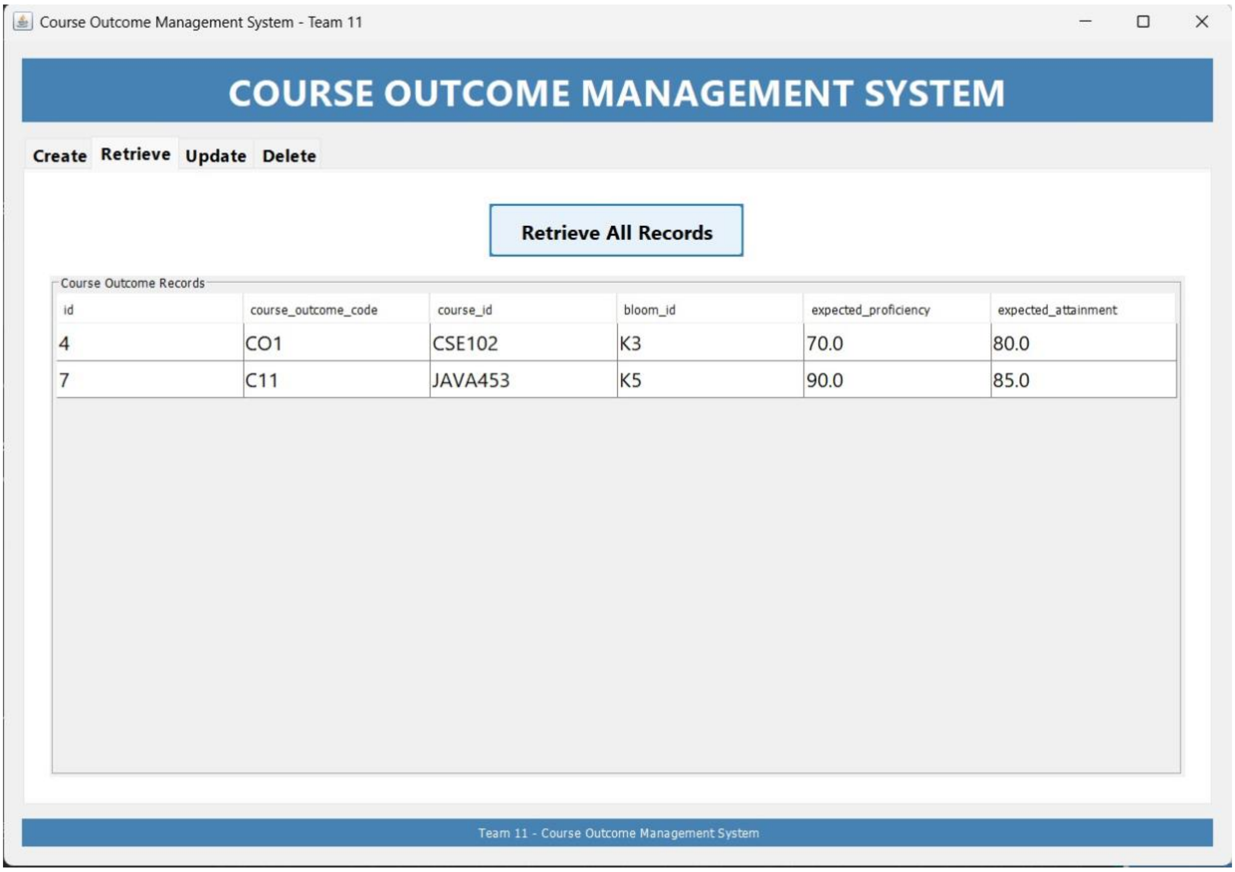
Success

1 record(s) affected

Record ID to Delete: 8

Team 11 - Course Outcome Management System

After Deletion:-



Course Outcome Management System - Team 11

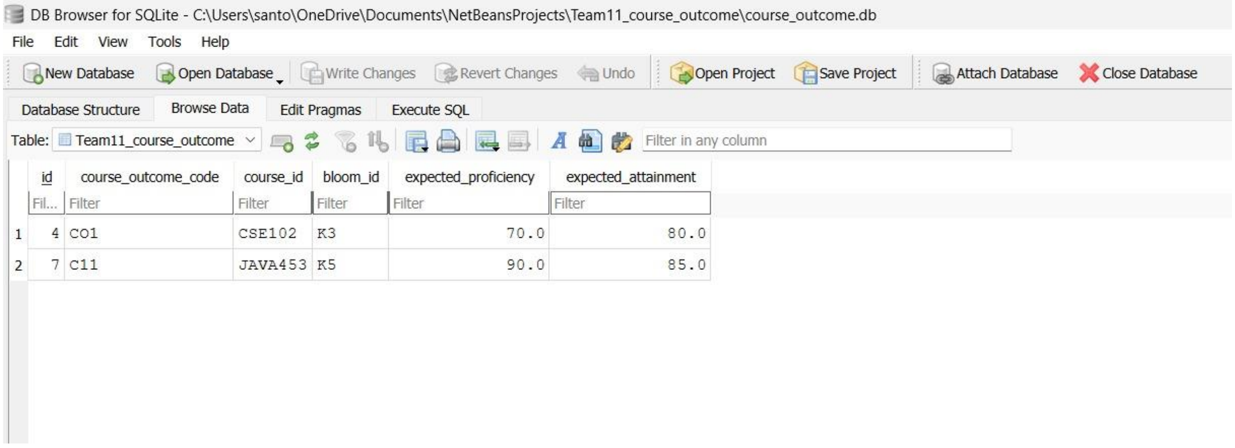
COURSE OUTCOME MANAGEMENT SYSTEM

Course Outcome Records

id	course_outcome_code	course_id	bloom_id	expected_proficiency	expected_attainment
4	CO1	CSE102	K3	70.0	80.0
7	C11	JAVA453	K5	90.0	85.0

Team 11 - Course Outcome Management System

DB Browser (SQL Lite):-



DB Browser for SQLite - C:\Users\santo\OneDrive\Documents\NetBeansProjects\Team11_course_outcome\course_outcome.db

Table: Team11_course_outcome

	id	course_outcome_code	course_id	bloom_id	expected_proficiency	expected_attainment
1	4	CO1	CSE102	K3	70.0	80.0
2	7	C11	JAVA453	K5	90.0	85.0

Conclusion:-

The **Course Outcome Setting** developed is a simple and user-friendly Java application designed to help manage course outcomes more efficiently. With an easy-to-use interface built using Swing and a solid backend using JDBC, it allows users to add, view, update, and delete course outcome records with ease. We made sure to include input validation and meaningful messages so that users get immediate feedback and avoid mistakes. The system displays all records clearly in a table, making it easy to understand and manage the data. Overall, this project helped us learn how to build practical software that can support academic tasks in a smooth and organized way.