OBE IMPLEMENTATION:COURSE OUTCOME SETTING

by

Group Name: Tech Titans

Srija Yadla [AP22110010108]

Shanmukha Krishna Chaitanya Munagala [AP22110010089]

Rammohan Rao Namburi [AP22110010082]

Hema Kakarlapudi [AP22110010080]

Venkata Naga Harshit Gulipali [AP22110010076]

A report for the CSE307:Mobile Application Development using JAVA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SRM UNIVERSITY AP::AMARAVATI

INDEX

Introduction

Our University (herewith considered as SRM-AP) is going to implement OBE(Outcome Based Education) in their university and you are assigned in the project to develop a CURD(Create,Update,Retrieve and Delete) windows and mobile application using JAVA programming and Android studio for the same.

Project Modules:

Various Modules available in the project are

1.Blooms Level setting

2.Program Level Objective Setting

3.University

4.Schools

5.Department

6.Programs

7.Courses

8.Course objective setting

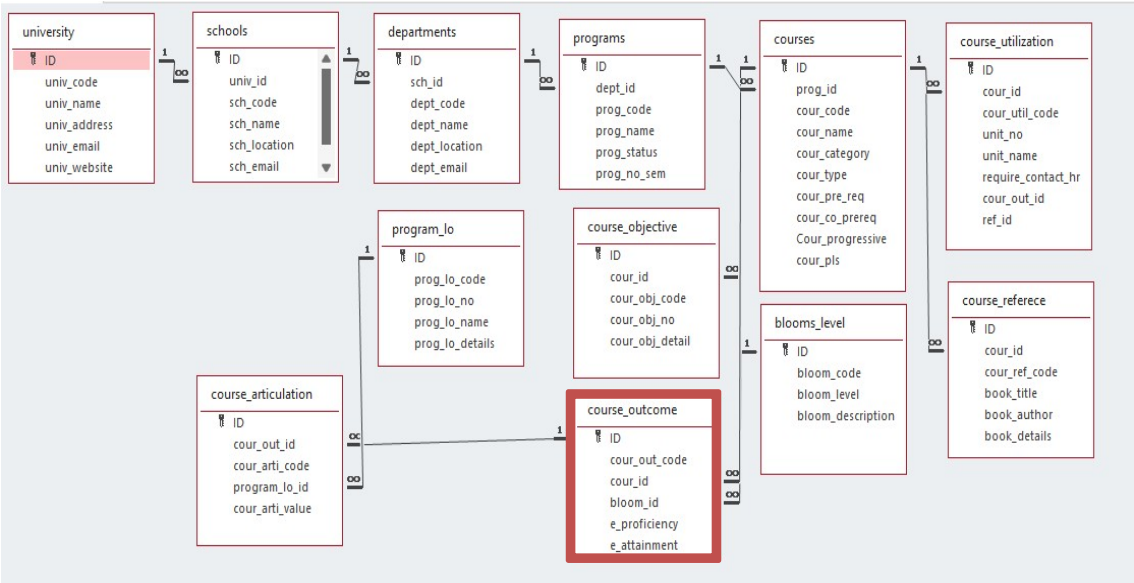
9.Course Outcome Setting

10.Course Articulation matrix Setting

11.Course Utilization Setting

12.Course Reference Setting.

Architecture Diagram



Module Description

Module Name: Course Outcome Setting

Module Description:

The Course Outcome module is responsible for managing course-related outcomes, including Bloom's taxonomy level, expected proficiency, and attainment. It enables the creation, update, retrieval, and deletion (CURD) of course outcome records and ensures efficient storage and access using a relational database like MySQL or SQLite. The module is developed in Java and interacts with the database through JDBC.

Programming Details naming conventions to be used:

class name/activity name: TechTitans_Course_Outcome

- **Function/method name**
  - **Create:** void insertData()
  - **Update:** void updateData()
  - **Retrieve:** void displayData()
  - **Delete:** void deleteData()

Table details:course_outcome

Field Name	**Data type**
Id	Integer (Primary Key, Auto-increment)
cour_out_code	String
cour_id	String(Foreign Key referencing course table)
bloom_id	String
e_proficiency	Real
e_attainment	Real

Source Code

```
import javax.swing.*;
import javax.swing.border.EmptyBorder;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.JTableHeader;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;

public class TechTitans extends JFrame {
    Connection conn;
    JTable table;
    DefaultTableModel model;
    JTextField tfCode, tfCID, tfBloom, tfProficiency, tfAttainment;

    public TechTitans() {
        setTitle("Course Outcome Manager");
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setSize(900, 500);
        setLocationRelativeTo(null);  // Center window
        setLayout(new BorderLayout());

        connect();

        // === TABLE SETUP ===
        model = new DefaultTableModel(new String[]{"ID", "Code", "CID", "Bloom",
"Proficiency", "Attainment"}, 0);
        table = new JTable(model);
        table.setFont(new Font("SansSerif", Font.PLAIN, 14));
        table.setRowHeight(28);

        JTableHeader header = table.getTableHeader();
        header.setFont(new Font("SansSerif", Font.BOLD, 16));
        header.setBackground(new Color(200, 220, 240));

        JScrollPane scrollPane = new JScrollPane(table);
        add(scrollPane, BorderLayout.CENTER);

        // === FORM PANEL ===
        // Using BorderLayout for the main form panel
        JPanel formPanel = new JPanel(new BorderLayout(10, 10));
        formPanel.setBorder(new EmptyBorder(10, 10, 10, 10));
        formPanel.setBackground(new Color(245, 250, 255));

        // Panel for input fields
        JPanel inputPanel = new JPanel(new GridLayout(2, 5, 10, 5));
        inputPanel.setBackground(new Color(245, 250, 255));

        // Panel for buttons
```

```
        JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.CENTER, 15, 5));
        buttonPanel.setBackground(new Color(245, 250, 255));

        tfCode = new JTextField(); tfCID = new JTextField(); tfBloom = new JTextField();
        tfProficiency = new JTextField(); tfAttainment = new JTextField();

        // Add labels to input panel
        inputPanel.add(new JLabel("Code"));
        inputPanel.add(new JLabel("CID"));
        inputPanel.add(new JLabel("Bloom"));
        inputPanel.add(new JLabel("Proficiency"));
        inputPanel.add(new JLabel("Attainment"));

        // Add input fields to input panel
        inputPanel.add(tfCode);
        inputPanel.add(tfCID);
        inputPanel.add(tfBloom);
        inputPanel.add(tfProficiency);
        inputPanel.add(tfAttainment);

        // Buttons
        JButton btnAdd = new JButton("Add");
        JButton btnUpdate = new JButton("Update");
        JButton btnDelete = new JButton("Delete");

        Font btnFont = new Font("SansSerif", Font.BOLD, 14);
        btnAdd.setFont(btnFont);
        btnUpdate.setFont(btnFont);
        btnDelete.setFont(btnFont);

        // Add buttons to button panel
        buttonPanel.add(btnAdd);
        buttonPanel.add(btnUpdate);
        buttonPanel.add(btnDelete);

        // Add input and button panels to the main form panel
        formPanel.add(inputPanel, BorderLayout.NORTH);
        formPanel.add(buttonPanel, BorderLayout.CENTER);

        // Add form panel to the frame
        add(formPanel, BorderLayout.SOUTH);

        displayData();

        // === BUTTON EVENTS ===
        btnAdd.addActionListener(e -> {
            try {
                String sql = "INSERT INTO course_outcome (cour_out_code, cour_id, bloom_id,
e_proficiency, e_attainment) VALUES (?, ?, ?, ?, ?)";
                PreparedStatement pst = conn.prepareStatement(sql);
```

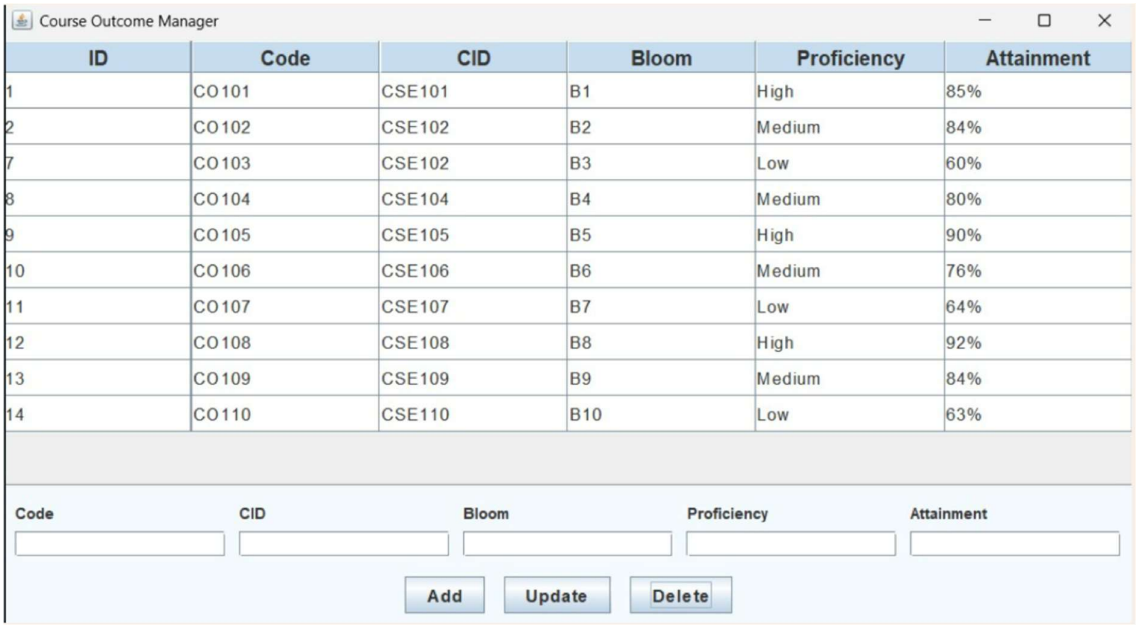
```
            pst.setString(1, tfCode.getText());
            pst.setString(2, tfCID.getText());
            pst.setString(3, tfBloom.getText());
            pst.setString(4, tfProficiency.getText());
            pst.setString(5, tfAttainment.getText());
            pst.executeUpdate();
            displayData();
            clearFields();
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    });

    btnUpdate.addActionListener(e -> {
        int selected = table.getSelectedRow();
        if (selected >= 0) {
            try {
                String id = model.getValueAt(selected, 0).toString();
                String sql = "UPDATE course_outcome SET cour_out_code=?, cour_id=?,
bloom_id=?, e_proficiency=?, e_attainment=? WHERE id=?";
                PreparedStatement pst = conn.prepareStatement(sql);
                pst.setString(1, tfCode.getText());
                pst.setString(2, tfCID.getText());
                pst.setString(3, tfBloom.getText());
                pst.setString(4, tfProficiency.getText());
                pst.setString(5, tfAttainment.getText());
                pst.setInt(6, Integer.parseInt(id));
                pst.executeUpdate();
                displayData();
                clearFields();
            } catch (Exception ex) {
                ex.printStackTrace();
            }
        }
    });

    btnDelete.addActionListener(e -> {
        int selected = table.getSelectedRow();
        if (selected >= 0) {
            try {
                String id = model.getValueAt(selected, 0).toString();
                String sql = "DELETE FROM course_outcome WHERE id=?";
                PreparedStatement pst = conn.prepareStatement(sql);
                pst.setInt(1, Integer.parseInt(id));
                pst.executeUpdate();
                displayData();
                clearFields();
            } catch (Exception ex) {
                ex.printStackTrace();
            }
```

```
        }
    });

    table.addMouseListener(new MouseAdapter() {
        public void mouseClicked(MouseEvent e) {
            int i = table.getSelectedRow();
            tfCode.setText(model.getValueAt(i, 1).toString());
            tfCID.setText(model.getValueAt(i, 2).toString());
            tfBloom.setText(model.getValueAt(i, 3).toString());
            tfProficiency.setText(model.getValueAt(i, 4).toString());
            tfAttainment.setText(model.getValueAt(i, 5).toString());
        }
    });

    setVisible(true);
}

void connect() {
    try {
        conn = DriverManager.getConnection("jdbc:sqlite:javaapp.db");
    } catch (Exception e) {
        e.printStackTrace();
    }
}

void displayData() {
    model.setRowCount(0);
    try {
        Statement stmt = conn.createStatement();
        ResultSet rs = stmt.executeQuery("SELECT * FROM course_outcome");
        while (rs.next()) {
            model.addRow(new Object[]{
                    rs.getInt("id"),
                    rs.getString("cour_out_code"),
                    rs.getString("cour_id"),
                    rs.getString("bloom_id"),
                    rs.getString("e_proficiency"),
                    rs.getString("e_attainment")
            });
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

void clearFields() {
    tfCode.setText("");
    tfCID.setText("");
    tfBloom.setText("");
    tfProficiency.setText("");
```

```
        tfAttainment.setText("");
    }

    public static void main(String[] args) {
        new TechTitans();
    }
}
```

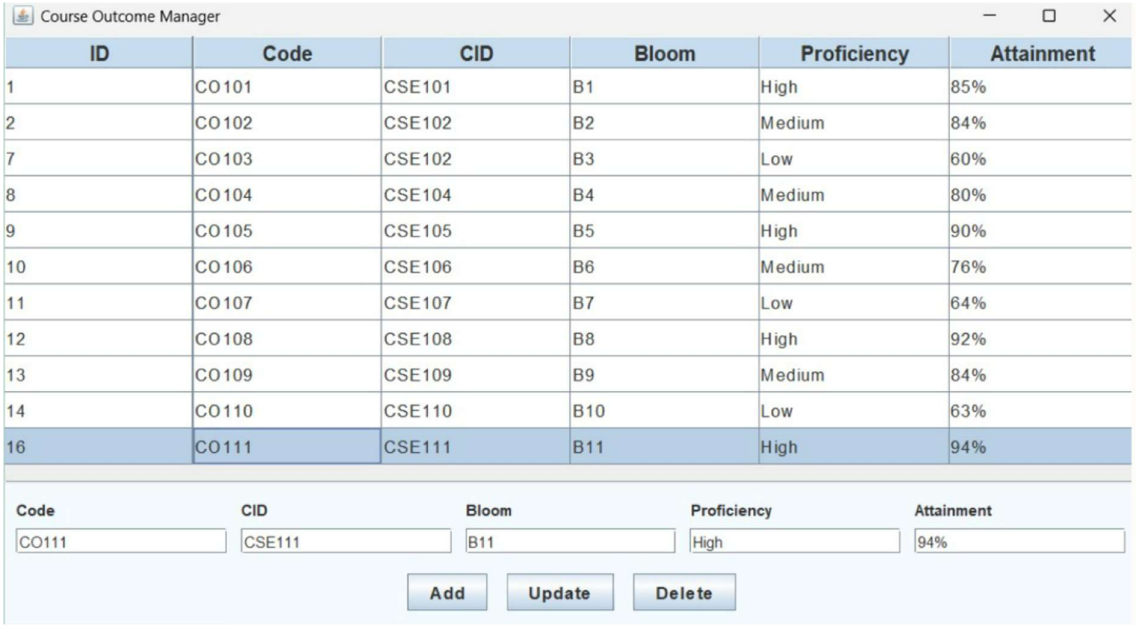
Screen Shots

On Loading the window:Already Stored Records in Databases are Displayed



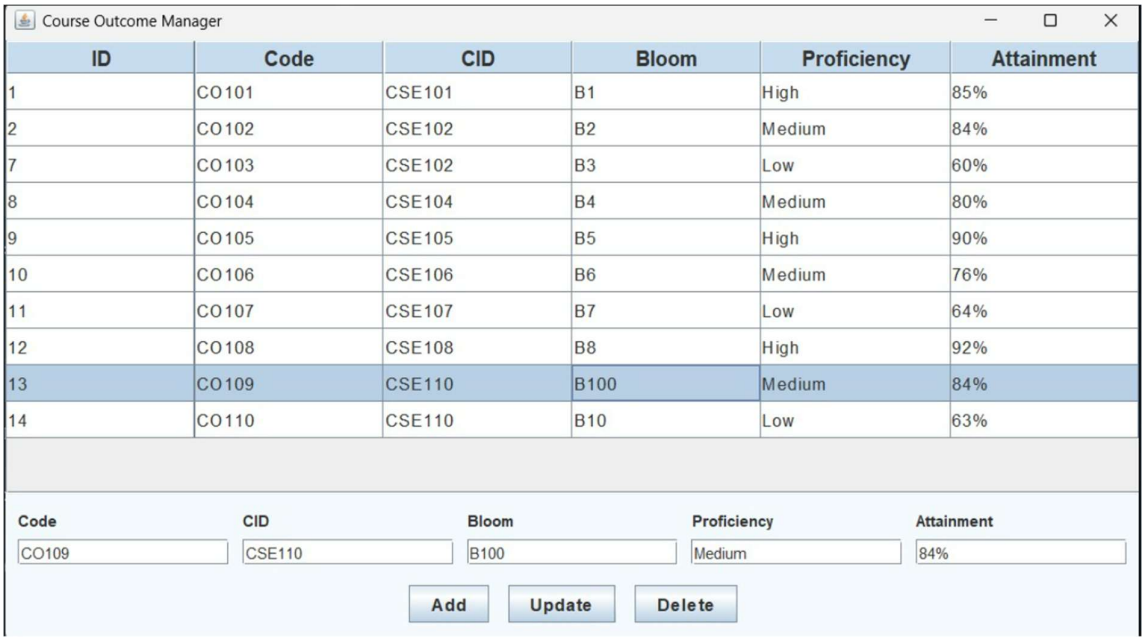
ID	Code	CID	Bloom	Proficiency	Attainment
1	CO101	CSE101	B1	High	85%
2	CO102	CSE102	B2	Medium	84%
7	CO103	CSE102	B3	Low	60%
8	CO104	CSE104	B4	Medium	80%
9	CO105	CSE105	B5	High	90%
10	CO106	CSE106	B6	Medium	76%
11	CO107	CSE107	B7	Low	64%
12	CO108	CSE108	B8	High	92%
13	CO109	CSE109	B9	Medium	84%
14	CO110	CSE110	B10	Low	63%

ADDING a Record with ID : 16, CODE:CO111



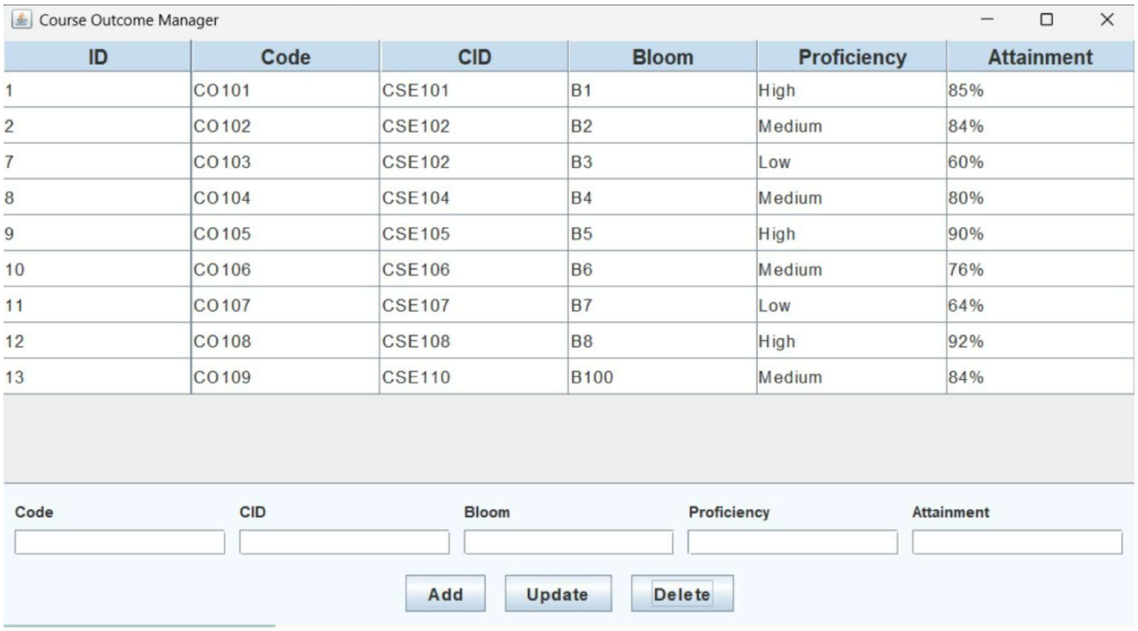
ID	Code	CID	Bloom	Proficiency	Attainment
1	CO101	CSE101	B1	High	85%
2	CO102	CSE102	B2	Medium	84%
7	CO103	CSE102	B3	Low	60%
8	CO104	CSE104	B4	Medium	80%
9	CO105	CSE105	B5	High	90%
10	CO106	CSE106	B6	Medium	76%
11	CO107	CSE107	B7	Low	64%
12	CO108	CSE108	B8	High	92%
13	CO109	CSE109	B9	Medium	84%
14	CO110	CSE110	B10	Low	63%
16	CO111	CSE111	B11	High	94%

Updating a Record ID-13 BLOOM is updated to B100



ID	Code	CID	Bloom	Proficiency	Attainment
1	CO101	CSE101	B1	High	85%
2	CO102	CSE102	B2	Medium	84%
7	CO103	CSE102	B3	Low	60%
8	CO104	CSE104	B4	Medium	80%
9	CO105	CSE105	B5	High	90%
10	CO106	CSE106	B6	Medium	76%
11	CO107	CSE107	B7	Low	64%
12	CO108	CSE108	B8	High	92%
13	CO109	CSE110	B100	Medium	84%
14	CO110	CSE110	B10	Low	63%

Delete: ID-14 Record is deleted



ID	Code	CID	Bloom	Proficiency	Attainment
1	CO101	CSE101	B1	High	85%
2	CO102	CSE102	B2	Medium	84%
7	CO103	CSE102	B3	Low	60%
8	CO104	CSE104	B4	Medium	80%
9	CO105	CSE105	B5	High	90%
10	CO106	CSE106	B6	Medium	76%
11	CO107	CSE107	B7	Low	64%
12	CO108	CSE108	B8	High	92%
13	CO109	CSE110	B100	Medium	84%

Conclusion:

The TechTitans Course Outcome Setting module effectively demonstrates CRUD operations using Java and SQLite via JDBC. The system enables seamless interaction between the application and a local database, supporting real-time operations for outcome management. This project enhanced our understanding of Java-JDBC integration, data persistence, UI-based CRUD functionalities, and structured exception handling. It emphasizes the practical implementation of database-driven applications and the significance of modular code design in building scalable software systems.