# ELECTRICITY BILLING SYSTEM

**PROGRAM**:-

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class ElectricityBillingSystem extends JFrame {

    private JTextField tfName, tfAddress, tfContact, tfEmail, tfUnits, tfBillAmount, tfMeterNumber;

    private JButton btnCalculate, btnClear, btnDelete, btnUpdate;

    private JLabel lblName, lblAddress, lblContact, lblEmail, lblUnits, lblBillAmount, lblMeterNumber, lblTitle;

    private DefaultTableModel tableModel;

    private JTable resultsTable;

    private int selectedRow = -1; // Variable to track the selected row

    public ElectricityBillingSystem() {

        setTitle("Electricity Billing System");

        setLayout(new BorderLayout(10, 10));

        setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        setSize(700, 500);

        // Create the tabbed pane

        JTabbedPane tabbedPane = new JTabbedPane();

        // First tab: Billing System

        JPanel billingPanel = createBillingPanel();

        tabbedPane.addTab("Billing System", billingPanel);

        // Second tab: Results Table

        JPanel resultsPanel = createResultsPanel();

        tabbedPane.addTab("Results", resultsPanel);

        add(tabbedPane, BorderLayout.CENTER);

        setLocationRelativeTo(null);  // Center window

        setVisible(true);

    }

    private JPanel createBillingPanel() {

        JPanel billingPanel = new JPanel();

        billingPanel.setLayout(new BorderLayout());

        // Title label at the top with custom background color

        lblTitle = new JLabel("Electricity Billing System", JLabel.CENTER);

        lblTitle.setFont(new Font("Arial", Font.BOLD, 18));

        lblTitle.setForeground(Color.WHITE);

        lblTitle.setBackground(new Color(0, 102, 204));

        lblTitle.setOpaque(true);

        billingPanel.add(lblTitle, BorderLayout.NORTH);

        // Panel for customer details (center)

        JPanel panelCenter = new JPanel();

        panelCenter.setLayout(new GridLayout(8, 2, 10, 10)); // Added one more row for meter number

        panelCenter.setBackground(new Color(240, 240, 240));  // Light gray background

        billingPanel.add(panelCenter, BorderLayout.CENTER);

        // Creating labels and text fields with consistent styling

        lblName = new JLabel("Customer Name:");

        lblName.setFont(new Font("Arial", Font.PLAIN, 14));

        lblName.setForeground(new Color(0, 102, 204));

        tfName = new JTextField();

        styleTextField(tfName);

        panelCenter.add(lblName);

        panelCenter.add(tfName);

        lblAddress = new JLabel("Customer Address:");

        lblAddress.setFont(new Font("Arial", Font.PLAIN, 14));

        lblAddress.setForeground(new Color(0, 102, 204));

        tfAddress = new JTextField();

        styleTextField(tfAddress);

        panelCenter.add(lblAddress);

        panelCenter.add(tfAddress);

        lblContact = new JLabel("Contact Number:");

        lblContact.setFont(new Font("Arial", Font.PLAIN, 14));

        lblContact.setForeground(new Color(0, 102, 204));

        tfContact = new JTextField();

        styleTextField(tfContact);

        panelCenter.add(lblContact);

        panelCenter.add(tfContact);

        lblEmail = new JLabel("Email:");

        lblEmail.setFont(new Font("Arial", Font.PLAIN, 14));

        lblEmail.setForeground(new Color(0, 102, 204));

        tfEmail = new JTextField();

        styleTextField(tfEmail);

        panelCenter.add(lblEmail);

        panelCenter.add(tfEmail);

        lblMeterNumber = new JLabel("Meter Number:");

        lblMeterNumber.setFont(new Font("Arial", Font.PLAIN, 14));

        lblMeterNumber.setForeground(new Color(0, 102, 204));

        tfMeterNumber = new JTextField();

        styleTextField(tfMeterNumber);

        panelCenter.add(lblMeterNumber);

        panelCenter.add(tfMeterNumber);

        lblUnits = new JLabel("Units Consumed:");

        lblUnits.setFont(new Font("Arial", Font.PLAIN, 14));

        lblUnits.setForeground(new Color(0, 102, 204));

        tfUnits = new JTextField();

        styleTextField(tfUnits);

        panelCenter.add(lblUnits);

        panelCenter.add(tfUnits);

        lblBillAmount = new JLabel("Bill Amount: ");

        lblBillAmount.setFont(new Font("Arial", Font.PLAIN, 14));

        lblBillAmount.setForeground(new Color(0, 102, 204));

        tfBillAmount = new JTextField();

        tfBillAmount.setEditable(false);

        styleTextField(tfBillAmount);

        panelCenter.add(lblBillAmount);

        panelCenter.add(tfBillAmount);

        // Panel for buttons (bottom)

        JPanel panelSouth = new JPanel();

        panelSouth.setLayout(new FlowLayout());

        panelSouth.setBackground(new Color(0, 102, 204));

        billingPanel.add(panelSouth, BorderLayout.SOUTH);

        // Calculate button with custom background and font

        btnCalculate = new JButton("Calculate Bill");

        btnCalculate.setFont(new Font("Arial", Font.BOLD, 14));

        btnCalculate.setBackground(new Color(0, 204, 102));

        btnCalculate.setForeground(Color.WHITE);

        btnCalculate.setBorder(BorderFactory.createLineBorder(Color.BLACK, 1));

        // Clear button with custom background and font

        btnClear = new JButton("Clear");

        btnClear.setFont(new Font("Arial", Font.BOLD, 14));

        btnClear.setBackground(new Color(255, 77, 77));

        btnClear.setForeground(Color.WHITE);

        btnClear.setBorder(BorderFactory.createLineBorder(Color.BLACK, 1));

        panelSouth.add(btnCalculate);

        panelSouth.add(btnClear);

        btnCalculate.addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent e) {

                try {

                    String customerName = tfName.getText();

                    String customerAddress = tfAddress.getText();

                    String contactNumber = tfContact.getText();

                    String email = tfEmail.getText();

                    String meterNumber = tfMeterNumber.getText();

                    int unitsConsumed = Integer.parseInt(tfUnits.getText());

                    if (customerName.isEmpty() || customerAddress.isEmpty() || contactNumber.isEmpty()

                            || email.isEmpty() || meterNumber.isEmpty() || unitsConsumed < 0) {

                        JOptionPane.showMessageDialog(null, "Please enter valid information.");

                        return;

                    }

                    double billAmount = calculateBill(unitsConsumed);

                    tfBillAmount.setText("Rs. " + String.format("%.2f", billAmount));

                    // Add bill details as a new row in the table

                    Object[] rowData = {customerName, meterNumber, unitsConsumed, String.format("Rs. %.2f", billAmount)};

                    tableModel.addRow(rowData);

                } catch (NumberFormatException ex) {

                    JOptionPane.showMessageDialog(null, "Invalid input. Please enter numeric values for units.");

                }

            }

        });

        btnClear.addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent e) {

                tfName.setText("");

                tfAddress.setText("");

                tfContact.setText("");

                tfEmail.setText("");

                tfMeterNumber.setText("");

                tfUnits.setText("");

                tfBillAmount.setText("");

            }

        });

        return billingPanel;

    }

    private JPanel createResultsPanel() {

        JPanel resultsPanel = new JPanel();

        resultsPanel.setLayout(new BorderLayout());

        // JTable to display results in table format

        String[] columnNames = {"Customer Name", "Meter Number", "Units Consumed", "Bill Amount"};

        tableModel = new DefaultTableModel(columnNames, 0);

        resultsTable = new JTable(tableModel);

        JScrollPane scrollPane = new JScrollPane(resultsTable);

        resultsPanel.add(scrollPane, BorderLayout.CENTER);

        // Panel for Delete and Update buttons

        JPanel buttonPanel = new JPanel();

        buttonPanel.setLayout(new FlowLayout());

        resultsPanel.add(buttonPanel, BorderLayout.SOUTH);

        btnDelete = new JButton("Delete Selected Row");

        btnUpdate = new JButton("Update Selected Row");

        buttonPanel.add(btnDelete);

        buttonPanel.add(btnUpdate);

        // Delete button action

        btnDelete.addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent e) {

                int row = resultsTable.getSelectedRow();

                if (row != -1) {

                    tableModel.removeRow(row);

                } else {

                    JOptionPane.showMessageDialog(null, "Please select a row to delete.");

                }

            }

        });

        // Update button action

        btnUpdate.addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent e) {

                int row = resultsTable.getSelectedRow();

                if (row != -1) {

                    // Retrieve data from selected row

                    String customerName = (String) tableModel.getValueAt(row, 0);

                    String meterNumber = (String) tableModel.getValueAt(row, 1);

                    int unitsConsumed = (int) tableModel.getValueAt(row, 2);

                    String billAmount = (String) tableModel.getValueAt(row, 3);

                    // Set the data in the form fields

                    tfName.setText(customerName);

                    tfMeterNumber.setText(meterNumber);

                    tfUnits.setText(String.valueOf(unitsConsumed));

                    tfBillAmount.setText(billAmount);

                    // Update the row with the new data when user calculates a new bill

                    btnCalculate.addActionListener(new ActionListener() {

                        @Override

                        public void actionPerformed(ActionEvent e) {

                            try {

                                String newCustomerName = tfName.getText();

                                String newMeterNumber = tfMeterNumber.getText();

                                int newUnitsConsumed = Integer.parseInt(tfUnits.getText());

                                if (newCustomerName.isEmpty() || newMeterNumber.isEmpty() || newUnitsConsumed < 0) {

                                    JOptionPane.showMessageDialog(null, "Please enter valid information.");

                                    return;

                                }

                                double newBillAmount = calculateBill(newUnitsConsumed);

                                tfBillAmount.setText("Rs. " + String.format("%.2f", newBillAmount));

                                // Update the row with new values

                                tableModel.setValueAt(newCustomerName, row, 0);

                                tableModel.setValueAt(newMeterNumber, row, 1);

                                tableModel.setValueAt(newUnitsConsumed, row, 2);

                                tableModel.setValueAt(String.format("Rs. %.2f", newBillAmount), row, 3);

                            } catch (NumberFormatException ex) {

                                JOptionPane.showMessageDialog(null, "Invalid input. Please enter numeric values for units.");

                            }

                        }

                    });

                } else {

                    JOptionPane.showMessageDialog(null, "Please select a row to update.");

                }

            }

        });

        return resultsPanel;

    }

    private double calculateBill(int units) {

        double billAmount = 0;

        if (units <= 100) {

            billAmount = units \* 5.0;

        } else if (units <= 200) {

            billAmount = 100 \* 5.0 + (units - 100) \* 7.0;

        } else if (units <= 500) {

            billAmount = 100 \* 5.0 + 100 \* 7.0 + (units - 200) \* 10.0;

        } else {

            billAmount = 100 \* 5.0 + 100 \* 7.0 + 300 \* 10.0 + (units - 500) \* 15.0;

        }

        return billAmount;

    }

    // Method to style text fields with consistent look

    private void styleTextField(JTextField textField) {

        textField.setFont(new Font("Arial", Font.PLAIN, 14));

        textField.setBackground(Color.WHITE);

        textField.setForeground(Color.BLACK);

        textField.setBorder(BorderFactory.createLineBorder(new Color(0, 102, 204), 2));

    }

    public static void main(String[] args) {

        new ElectricityBillingSystem();

    }

}