CODINGS FOR MINI PROJECTS

```
package dao;


import model.Employee;
import java.sql.*;
import java.util.ArrayList;
import java.util.List;


public class EmployeeDAO {
    private Connection conn;


    public EmployeeDAO(Connection conn) {
        this.conn = conn;
    }


    public void addEmployee(Employee emp) throws SQLException {
        String sql = "INSERT INTO Employee (name, department, designation, salary) VALUES (?,
?, ?, ?)";
        PreparedStatement stmt = conn.prepareStatement(sql);
        stmt.setString(1, emp.getName());
        stmt.setString(2, emp.getDepartment());
        stmt.setString(3, emp.getDesignation());
        stmt.setDouble(4, emp.getSalary());
        stmt.executeUpdate();
    }


    public void updateEmployee(Employee emp) throws SQLException {
        String sql = "UPDATE Employee SET name=?, department=?, designation=?, salary=?
WHERE id=?";
```

```
    PreparedStatement stmt = conn.prepareStatement(sql);
    stmt.setString(1, emp.getName());
    stmt.setString(2, emp.getDepartment());
    stmt.setString(3, emp.getDesignation());
    stmt.setDouble(4, emp.getSalary());
    stmt.setInt(5, emp.getId());
    stmt.executeUpdate();
}

public void deleteEmployee(int id) throws SQLException {
    String sql = "DELETE FROM Employee WHERE id=?";
    PreparedStatement stmt = conn.prepareStatement(sql);
    stmt.setInt(1, id);
    stmt.executeUpdate();
}

public List<Employee> getAllEmployees() throws SQLException {
    List<Employee> list = new ArrayList<>();
    String sql = "SELECT * FROM Employee";
    Statement stmt = conn.createStatement();
    ResultSet rs = stmt.executeQuery(sql);
    while (rs.next()) {
        Employee emp = new Employee(
                rs.getInt("id"),
                rs.getString("name"),
                rs.getString("department"),
                rs.getString("designation"),
                rs.getDouble("salary")
```

```
            );
            list.add(emp);
        }
        return list;
    }

    public boolean validateLogin(String username, String password) throws SQLException {
        String sql = "SELECT * FROM users WHERE username=? AND password=?";
        PreparedStatement stmt = conn.prepareStatement(sql);
        stmt.setString(1, username);
        stmt.setString(2, password);
        ResultSet rs = stmt.executeQuery();
        return rs.next(); // returns true if a match is found
    }

}

package db;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DBConnection {
    public static Connection getConnection() throws SQLException {
        String url = "jdbc:mysql://localhost:3306/payroll";
        String user = "root"; // replace with your MySQL username
        String password = "Qwe!asd@zxc#poi$lkj%mnb"; // replace with your password
```

```
        return DriverManager.getConnection(url, user, password);
    }
}

public class LoginTester {
    public static void main(String[] args) {
        try (Connection conn = DBConnection.getConnection()) {
            String sql = "SELECT * FROM users WHERE username=? AND password=?";
            PreparedStatement stmt = conn.prepareStatement(sql);
            stmt.setString(1, "admin");
            stmt.setString(2, "admin123");
            ResultSet rs = stmt.executeQuery();

            if (rs.next()) {
                System.out.println("✅ Login successful!");
            } else {
                System.out.println("❌ Login failed.");
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

LOGIN

```
package main;

import ui.LoginForm;
```

```
public class Main {
    public static void main(String[] args) {
        // Launch the login screen on the Event Dispatch Thread
        javax.swing.SwingUtilities.invokeLater(() -> new LoginForm());
    }
}
```

```
package model;

public class Employee {
    private int id;
    private String name;
    private String department;
    private String designation;
    private double salary;

    public Employee(int id, String name, String department, String designation, double salary)
{
        this.id = id;
        this.name = name;
        this.department = department;
        this.designation = designation;
        this.salary = salary;
    }

    public int getId() { return id; }
    public String getName() { return name; }
```

```
    public String getDepartment() { return department; }
    public String getDesignation() { return designation; }
    public double getSalary() { return salary; }

    public double getNetSalary() {
        double tax = salary * 0.1;
        double pf = salary * 0.05;
        return salary - tax - pf;
    }
}

package ui;

import dao.EmployeeDAO;
import db.DBConnection;
import model.Employee;

import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.awt.event.*;
import java.sql.Connection;
import java.util.List;

public class EmployeeForm extends JFrame {
    private JTextField nameField, departmentField, designationField, salaryField;
    private JButton addButton, updateButton, deleteButton;
    private JTable table;
```

```
private DefaultTableModel model;
private int selectedId = -1;

public EmployeeForm() {
    setTitle("Employee Payroll System");
    setSize(800, 500);
    setDefaultCloseOperation(EXIT_ON_CLOSE);
    setLocationRelativeTo(null);
    setLayout(new BorderLayout());

    // Form Panel
    JPanel formPanel = new JPanel(new GridLayout(5, 2, 10, 10));
    formPanel.setBorder(BorderFactory.createTitledBorder("Employee Details"));
    nameField = new JTextField();
    departmentField = new JTextField();
    designationField = new JTextField();
    salaryField = new JTextField();
    formPanel.add(new JLabel("Name")); formPanel.add(nameField);
    formPanel.add(new JLabel("Department")); formPanel.add(departmentField);
    formPanel.add(new JLabel("Designation")); formPanel.add(designationField);
    formPanel.add(new JLabel("Salary")); formPanel.add(salaryField);

    addButton = new JButton("Add");
    updateButton = new JButton("Update");
    deleteButton = new JButton("Delete");
    formPanel.add(addButton); formPanel.add(updateButton);
    add(formPanel, BorderLayout.NORTH);
```

```
        // Table
        model = new DefaultTableModel(new String[]{"ID", "Name", "Department", "Designation", "Salary", "Net Salary"}, 0);
        table = new JTable(model);
        add(new JScrollPane(table), BorderLayout.CENTER);

        // Listeners
        addButton.addActionListener(e -> handleAdd());
        updateButton.addActionListener(e -> handleUpdate());
        deleteButton.addActionListener(e -> handleDelete());
        table.getSelectionModel().addListSelectionListener(e -> populateFields());

        loadEmployees();
        setVisible(true);
    }

    private void handleAdd() {
        try (Connection conn = DBConnection.getConnection()) {
            EmployeeDAO dao = new EmployeeDAO(conn);
            Employee emp = new Employee(0, nameField.getText(), departmentField.getText(), designationField.getText(), Double.parseDouble(salaryField.getText()));
            dao.addEmployee(emp);
            JOptionPane.showMessageDialog(this, "Employee added successfully!", "Success", JOptionPane.INFORMATION_MESSAGE);
            loadEmployees();
            clearFields();
        } catch (Exception ex) {
            showError(ex);
        }
```

```
    }

    private void handleUpdate() {
        if (selectedId == -1) return;
        try (Connection conn = DBConnection.getConnection()) {
            EmployeeDAO dao = new EmployeeDAO(conn);
            Employee emp = new Employee(selectedId, nameField.getText(),
departmentField.getText(), designationField.getText(),
Double.parseDouble(salaryField.getText()));
            dao.updateEmployee(emp);
            JOptionPane.showMessageDialog(this, "Employee updated successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
            loadEmployees();
            clearFields();
        } catch (Exception ex) {
        showError(ex);
    }
}

private void handleDelete() {
    if (selectedId == -1) return;
    int confirm = JOptionPane.showConfirmDialog(this, "Are you sure you want to delete this
employee?", "Confirm Delete", JOptionPane.YES_NO_OPTION);
    if (confirm != JOptionPane.YES_OPTION) return;

    try (Connection conn = DBConnection.getConnection()) {
        EmployeeDAO dao = new EmployeeDAO(conn);
        dao.deleteEmployee(selectedId);
```

```
        JOptionPane.showMessageDialog(this, "Employee deleted successfully!", "Deleted",
JOptionPane.INFORMATION_MESSAGE);
        loadEmployees();
        clearFields();
    } catch (Exception ex) {
        showError(ex);
    }
}

private void loadEmployees() {
    try (Connection conn = DBConnection.getConnection()) {
        EmployeeDAO dao = new EmployeeDAO(conn);
        List<Employee> employees = dao.getAllEmployees();
        model.setRowCount(0);
        for (Employee emp : employees) {
            model.addRow(new Object[]{
                    emp.getId(),
                    emp.getName(),
                    emp.getDepartment(),
                    emp.getDesignation(),
                    emp.getSalary(),
                    emp.getNetSalary()
            });
        }
    } catch (Exception ex) {
        showError(ex);
    }
}
```

```
private void populateFields() {
    int row = table.getSelectedRow();
    if (row == -1) return;

    selectedId = (int) model.getValueAt(row, 0);
    nameField.setText((String) model.getValueAt(row, 1));
    departmentField.setText((String) model.getValueAt(row, 2));
    designationField.setText((String) model.getValueAt(row, 3));
    salaryField.setText(String.valueOf(model.getValueAt(row, 4)));
}

private void clearFields() {
    nameField.setText("");
    departmentField.setText("");
    designationField.setText("");
    salaryField.setText("");
    selectedId = -1;
}

private void showError(Exception ex) {
    JOptionPane.showMessageDialog(this, "Error: " + ex.getMessage(), "Error",
JOptionPane.ERROR_MESSAGE);
}
}

package ui;
```

```
import dao.EmployeeDAO;
import db.DBConnection;

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.Connection;

public class LoginForm extends JFrame {
    private JTextField usernameField;
    private JPasswordField passwordField;
    private JButton loginButton;

    public LoginForm() {
        setTitle("Login - Employee Payroll System");
        setSize(350, 200);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        setLayout(new GridLayout(3, 2, 10, 10));

        // UI Components
        add(new JLabel("Username:"));
        usernameField = new JTextField();
        add(usernameField);

        add(new JLabel("Password:"));
        passwordField = new JPasswordField();
        add(passwordField);
```

```
        loginButton = new JButton("Login");
        add(loginButton);
        add(new JLabel()); // empty cell for spacing

        // Action Listener
        loginButton.addActionListener(e -> handleLogin());

        setVisible(true);
    }

    private void handleLogin() {
        String username = usernameField.getText().trim();
        String password = new String(passwordField.getPassword()).trim();

        if (username.isEmpty() || password.isEmpty()) {
            JOptionPane.showMessageDialog(this, "Please enter both username and password.",
"Missing Info", JOptionPane.WARNING_MESSAGE);
            return;
        }

        try (Connection conn = DBConnection.getConnection()) {
            EmployeeDAO dao = new EmployeeDAO(conn);
            boolean isValid = dao.validateLogin(username, password);

            if (isValid) {
                JOptionPane.showMessageDialog(this, "Login successful!", "Welcome",
JOptionPane.INFORMATION_MESSAGE);
                new EmployeeForm(); // launch main UI
```

```
            dispose(); // close login window
        } else {
            JOptionPane.showMessageDialog(this, "Invalid credentials. Please try again.",
"Login Failed", JOptionPane.ERROR_MESSAGE);
        }
      } catch (Exception ex) {
        JOptionPane.showMessageDialog(this, "Error: " + ex.getMessage(), "Database Error",
JOptionPane.ERROR_MESSAGE);
      }
   }
}
```