1. ***Set Up MySQL Database***

USE employee\_db;

CREATE TABLE employees (

id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(100),

position VARCHAR(100),

salary DECIMAL(10, 2)

);

***Insert some sample data***

INSERT INTO employees (name, position, salary) VALUES (&#39;John Doe&#39;, &#39;Software

Engineer&#39;, 75000);

INSERT INTO employees (name, position, salary) VALUES (&#39;Jane Smith&#39;, &#39;HR

Manager&#39;, 65000);

INSERT INTO employees (name, position, salary) VALUES (&#39;Steve Brown&#39;, &#39;Team

Lead&#39;, 85000);

1. ***Establish JDBC Connection***

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class DatabaseConnection {

private static final String URL =

&quot;jdbc:mysql://localhost:3306/employee\_db&quot;; // Database URL

private static final String USER = &quot;root&quot;; // Your MySQL username

private static final String PASSWORD = &quot;password&quot;; // Your MySQL password

public static Connection getConnection() throws SQLException {

try {

// Load the JDBC driver

Class.forName(&quot;com.mysql.cj.jdbc.Driver&quot;);

// Return the database connection

return DriverManager.getConnection(URL, USER, PASSWORD);

} catch (ClassNotFoundException | SQLException e) {

System.out.println(&quot;Connection failed: &quot; + e.getMessage());

throw new SQLException(&quot;Failed to establish connection.&quot;);

}

}

}

1. ***Perform CRUD Operations***

Code for EmployeeDAO.java:

import java.sql.\*;

import java.util.ArrayList;

import java.util.List;

public class EmployeeDAO {

***// Create an employee***

public static void addEmployee(String name, String position, double

salary) {

String sql = &quot;INSERT INTO employees (name, position, salary) VALUES

(?, ?, ?)&quot;;

try (Connection conn = DatabaseConnection.getConnection();

PreparedStatement stmt = conn.prepareStatement(sql)) {

stmt.setString(1, name);

stmt.setString(2, position);

stmt.setDouble(3, salary);

int rowsAffected = stmt.executeUpdate();

System.out.println(&quot;Employee added successfully. Rows affected: &quot;

+ rowsAffected);

} catch (SQLException e) {

e.printStackTrace();

}

}

***// Read all employees***

public static List&lt;Employee&gt; getAllEmployees() {

List&lt;Employee&gt; employees = new ArrayList&lt;&gt;();

String sql = &quot;SELECT \* FROM employees&quot;;

try (Connection conn = DatabaseConnection.getConnection();

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery(sql)) {

while (rs.next()) {

Employee employee = new Employee(

rs.getInt(&quot;id&quot;),

rs.getString(&quot;name&quot;),

rs.getString(&quot;position&quot;),

rs.getDouble(&quot;salary&quot;)

);

employees.add(employee);

}

} catch (SQLException e) {

e.printStackTrace();

}

return employees;

}

***// Update an employee information***

public static void updateEmployee(int id, String name, String position,

double salary) {

String sql = &quot;UPDATE employees SET name = ?, position = ?, salary = ?

WHERE id = ?&quot;;

try (Connection conn = DatabaseConnection.getConnection();

PreparedStatement stmt = conn.prepareStatement(sql)) {

stmt.setString(1, name);

stmt.setString(2, position);

stmt.setDouble(3, salary);

stmt.setInt(4, id);

int rowsAffected = stmt.executeUpdate();

System.out.println(&quot;Employee updated successfully. Rows affected:

&quot; + rowsAffected);

} catch (SQLException e) {

e.printStackTrace();

}

}

***// Delete an employee***

public static void deleteEmployee(int id) {

String sql = &quot;DELETE FROM employees WHERE id = ?&quot;;

try (Connection conn = DatabaseConnection.getConnection();

PreparedStatement stmt = conn.prepareStatement(sql)) {

stmt.setInt(1, id);

int rowsAffected = stmt.executeUpdate();

System.out.println(&quot;Employee deleted successfully. Rows affected:

&quot; + rowsAffected);

} catch (SQLException e) {

e.printStackTrace();

}

}

}

1. ***Create Employee.java Class***

public class Employee {

private int id;

private String name;

private String position;

private double salary;

public Employee(int id, String name, String position, double salary) {

this.id = id;

this.name = name;

this.position = position;

this.salary = salary;

}

// Getters and setters

public int getId() { return id; }

public void setId(int id) { this.id = id; }

public String getName() { return name; }

public void setName(String name) { this.name = name; }

public String getPosition() { return position; }

public void setPosition(String position) { this.position = position; }

public double getSalary() { return salary; }

public void setSalary(double salary) { this.salary = salary; }

@Override

public String toString() {

return &quot;Employee{id=&quot; + id + &quot;, name=&#39;&quot; + name + &quot;&#39;, position=&#39;&quot; +

position + &quot;&#39;, salary=&quot; + salary + &#39;}&#39;;

}

}

1. ***Test the Application***

import java.util.List;

public class Main {

public static void main(String[] args) {

***// Add employees***

EmployeeDAO.addEmployee(&quot;Alice Cooper&quot;, &quot;Developer&quot;, 70000);

EmployeeDAO.addEmployee(&quot;Bob Marley&quot;, &quot;Manager&quot;, 80000);

***// Update employee***

EmployeeDAO.updateEmployee(1, &quot;John Doe&quot;, &quot;Senior Software Engineer&quot;,

90000);

***// Get all employees***

List&lt;Employee&gt; employees = EmployeeDAO.getAllEmployees();

employees.forEach(System.out::println);

// Delete employee

EmployeeDAO.deleteEmployee(2);

}

