**Employee Database Management**

**Employee Model**

package model;

public class Employee {

private int id;

private String name;

private double salary;

// Constructors

public Employee() {}

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

this.id = id;

this.name = name;

this.salary = salary;

}

// Getters & 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 double getSalary() { return salary; }

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

@Override

public String toString() {

return "Employee [id=" + id + ", name=" + name + ", salary=" + salary + "]";

}

}

**EmployeeDAO**

package dao;

import model.Employee;

import java.util.List;

public interface EmployeeDAO {

void save(Employee employee);

Employee getById(int id);

List<Employee> getAll();

void update(Employee employee);

void delete(int id);

}

EmployeeDAOImpl

package dao;

import model.Employee;

import org.springframework.jdbc.core.JdbcTemplate;

import org.springframework.jdbc.core.RowMapper;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.List;

public class EmployeeDAOImpl implements EmployeeDAO {

private JdbcTemplate jdbcTemplate;

public void setJdbcTemplate(JdbcTemplate jdbcTemplate) {

this.jdbcTemplate = jdbcTemplate;

}

private RowMapper<Employee> rowMapper = new RowMapper<>() {

public Employee mapRow(ResultSet rs, int rowNum) throws SQLException {

return new Employee(rs.getInt("id"), rs.getString("name"), rs.getDouble("salary"));

}

};

public void save(Employee emp) {

String sql = "INSERT INTO employee (id, name, salary) VALUES (?, ?, ?)";

jdbcTemplate.update(sql, emp.getId(), emp.getName(), emp.getSalary());

}

public Employee getById(int id) {

String sql = "SELECT \* FROM employee WHERE id=?";

return jdbcTemplate.queryForObject(sql, rowMapper, id);

}

public List<Employee> getAll() {

String sql = "SELECT \* FROM employee";

return jdbcTemplate.query(sql, rowMapper);

}

public void update(Employee emp) {

String sql = "UPDATE employee SET name=?, salary=? WHERE id=?";

jdbcTemplate.update(sql, emp.getName(), emp.getSalary(), emp.getId());

}

public void delete(int id) {

String sql = "DELETE FROM employee WHERE id=?";

jdbcTemplate.update(sql, id);

}

}

**Spring configuration**

package config;

import dao.EmployeeDAOImpl;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.jdbc.core.JdbcTemplate;

import org.springframework.jdbc.datasource.DriverManagerDataSource;

import javax.sql.DataSource;

@Configuration

public class AppConfig {

@Bean

public DataSource dataSource() {

DriverManagerDataSource ds = new DriverManagerDataSource();

ds.setDriverClassName("com.mysql.cj.jdbc.Driver");

ds.setUrl("jdbc:mysql://localhost:3306/employee\_db");

ds.setUsername("root");

ds.setPassword("your\_password");

return ds;

}

@Bean

public JdbcTemplate jdbcTemplate() {

return new JdbcTemplate(dataSource());

}

@Bean

public EmployeeDAOImpl employeeDAO() {

EmployeeDAOImpl dao = new EmployeeDAOImpl();

dao.setJdbcTemplate(jdbcTemplate());

return dao;

}

}

**Main APP**

import config.AppConfig;

import dao.EmployeeDAO;

import model.Employee;

import org.springframework.context.ApplicationContext;

import org.springframework.context.annotation.AnnotationConfigApplicationContext;

import java.util.Scanner;

public class MainApp {

public static void main(String[] args) {

ApplicationContext context = new AnnotationConfigApplicationContext(AppConfig.class);

EmployeeDAO dao = context.getBean(EmployeeDAO.class);

Scanner scanner = new Scanner(System.in);

while (true) {

System.out.println("\n1. Add\n2. View All\n3. View by ID\n4. Update\n5. Delete\n6. Exit");

int choice = scanner.nextInt();

switch (choice) {

case 1 -> {

System.out.print("Enter ID, Name, Salary: ");

dao.save(new Employee(scanner.nextInt(), scanner.next(), scanner.nextDouble()));

}

case 2 -> dao.getAll().forEach(System.out::println);

case 3 -> {

System.out.print("Enter ID: ");

System.out.println(dao.getById(scanner.nextInt()));

}

case 4 -> {

System.out.print("Enter ID to Update: ");

int id = scanner.nextInt();

System.out.print("Enter New Name and Salary: ");

dao.update(new Employee(id, scanner.next(), scanner.nextDouble()));

}

case 5 -> {

System.out.print("Enter ID to Delete: ");

dao.delete(scanner.nextInt());

}

case 6 -> {

System.out.println("Exiting...");

scanner.close();

return;

}

default -> System.out.println("Invalid choice!");

}

}

}

}

**MYSQL Setup**

CREATE DATABASE employee\_db;

USE employee\_db;

CREATE TABLE employee (

id INT PRIMARY KEY,

name VARCHAR(100),

salary DOUBLE

);

**Phaninder Gunda**