Employee Management System Setup Guide

Required Softwares:

1. JDK 17 or higher
2. Eclipse IDE for Enterprise Java
3. Apache Tomcat 10.1.36
4. MySQL Server - Download
5. MySQL Workbench - Download

Install Apache Tomcat V10.1

STEP 1: Configure Tomcat in Eclipse

1. Open Eclipse IDE
2. Go to Window → Preferences
3. Navigate to Server → Runtime Environments
4. Click Add → Select Apache Tomcat v10.1
5. Click **Next** → Browse to your Tomcat folder: C:\path\to\apache-tomcat-10.1.36
6. Click **Finish** and Apply and Close

STEP 2: Create Database

1. Open MySQL Workbench
2. Connect to your MySQL Server open new query
3. Copy and execute the entire SQL script from employee_management.sql and press enter

employee_management.sql

-- Employee Management System Database Schema

(goto workbench. Then new query. copy past and run, check if there any DB created or not after run this, refresh and check)

```
CREATE DATABASE IF NOT EXISTS employee_management;

USE employee_management;


-- Table 1: Admin/User table for login

CREATE TABLE IF NOT EXISTS admin (

    admin_id INT PRIMARY KEY AUTO_INCREMENT,

    username VARCHAR(50) UNIQUE NOT NULL,

    password VARCHAR(255) NOT NULL,

    email VARCHAR(100) NOT NULL,

    full_name VARCHAR(100) NOT NULL,

    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP

);


-- Table 2: Department table

CREATE TABLE IF NOT EXISTS department (

    dept_id INT PRIMARY KEY AUTO_INCREMENT,

    dept_name VARCHAR(100) NOT NULL,

    dept_code VARCHAR(20) UNIQUE NOT NULL,

    location VARCHAR(100),

    manager_name VARCHAR(100),

    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP

);
```

```
-- Table 3: Employee table
CREATE TABLE IF NOT EXISTS employee (
    emp_id INT PRIMARY KEY AUTO_INCREMENT,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    phone VARCHAR(20),
    dept_id INT,
    position VARCHAR(100),
    salary DECIMAL(10, 2),
    hire_date DATE,
    status VARCHAR(20) DEFAULT 'Active',
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (dept_id) REFERENCES department(dept_id) ON DELETE SET NULL
);

-- Table 4: Attendance table (for additional functionality)
CREATE TABLE IF NOT EXISTS attendance (
    attendance_id INT PRIMARY KEY AUTO_INCREMENT,
    emp_id INT NOT NULL,
    attendance_date DATE NOT NULL,
    check_in_time TIME,
    check_out_time TIME,
    status VARCHAR(20) DEFAULT 'Present',
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (emp_id) REFERENCES employee(emp_id) ON DELETE CASCADE
);
```

```
-- Insert sample admin user (password: admin123)
INSERT INTO admin (username, password, email, full_name) VALUES
('admin', 'admin123', 'admin@employee.com', 'System Administrator');

-- Insert sample departments
INSERT INTO department (dept_name, dept_code, location, manager_name) VALUES
('Information Technology', 'IT', 'Building A - Floor 3', 'John Smith'),
('Human Resources', 'HR', 'Building B - Floor 1', 'Sarah Johnson'),
('Finance', 'FIN', 'Building A - Floor 2', 'Michael Brown'),
('Marketing', 'MKT', 'Building C - Floor 1', 'Emily Davis'),
('Operations', 'OPS', 'Building B - Floor 2', 'David Wilson');

-- Insert sample employees
INSERT INTO employee (first_name, last_name, email, phone, dept_id, position, salary,
hire_date, status) VALUES
('Alice', 'Johnson', 'alice.j@company.com', '+94771234567', 1, 'Senior Developer', 85000.00,
'2022-01-15', 'Active'),
('Bob', 'Williams', 'bob.w@company.com', '+94772345678', 1, 'Junior Developer', 55000.00,
'2023-03-20', 'Active'),
('Carol', 'Martinez', 'carol.m@company.com', '+94773456789', 2, 'HR Manager', 75000.00,
'2021-06-10', 'Active'),
('David', 'Garcia', 'david.g@company.com', '+94774567890', 3, 'Accountant', 65000.00, '2022-09-
05', 'Active'),
('Emma', 'Rodriguez', 'emma.r@company.com', '+94775678901', 4, 'Marketing Specialist',
60000.00, '2023-01-12', 'Active'),
('Frank', 'Lee', 'frank.l@company.com', '+94776789012', 5, 'Operations Manager', 80000.00,
'2020-11-22', 'Active');
```

```
-- Insert sample attendance records
INSERT INTO attendance (emp_id, attendance_date, check_in_time, check_out_time, status)
VALUES
(1, '2024-10-14', '08:30:00', '17:30:00', 'Present'),
(2, '2024-10-14', '09:00:00', '18:00:00', 'Present'),
(3, '2024-10-14', '08:45:00', '17:15:00', 'Present'),
(4, '2024-10-14', '08:30:00', '17:30:00', 'Present'),
(5, '2024-10-14', '09:15:00', '17:45:00', 'Present'),
(1, '2024-10-15', '08:35:00', '17:25:00', 'Present'),
(2, '2024-10-15', '09:05:00', '18:10:00', 'Present');

-- Display all data
SELECT 'Admin Users:' as '';
SELECT * FROM admin;

SELECT 'Departments:' as '';
SELECT * FROM department;

SELECT 'Employees:' as '';
SELECT * FROM employee;

SELECT 'Attendance Records:' as '';
SELECT * FROM attendance;
```

STEP 4: Add MySQL Connector

1. Download MySQL Connector/J: Download Here or mn ekath zip eka drive ekat dnnm
2. Extract the ZIP file
3. Copy to project: src/main/webapp/WEB-INF/lib/
4. Right-click project → **Build Path** → **Configure Build Path**
5. **Libraries** tab → **Add JARs** → Select the MySQL connector → **Apply and Close**

STEP 4: Change SQL credentials

Open DBConnectionUtil.java and update here password with your Workbench password.
private static final String PASSWORD = "Mysql908";

STEP 5: RUN project

1. Right-click on project → **Run As** → **Run on Server**
2. Select **Tomcat v10.1** → **Finish**
3. Browser will open automatically at: http://localhost:8080/EmployeeManagementSystem/

User login

Username: admin
Password: admin123

What did I used in here?

MVC Architecture:

- Model: Java classes (Employee, Department, Admin)
- View: JSP pages
- Controller: Servlets

OOP Concepts Used:

- **Encapsulation**: Private fields with getters/setters
- **Inheritance**: Can extend from base classes
- **Abstraction**: DAO pattern abstracts database operations
- **Polymorphism**: Method overriding in servlets (doGet, doPost)

Design Patterns:

- **Singleton**: Database connection
- **DAO Pattern**: Separate data access logic
- **MVC Pattern**: Separation of concerns

CRUD Operations:

- **Create**: INSERT queries in DAO
- **Read**: SELECT queries
- **Update**: UPDATE queries
- **Delete**: DELETE queries

Security Features:

- Session management
- Login authentication
- SQL prepared statements (prevents SQL injection)