

Advanced Java [Day – 3]

UID: 24MCI10204

Name: Rahul Saxena

Branch: 24MCA – AI & ML

Question 1: Employee Directory servlet Objective Create a java servlet that connects to a MySQL or any JDBC Supported database to display all employee and enable searching by there id

Code:

DbConnection.java

```
package EmployeeManagement;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
public class DBConnection {
    public static Connection getConnection() throws SQLException, ClassNotFoundException {
        String url = "jdbc:mysql://localhost:3306/Employeeedb";
        String user = "root";
        String password = "12345";
        Class.forName("com.mysql.cj.jdbc.Driver");
        return DriverManager.getConnection(url, user, password);
    }
}
```

EmployeeServlet.java

```
package EmployeeManagement;

import jakarta.servlet.http.HttpServlet;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.*;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.*;

public class EmployeeServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        String searchId = request.getParameter("id");

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<h2>Employee Directory</h2>");
        out.println("<form method='get'>Search by ID: <input type='text' name='id'> <input type='submit'>
value='Search'></form>");
        out.println("<table
border='1'><tr><th>ID</th><th>Name</th><th>Email</th><th>Department</th></tr>");

        try (Connection con = DBConnection.getConnection()) {
            String query = "SELECT * FROM employees";
```

```

        if (searchId != null && !searchId.trim().isEmpty()) {
            query += " WHERE id = ?";
        }

        PreparedStatement stmt = con.prepareStatement(query);

        if (searchId != null && !searchId.trim().isEmpty()) {
            stmt.setInt(1, Integer.parseInt(searchId));
        }

        ResultSet rs = stmt.executeQuery();
        while (rs.next()) {
            out.println("<tr><td>" + rs.getInt("id") + "</td><td>" +
                rs.getString("name") + "</td><td>" +
                rs.getString("email") + "</td><td>" +
                rs.getString("department") + "</td></tr>");
        }

    } catch (Exception e) {
        e.printStackTrace(out);
    }
    out.println("</table>");
}
}

```

Question 2: Employee Onboarding in an HR System

Problem Statement:

In a Human Resource (HR) management system, when a new employee joins the company, the system needs to perform several tasks:

1. **Initialize Employee Data:** Set up the employee's profile with default values.
2. **Connect to External Services:** Establish connections to external systems like email services, payroll systems, and authentication services.
3. **Send Welcome Email:** Send a welcome email to the new employee.
4. **Clean Up Resources:** Release resources like database connections and close external service connections when the employee's profile is no longer needed

Code:

```

package EmployeeManagement;

class Employee {
    int id;
    String name;
    String email;
    String department;
    public Employee(int id, String name, String email, String department) {

```

```

        this.id = id;
        this.name = name;
        this.email = email;
        this.department = department;
    }
}

class EmailService {
    void connect() {
        System.out.println("Connected to Email Service.");
    }
    void sendWelcomeEmail(String email, String name) {
        System.out.println("Sending welcome email to " + name + " at " + email);
    }
    void disconnect() {
        System.out.println("Disconnected from Email Service.");
    }
}

class PayrollService {
    void connect() {
        System.out.println("Connected to Payroll System.");
    }
    void registerEmployee(Employee emp) {
        System.out.println("Registered " + emp.name + " in Payroll.");
    }
    void disconnect() {
        System.out.println("Disconnected from Payroll System.");
    }
}

class AuthService {
    void connect() {
        System.out.println("Connected to Authentication Service.");
    }
    void createCredentials(Employee emp) {
        System.out.println("Created login credentials for " + emp.name);
    }
    void disconnect() {
        System.out.println("Disconnected from Authentication Service.");
    }
}

public class EmployeeOnboarding {
    public static void onboardNewEmployee(Employee emp) {
        System.out.println("Starting onboarding for: " + emp.name);
        EmailService emailService = new EmailService();
        PayrollService payrollService = new PayrollService();
    }
}

```

```

    AuthService authService = new AuthService();
    emailService.connect();
    payrollService.connect();
    authService.connect();
    emailService.sendWelcomeEmail(emp.email, emp.name);
    payrollService.registerEmployee(emp);
    authService.createCredentials(emp);
    emailService.disconnect();
    payrollService.disconnect();
    authService.disconnect();
    System.out.println("Onboarding completed for: " + emp.name);
}
public static void main(String[] args)
    Employee newEmp = new Employee(1001, "Rahul Saxena", "rahul@example.com", "IT");
    onboardNewEmployee(newEmp);
}
}

```

Onboarding Servlet:

```

package EmployeeManagement;

import java.io.*;
import jakarta.servlet.*;
import jakarta.servlet.http.*;

public class OnboardingServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        Employee emp = new Employee(101, "Rahul Saxena", "rahul@example.com", "IT");

        out.println("<h2>Starting Employee Onboarding</h2>");
        out.println("<pre>");

        EmailService emailService = new EmailService();
        PayrollService payrollService = new PayrollService();
        AuthService authService = new AuthService();

        out.println("-> Connecting to Email Service...");
        emailService.connect();
        out.println("-> Sending welcome email...");
        emailService.sendWelcomeEmail(emp.email, emp.name);
    }
}

```

```
out.println("-> Connecting to Payroll System...");
payrollService.connect();
out.println("-> Registering employee in Payroll...");
payrollService.registerEmployee(emp);
```

```
out.println("-> Connecting to Authentication Service...");
authService.connect();
out.println("-> Creating login credentials...");
authService.createCredentials(emp);
```

```
out.println("-> Disconnecting from all services...");
emailService.disconnect();
payrollService.disconnect();
authService.disconnect();
```

```
out.println("\nOnboarding completed for: " + emp.name);
out.println("</pre>");
```

```
}
```

```
}
```