

## CODE:

### Employee Dao.java

```
package com.hcl.dao;

import org.apache.log4j.Logger;
import org.hibernate.Session;
import org.hibernate.Transaction;

import com.hcl.model.Employee;
import com.hcl.util.HibernateUtil;

public class EmployeeDao {

    static Logger log = Logger.getLogger(EmployeeDao.class.getName());

    public boolean saveUser(Employee employee) {

        Transaction transaction = null;

        try (Session session = HibernateUtil.getSessionFactory().openSession()) {

            transaction = session.beginTransaction();

            session.save(employee);

            //hibernate insert query to save user details in DB

            transaction.commit();

            log.info("Data is saved");

            return true;

        } catch (Exception e) {

            e.printStackTrace();

            return false;

        }

    }

}
```

```

public String validate(String username, String password) {
    Transaction transaction = null;
    Employee user = null;
    try (Session session = HibernateUtil.getSessionFactory().openSession()) {
        transaction = session.beginTransaction();
        //To find if employee exists in DB
        user = findIfEmployeeExists(username);
        //If user exist, go to login page
        if (user != null) {
            if (user.getPassword().equals(password)) {
                log.info("Pass");
                return "Pass";
            } else {
                //If user exist but gave wrong password, ask again to give
correct password

                log.info("Exists");
                return "Exists";
            }
        } else {
            //If user doesn't exist, ask to register
            log.info("NotExists");
            return "NotExists";
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    log.info("TryAgain");
    return "TryAgain";
}

```

//To find if employee exists in DB

```
    public Employee findIfEmployeeExists(String username) {  
        Transaction transaction = null;  
        Employee employee = null;  
        try (Session session = HibernateUtil.getSessionFactory().openSession()) {  
            transaction = session.beginTransaction();  
            //hibernate select query to find if user exists in DB  
            employee = (Employee) session.createQuery("FROM Employee U  
WHERE U.username = :username")  
                .setParameter("username", username).uniqueResult();  
            log.info("Finding if data exists");  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
        return employee;  
    }  
}
```

### **Employee.java**

```
package com.hcl.model;
```

```
import java.io.Serializable;
```

```
import javax.persistence.Column;
```

```
import javax.persistence.Entity;
```

```
import javax.persistence.GeneratedValue;
```

```
import javax.persistence.GenerationType;
```

```
import javax.persistence.Id;
```

```
import javax.persistence.Table;
```

```
@Entity
@Table(name = "employees")
public class Employee implements Serializable {
    private static final long serialVersionUID = 1L;

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;

    @Column(name = "first_name")
    private String firstName;

    @Column(name = "last_name")
    private String lastName;

    @Column(name = "user_name")
    private String username;

    @Column(name = "password")
    private String password;

    @Column(name = "email")
    private String email;

    @Column(name = "contactNumber")
    private String contactNumber;

    /**
```

```
* @return the email
*/
public String getEmail() {
    return email;
}

/**
 * @param email the email to set
 */
public void setEmail(String email) {
    this.email = email;
}

/**
 * @return the contactNumber
 */
public String getContactNumber() {
    return contactNumber;
}

/**
 * @param contactNumber the contactNumber to set
 */
public void setContactNumber(String contactNumber) {
    this.contactNumber = contactNumber;
}

public String getFirstName() {
```

```
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }

    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {
        this.username = username;
    }

    public String getPassword() {
        return password;
    }

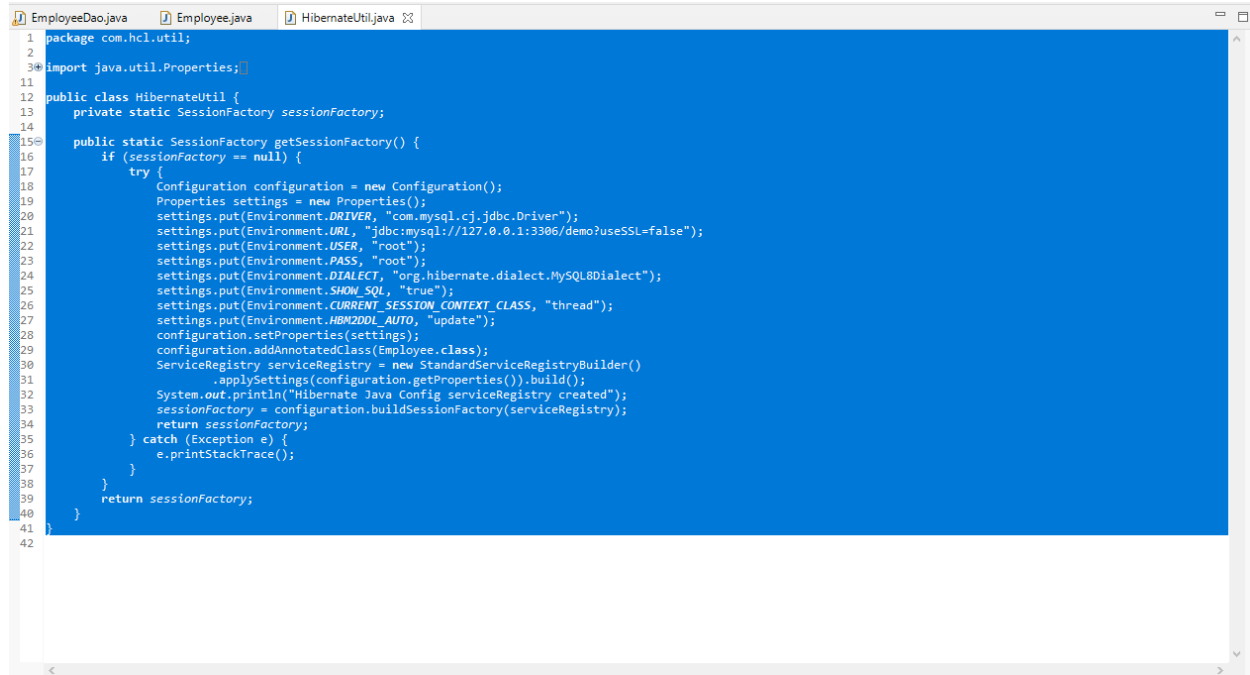
    public void setPassword(String password) {
```

```

        this.password = password;
    }
}

```

## HibernateUtil.java



```

1 package com.hcl.util;
2
3 import java.util.Properties;
4
11 public class HibernateUtil {
12     private static SessionFactory sessionFactory;
13
14     public static SessionFactory getSessionFactory() {
15         if (sessionFactory == null) {
16             try {
17                 Configuration configuration = new Configuration();
18                 Properties settings = new Properties();
19                 settings.put(Environment.DRIVER, "com.mysql.cj.jdbc.Driver");
20                 settings.put(Environment.URL, "jdbc:mysql://127.0.0.1:3306/demo?useSSL=false");
21                 settings.put(Environment.USER, "root");
22                 settings.put(Environment.PASS, "root");
23                 settings.put(Environment.DIALECT, "org.hibernate.dialect.MySQL8Dialect");
24                 settings.put(Environment.SHOW_SQL, "true");
25                 settings.put(Environment.CURRENT_SESSION_CONTEXT_CLASS, "thread");
26                 settings.put(Environment.HBM2DDL_AUTO, "update");
27                 configuration.setProperties(settings);
28                 configuration.addAnnotatedClass(Employee.class);
29                 ServiceRegistry serviceRegistry = new StandardServiceRegistryBuilder()
30                     .applySettings(configuration.getProperties()).build();
31                 System.out.println("Hibernate Java Config serviceRegistry created");
32                 sessionFactory = configuration.buildSessionFactory(serviceRegistry);
33                 return sessionFactory;
34             } catch (Exception e) {
35                 e.printStackTrace();
36             }
37         }
38         return sessionFactory;
39     }
40 }
41
42

```

## EmployeeLoginServlet.java

```

package com.hcl.web;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

```

```
import org.apache.log4j.Logger;
```

```
import com.hcl.dao.EmployeeDao;
```

```
@WebServlet("/login")
```

```
public class EmployeeLoginServlet extends HttpServlet {
```

```
    static Logger log = Logger.getLogger(EmployeeDao.class.getName());
```

```
    private static final long serialVersionUID = 1L;
```

```
    private EmployeeDao loginDao;
```

```
    public EmployeeLoginServlet() {
```

```
        super();
```

```
        loginDao = new EmployeeDao();
```

```
    }
```

```
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
```

```
        throws ServletException, IOException {
```

```
        String message = null;
```

```
        try {
```

```
            authenticate(request, response);
```

```
        } catch (Exception e) {
```

```
            //If unexpected error occurred, ask to login again
```

```
            RequestDispatcher dispatcher =
```

```
request.getRequestDispatcher("login.jsp");
```

```
            message = "Unexpected Error Occured! Try again!";
```

```
            //sending message to user
```

```
            log.info("Unexpected Error Occured! Try again!");
```

```
            request.setAttribute("message", message);
```

```
            dispatcher.forward(request, response);
```



```
    }  
}
```

```
private void authenticate(HttpServletRequest request, HttpServletResponse response)  
throws Exception {
```

```
    String username = request.getParameter("username");  
    String password = request.getParameter("password");  
    String message = null;  
    String check = loginDao.validate(username, password);  
    if (check.equals("Pass")) {  
        //If user exist, Successfull login  
        System.out.println("LoginSuccess");  
        log.info("LoginSuccess");  
        RequestDispatcher dispatcher =  
request.getRequestDispatcher("SuccessLogin.jsp");  
        dispatcher.forward(request, response);  
    } else if (check.equals("Exists")) {  
        //If user exist but gave wrong password, ask again to give correct  
password  
        System.out.println("Exists");  
        log.info("Exists");  
        RequestDispatcher dispatcher =  
request.getRequestDispatcher("login.jsp");  
        message = "Wrong Password! Try Again!";  
        //sending message to user  
        request.setAttribute("message", message);  
        dispatcher.forward(request, response);  
    } else if (check.equals("NotExists")) {  
        //If user doesn't exist, ask to register  
        System.out.println("NotExists");
```

```

        log.info("NotExists");

        RequestDispatcher dispatcher =
request.getRequestDispatcher("register.jsp");

        message = "Are you a New Employee? Please Register!";
        //sending message to user
        request.setAttribute("message", message);
        dispatcher.forward(request, response);
    } else {
        //If unexpected error occurred, ask to login again
        System.out.println("Unexpected");
        log.info("Unexpected Error Occured! Try again!");

        RequestDispatcher dispatcher =
request.getRequestDispatcher("login.jsp");

        message = "Unexpected Error Occured! Try again!";
        //sending message to user
        request.setAttribute("message", message);
        dispatcher.forward(request, response);
    }
}
}
}

```

### **EmployeeRegistrationServlet.java**

```

package com.hcl.web;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;

```

```

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.log4j.Logger;

import com.hcl.dao.EmployeeDao;
import com.hcl.model.Employee;

@WebServlet("/register")
public class EmployeeRegistrationServlet extends HttpServlet {
    static Logger log = Logger.getLogger(EmployeeDao.class.getName());
    private static final long serialVersionUID = 1L;
    private EmployeeDao employeeDao;

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        register(request, response);
    }

    private void register(HttpServletRequest request, HttpServletResponse response)
        throws IOException, ServletException {
        String message = null;
        Employee ifEmployeeExists =
employeeDao.findIfEmployeeExists(request.getParameter("username"));
        if (ifEmployeeExists == null) {
            System.out.println("Saved");
            //get all values entered by the user
            String firstName = request.getParameter("firstName");
            String lastName = request.getParameter("lastName");

```

```
String username = request.getParameter("username");
String password = request.getParameter("password");
String email = request.getParameter("email");
String contactNumber = request.getParameter("contactNumber");
Employee employee = new Employee();
//store the values in employee
employee.setFirstName(firstName);
employee.setLastName(lastName);
employee.setUsername(username);
employee.setPassword(password);
employee.setEmail(email);
employee.setContactNumber(contactNumber);
if (employeeDao.saveUser(employee)) {
    RequestDispatcher dispatcher =
request.getRequestDispatcher("login.jsp");
    dispatcher.forward(request, response);
} else {
    RequestDispatcher dispatcher =
request.getRequestDispatcher("register.jsp");
    message = "Enter correct values in the fields!";
    //sending message to user
    request.setAttribute("message", message);
    log.info(message);
    dispatcher.forward(request, response);
}
} else {
    System.out.println("Same User");
    RequestDispatcher dispatcher =
request.getRequestDispatcher("register.jsp");
```

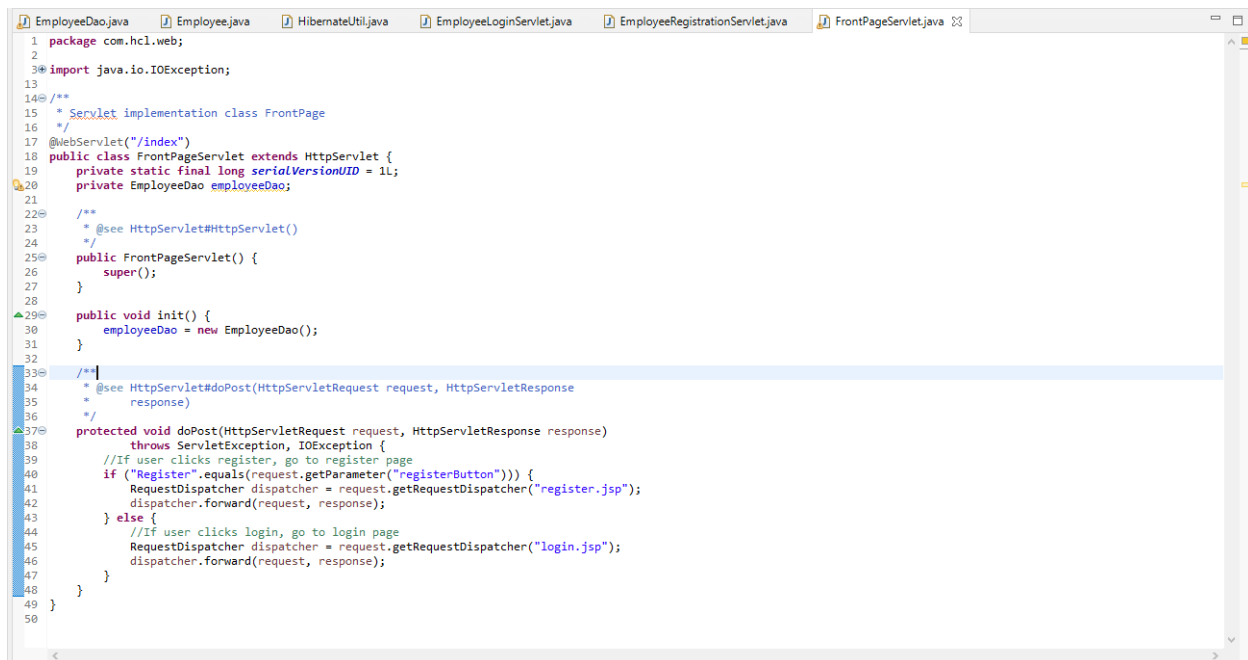
```

        //sending message to user
        message = "Please use different Username!";
        request.setAttribute("message", message);
        log.info(message);
        dispatcher.forward(request, response);
    }
}

public EmployeeRegistrationServlet() {
    super();
    employeeDao = new EmployeeDao();
}
}

```

## FrontPageServlet.java



```

1 package com.hcl.web;
2
3 import java.io.IOException;
4
5 /**
6  * Servlet implementation class FrontPage
7  */
8 @WebServlet("/index")
9 public class FrontPageServlet extends HttpServlet {
10     private static final long serialVersionUID = 1L;
11     private EmployeeDao employeeDao;
12
13     /**
14      * @see HttpServlet#HttpServlet()
15      */
16     public FrontPageServlet() {
17         super();
18     }
19
20     public void init() {
21         employeeDao = new EmployeeDao();
22     }
23
24     /**
25      * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
26      */
27     protected void doPost(HttpServletRequest request, HttpServletResponse response)
28         throws ServletException, IOException {
29         //If user clicks register, go to register page
30         if ("Register".equals(request.getParameter("registerButton"))) {
31             RequestDispatcher dispatcher = request.getRequestDispatcher("register.jsp");
32             dispatcher.forward(request, response);
33         } else {
34             //If user clicks login, go to login page
35             RequestDispatcher dispatcher = request.getRequestDispatcher("login.jsp");
36             dispatcher.forward(request, response);
37         }
38     }
39 }
40
41
42
43
44
45
46
47
48
49
50

```

## Log4j.properties

```
EmployeeDao.java Employee.java HibernateUtil.java EmployeeLoginServlet.java EmployeeRegistrationServlet.java FrontPageServlet.java log4j.properties
1 # Root logger option
2 log4j.rootLogger=DEBUG, stdout, file
3
4 # Redirect log messages to console
5 log4j.appender.stdout=org.apache.log4j.ConsoleAppender
6 log4j.appender.stdout.Target=System.out
7 log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
8 log4j.appender.stdout.layout.ConversionPattern=[%t] %-5p %c %x - %m%n
9
10 # Redirect log messages to a log file
11 log4j.appender.file=org.apache.log4j.RollingFileAppender
12 #outputs to Tomcat home
13 log4j.appender.file.File=${catalina.home}/logs/userLoginRegisterPhaseTwo.log
14 log4j.appender.file.MaxFileSize=5MB
15 log4j.appender.file.MaxBackupIndex=10
16 log4j.appender.file.layout=org.apache.log4j.PatternLayout
17 log4j.appender.file.layout.ConversionPattern=[%t] %-5p %c %x - %m%n
```

## Web.xml

```
EmployeeDao.java Employee.java HibernateUtil.java EmployeeLoginServlet.java EmployeeRegistrationServlet.java FrontPageServlet.java log4j.properties web.xml
1 <?DOCTYPE web-app PUBLIC
2 "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
3 "http://java.sun.com/dtd/web-app_2_3.dtd" >
4
5 <web-app>
6   <display-name>Archetype Created Web Application</display-name>
7   <servlet>
8     <servlet-name>FrontPageServlet</servlet-name>
9     <display-name>FrontPageServlet</display-name>
10    <description></description>
11    <servlet-class>com.hcl.web.FrontPageServlet</servlet-class>
12  </servlet>
13  <servlet-mapping>
14    <servlet-name>FrontPageServlet</servlet-name>
15    <url-pattern>/FrontPageServlet</url-pattern>
16  </servlet-mapping>
17 </web-app>
18
```

## Index.jsp

```
EmployeeDao.java Employee.java HibernateUtil.java EmployeeLoginServ... EmployeeRegistrati... FrontPageServetJa... log4j.properties web.xml index.jsp
1<html>
2<body>
3<h2>Welcome to Employee Association!</h2>
4<form action="<%request.getContextPath()%/index" method="post">
5<tr>
6<td>&nbsp;</td>
7<td><button type="submit" class="btn btn-primary"
8value="Register" name="registerButton">Register</button></td>
9</tr>
10<tr>
11<td>&nbsp;</td>
12<td><button type="submit" class="btn btn-primary" value="Member"
13name="memberButton">Already a Member?</button></td>
14</tr>
15</form>
16</body>
17</html>
18
```

## Login.jsp

```
EmployeeDao.j... Employee.java HibernateUtil... EmployeeLogin... EmployeeRegis... FrontPageServ... log4j.properti... web.xml index.jsp login.jsp
1<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2pageEncoding="ISO-8859-1"%>
3<!DOCTYPE html>
4<html>
5<head>
6<meta charset="ISO-8859-1">
7<title>Employee Login</title>
8</head>
9<body>
10<form action="<%request.getContextPath()%/Login" method="post">
11<%
12if (request.getAttribute("message") != null) {
13%>
14<h2><%request.getAttribute("message")%></h2>
15<%
16}
17%>
18<table>
19<tr>
20<td>User Name:</td>
21<td><input type="text" id="username" placeholder="User Name"
22name="username" required></td>
23</tr>
24<tr>
25<td>Password:</td>
26<td><input type="text" id="password" placeholder="Password"
27name="password" required></td>
28</tr>
29<tr>
30<td>&nbsp;</td>
31<td><button type="submit" class="btn btn-primary">Login</button></td>
32</tr>
33</table>
34</form>
35</body>
36</html>
```

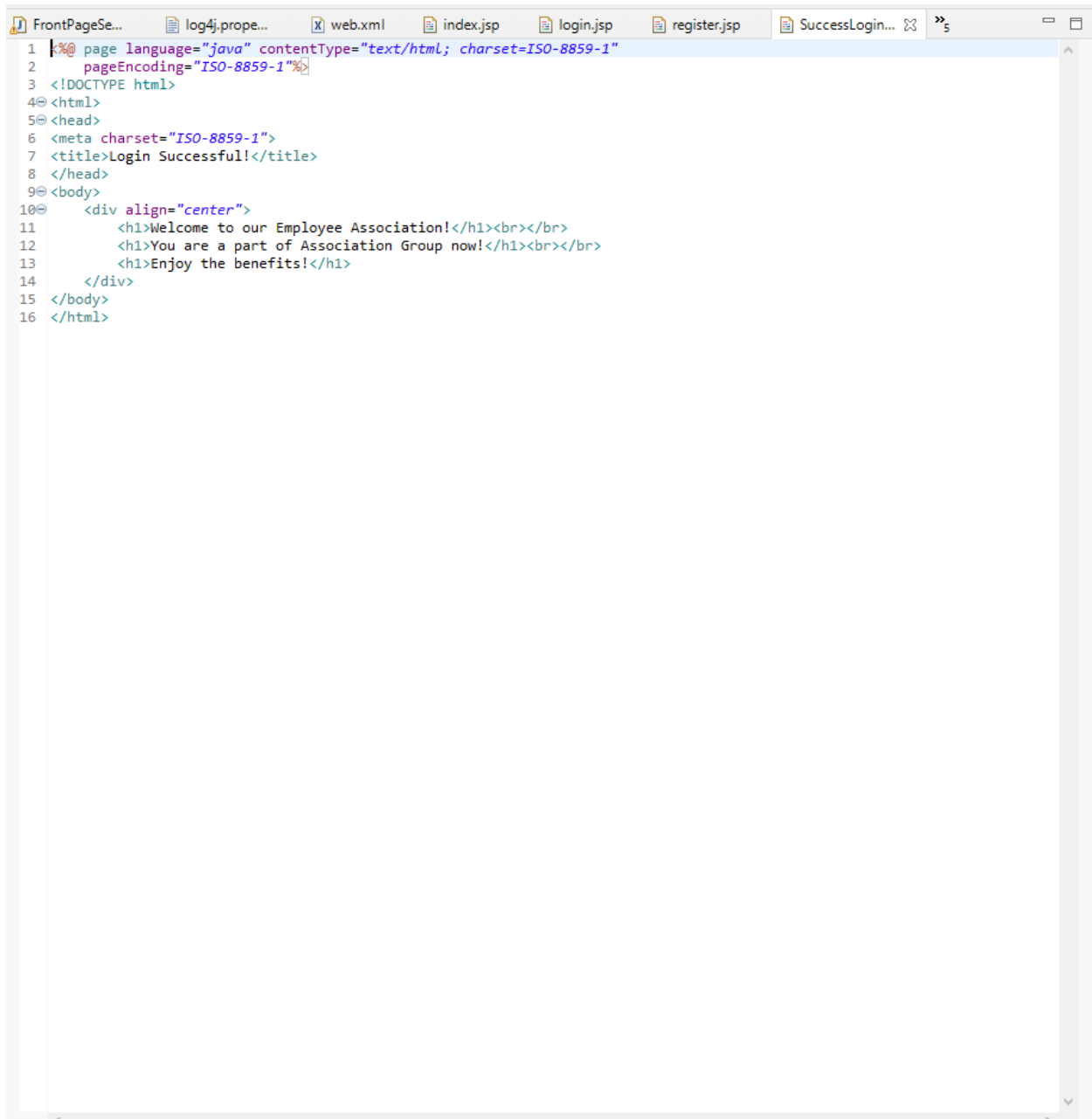
## Register.jsp

```
EmployeeRegi... FrontPageSe... log4j.prope... web.xml index.jsp login.jsp register.jsp »
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2   pageEncoding="ISO-8859-1"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6   <meta charset="ISO-8859-1">
7   <title>Employee Registration</title>
8 </head>
9 <body>
10   <form action="<%=request.getContextPath()%>/register" method="post">
11     <%
12       if (request.getAttribute("message") != null) {
13         <h2><%=request.getAttribute("message")%></h2>
14       <%
15       }
16     <%
17     >
18     <table>
19       <tr>
20         <td>First Name:</td>
21         <td><input type="text" id="uname" placeholder="First Name"
22           name="firstName" required></td>
23       </tr>
24       <tr>
25         <td>Last Name:</td>
26         <td><input type="text" id="uname" placeholder="Last Name"
27           name="lastName" required></td>
28       </tr>
29       <tr>
30         <td>User Name:</td>
31         <td><input type="text" id="username" placeholder="User Name"
32           name="username" required></td>
33       </tr>
34       <tr>
35         <td>Password:</td>
36         <td><input type="text" id="password" placeholder="Password"
37           name="password" required></td>
38       </tr>
39       <tr>
40         <td>E-Mail:</td>
41         <td><input type="text" id="email" placeholder="email"
42           name="email" required></td>
43       </tr>
44       <tr>
45         <td>Contact No:</td>
46         <td><input type="text" id="contactNumber"
47           placeholder="contactNumber" name="contactNumber" required></td>
48       </tr>
49       <tr>
50         <td>&nbsp;</td>
51         <td><button type="submit" class="btn btn-primary"
52           value="Register" name="submitButton">Register</button></td>
53       </tr>
54     </table>
55   </form>
56 </body>
57 </html>
```

Karthiga Gujuluva Ravichandran (grkarthikagr@outlook.c

## SuccessLogin.jsp





```
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2   pageEncoding="ISO-8859-1"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta charset="ISO-8859-1">
7 <title>Login Successful!</title>
8 </head>
9 <body>
10 <div align="center">
11 <h1>Welcome to our Employee Association!</h1><br></br>
12 <h1>You are a part of Association Group now!</h1><br></br>
13 <h1>Enjoy the benefits!</h1>
14 </div>
15 </body>
16 </html>
```

## Pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/maven-v4_0_0.xsd">
```

```
  <modelVersion>4.0.0</modelVersion>
```

```
<groupId>com.hcl</groupId>
<artifactId>userLoginRegisterPhaseTwo</artifactId>
<version>0.0.1-SNAPSHOT</version>
<packaging>war</packaging>

<name>userLoginRegisterPhaseTwo Maven Webapp</name>

<properties>
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  <maven.compiler.source>1.8</maven.compiler.source>
  <maven.compiler.target>1.8</maven.compiler.target>
</properties>

<dependencies>
  <dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>javax.servlet-api</artifactId>
    <version>4.0.1</version>
    <scope>provided</scope>
  </dependency>
  <dependency>
    <groupId>javax.servlet.jsp</groupId>
    <artifactId>javax.servlet.jsp-api</artifactId>
    <version>2.3.3</version>
    <scope>provided</scope>
  </dependency>
  <dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-core</artifactId>
    <version>5.4.27.Final</version>
  </dependency>
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.19</version>
  </dependency>
  <dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
    <version>1.18.16</version>
    <scope>provided</scope>
  </dependency>
  <dependency>
    <groupId>log4j</groupId>
    <artifactId>log4j</artifactId>
```

```

        <version>1.2.15</version>
        <exclusions>
            <exclusion>
                <groupId>com.sun.jmx</groupId>
                <artifactId>jmxri</artifactId>
            </exclusion>
            <exclusion>
                <groupId>com.sun.jdmk</groupId>
                <artifactId>jmxtools</artifactId>
            </exclusion>
            <exclusion>
                <groupId>javax.jms</groupId>
                <artifactId>jms</artifactId>
            </exclusion>
        </exclusions>
    </dependency>
    <dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>4.11</version>
        <scope>test</scope>
    </dependency>
</dependencies>

<build>
    <finalName>userLoginRegisterPhaseTwo</finalName>
    <pluginManagement>
        <plugins>
            <plugin>
                <artifactId>maven-clean-plugin</artifactId>
                <version>3.1.0</version>
            </plugin>
            <plugin>
                <artifactId>maven-resources-plugin</artifactId>
                <version>3.0.2</version>
            </plugin>
            <plugin>
                <artifactId>maven-surefire-plugin</artifactId>
                <version>2.22.1</version>
            </plugin>
            <plugin>
                <artifactId>maven-install-plugin</artifactId>
                <version>2.5.2</version>
            </plugin>
            <plugin>
                <artifactId>maven-deploy-plugin</artifactId>

```

```
        <version>2.8.2</version>
    </plugin>
    <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-compiler-plugin</artifactId>
        <configuration>
            <source>1.8</source>
            <target>1.8</target>
        </configuration>
    </plugin>
    <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-war-plugin</artifactId>
        <version>3.3.1</version>
        <configuration>
            <warName>user</warName>

    <warSourceDirectory>src/main/webapp</warSourceDirectory>
    </configuration>
    </plugin>
    </plugins>
    </pluginManagement>
</build>
</project>
```