Assignment1

Create a Spring Web Application that has functions to login, store and retrieve employee details.

* Login function for website:

+ Create a page with a hyperlink with "Login" value.

+ When hyperlink is clicked; Login form is shown for user login to the system.

+ When the user logs in incorrectly, the user is required to log in again.

+ Conversely, when the user logs in correctly (username = admin, password = blackcat), the user will be redirected to the new entry page with employee details.

* Employee details page:

+ The ‘Employee details’ form consists of "Emp ID", "Emp Name", "Birthday" and "Salary". A button "Add" should be available on the form.

+ Elements of form should check

* All input data can't be empty;
* 'Emp ID' should be 4 characters;
* 'Birthday' should be formatted 'dd/mm/yyyy';
* The minimum value of 'Salary' should be 500 USD.
* After the user fills information on the form and clicks the add button, Employee Details will show.
* Employee Details display information will be displayed on the website

**index.jsp**

<body>

<a href="login.htm">Login</a>

</body>

**login.jsp**

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@taglib prefix="form" uri="http://www.springframework.org/tags/form" %>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Login Employee</title>

</head>

<body>

<form:form action="loginProcess.htm" method="POST" modelAttribute="loginEmployeeForm">

<h1>Login</h1>

<table>

<tr>

<td><form:errors path="\*" cssStyle="color:red"/></td>

<td></td>

</tr>

<tr>

<td><form:label path="username">Username</form:label></td>

<td><form:input path="username" autofocus="true"/> <br>

<form:errors path="username" cssStyle="color:red"/>

</td>

</tr>

<tr>

<td><form:label path="password">Password</form:label></td>

<td><form:input type="password" path="password"/> <br>

<form:errors path="password" cssStyle="color:red"/></td>

</tr>

<tr>

<td></td>

<td><input type="submit" value="Sign in"/></td>

</tr>

</table>

</form:form>

</body>

</html>

**employeeDetails.jsp**

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Employee Details</title>

</head>

<body>

<h1>Welcome To <span>${employeeDetails.empName}'s</span> Information</h1>

<table>

<tbody>

<tr>

<td>Employee ID</td>

<td>${employeeDetails.empId}</td>

</tr>

<tr>

<td>Employee Name</td>

<td>${employeeDetails.empName}</td>

</tr>

<tr>

<td>Employee Birthday</td>

<td>${formattedBirthday}</td>

</tr>

<tr>

<td>Employee Salary</td>

<td>$ ${employeeDetails.empSalary}</td>

</tr>

</tbody>

</table>

</body>

</html>

**employeeInformation.jsp**

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@taglib prefix="form" uri="http://www.springframework.org/tags/form" %>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Employee Information</title>

</head>

<body>

<table>

<thead>

<tr>

<th>

<h2><form:errors path="\*" cssStyle="color:red"/></h2>

</th>

<th></th>

</tr>

</thead>

<tbody>

<tr>

<td><form:label path="empId">Employee ID</form:label></td>

<td><form:input path="empId" autofocus="true"/> <br>

<form:errors path="empId" cssStyle="color:red"/></td>

</tr>

<tr>

<td><form:label path="empName">Employee Name</form:label></td>

<td><form:input path="empName"/> <br>

<form:errors path="empName"cssStyle="color:red"/></td>

</tr>

<tr>

<td><form:label path="empBirthday">Employee Birthday</form:label></td>

<td><form:input type="date" path="empBirthday"/> <br>

<form:errors path="empBirthday" cssStyle="color:red"/></td>

</tr>

<tr>

<td><form:label path="empSalary">Employee Salary</form:label></td>

<td><form:input path="empSalary" placeholder="ex: 500 USD"/> <br>

<form:errors path="empSalary" cssStyle="color:red"/></td>

</tr>

<tr>

<td></td>

<td><input type="submit" name="Add employee"/></td>

</tr>

</tbody>

</table>

</body>

</html>

**com.controllers**

**UsersController.java**

package com.controllers;

import com.models.Employees;

import com.models.Users;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.validation.Valid;

import org.springframework.stereotype.Controller;

import org.springframework.ui.ModelMap;

import org.springframework.validation.BindingResult;

import org.springframework.web.bind.annotation.ModelAttribute;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.servlet.ModelAndView;

@Controller

@RequestMapping("/")

public class UsersController {

@RequestMapping(value = "login", method = RequestMethod.GET)

public ModelAndView showLoginForm() {

return new ModelAndView("login", "loginEmployeeForm", new Users());

}

@RequestMapping(value = "loginProcess", method = RequestMethod.POST)

public String loginEmployeeProcess(@ModelAttribute("loginEmployeeForm") Users user, BindingResult br, ModelMap model) {

if (br.hasErrors()) {

br.rejectValue("message", "error.message", "Username or Password are empty. Please try again!!");

return "login";

}

if (user.getUsername().isEmpty() && user.getPassword().isEmpty()) {

br.reject("message", "Username or Password are empty. Please try again!!");

return "login";

}

if (user.getUsername().equals("admin")) {

if (user.getPassword().equals("blackcat")) {

model.addAttribute("username", user.getUsername());

model.addAttribute("addEmployee", new Employees());

return "employeeInformation";

} else {

br.rejectValue("password", "error.password", "Password incorrect. Please try again!!");

}

} else {

br.rejectValue("username", "error.username", "Username incorrect. Please try again!!");

}

return "login";

}

@RequestMapping(value = "/add", method = RequestMethod.POST)

public String showEmployeeInformation(@ModelAttribute("addEmployee") @Valid Employees employee, BindingResult br, ModelMap model) {

if (br.hasErrors()) {

model.addAttribute("addEmployee", employee);

return "employeeInformation";

}

if (employee.getEmpId() != null) {

model.addAttribute("employeeDetails", employee);

try {

SimpleDateFormat formatter = new SimpleDateFormat("yyyy-MM-dd"); // Assuming your original format

Date birthDate = formatter.parse(employee.getEmpBirthday());

formatter = new SimpleDateFormat("dd/MM/yyyy"); // Desired output format

String formattedBirthday = formatter.format(birthDate);

model.addAttribute("formattedBirthday", formattedBirthday);

} catch (ParseException ex) {

Logger.getLogger(UsersController.class.getName()).log(Level.SEVERE, null, ex);

}

return "employeeDetails";

}

return "employeeInformation";

}

}

**com.models**

**Employees.java**

package com.models;

import javax.validation.constraints.Min;

import javax.validation.constraints.NotNull;

import javax.validation.constraints.Pattern;

import org.springframework.format.annotation.DateTimeFormat;

public class Employees {

@NotNull(message = "Employee Id can't be empty!")

@Pattern(regexp = "^\\w{4,}$", message = "Employee Id must be at least 4 characters")

private String empId;

@NotNull(message = "Employee Name can't be empty!")

private String empName;

@NotNull(message = "Employee Birthday can't be empty!")

@DateTimeFormat(pattern = "dd/MM/yyyy")

private String empBirthday;

@NotNull(message = "Employee Salary can't be empty!")

@Min(value = 500, message = "Employee Salary should be at least 500 USD")

private Double empSalary;

/\* Getter and Setter - Constructor\*/

}

**Users.java**

package com.models;

public class Users {

private String username;

private String password;

/\* Getter and Setter - Constructor\*/

}