Registeration Form having javascript validations and stroting every user on the database

1. Create index.jsp file as follows

<html>

<head>

<script type=*"text/javascript"* src=*"js/text.js"*></script>

<style>

**body**{

background-color: *lightblue*;}

*#box*{

position:*absolute*;

border: *1px solid grey*;

height:*50px*;

width:*80px*;

margin: *170px 450px 200px 95px*;

}</style>

<head>

<body>

<h1>Registeration Form</h1>

<div id=*"box"*></div>

<form action=*"reg"* method=*"post"* name=*"user"* onSubmit="return fun()">

<table>

<tr>

<td>First Name:</td>

<td><input type=*"text"* name=*"fname"* placeholder=*"First Name"*/></td>

<td></td>

</tr>

<tr>

<td>Last Name</td>

<td><input type=*"text"* name=*"lname"* placeholder=*"Last Name"*/></td>

<td></td>

</tr>

<tr>

<td>Email</td>

<td><input type=*"email"* name=*"email"* placeholder=*"abc@domain.com"*/></td>

<td></td>

</tr>

<tr>

<td>Address</td>

<td><input type=*"text"* name=*"address"* placeholder=*"Address"*/></td>

<td></td>

</tr>

<tr>

<td>Mobile number</td>

<td><input type=*"text"* name=*"mob"* placeholder=*"9002345699"*/></td>

<td></td>

</tr>

<tr>

<td>Date of Birth</td>

<td><input type=*"date"* name=*"dob"* placeholder=*"dd-mm-yy"*/></td>

<td></td>

</tr>

<tr>

<td>Gender</td>

<td><input type=*"radio"* name=*"gender"* value=*"male"*/>Male</td>

<td></td>

</tr>

<tr>

<td></td>

<td><input type=*"radio"* name=*"gender"* value=*"female"*/>Female</td>

<td></td>

</tr>

<tr>

<td>Hobbies</td>

<td><input type=*"checkbox"* name=*"hobbies"* value=*"hobby1"*/>Playing Cricket </td>

<td></td>

<td></td>

<td><input type=*"checkbox"* name=*"hobbies"* value=*"hobby2"*/>Watching Football </td>

<td></td>

</tr>

<tr>

<td></td>

<td><input type=*"checkbox"* name=*"hobbies"* value=*"hobby3"*/>Reading Books </td>

<td></td>

<td></td>

<td><input type=*"checkbox"* name=*"hobbies"* value=*"hobby4"*/>Playing Basketball </td>

<td></td>

</tr>

<tr>

<td></td>

<td><input type=*"submit"* value=*"register"*/> </td>

<td></td>

</tr>

</table>

</form>

</body>

</html>

1. Create a Javascript file where javascript validation for the UserForm is implemented.

Text.js

**function** fun(){

**var** fname=document.user.fname;

**var** lname=document.user.lname;

**var** email=document.user.email;

**var** address=document.user.address;

**var** mob=document.user.mob;

**var** dob=document.user.dob.value;

**var** male=document.user.gender[0].checked;

**var** female= document.user.gender[1].checked;

**var** hobby1= document.user.hobbies[0].checked;

**var** hobby2= document.user.hobbies[1].checked;

**var** hobby3= document.user.hobbies[2].checked;

**var** hobby4= document.user.hobbies[3].checked;

**if** (fname.value==""){

window.alert("Please Enter First Name")

fname.focus()

**return** **false**

}

**if** (lname.value==""){

window.alert("Please Enter Last Name");

lname.focus()

**return** **false**

}

**if** (/^\w+([\.-]?\w+)\*@\w+([\.-]?\w+)\*(\.\w{2,3})+$/.test(email.value))

{

} **else** {

alert("You have entered an invalid email address!")

email.focus()

**return** (**false**)

}

**if** (address.value==""){

window.alert("Address cant be left blank")

address.focus()

**return** **false**

}

**if** (/^\d{10}$/.test(mob.value)) {

// value is ok, use it

} **else** {

window.alert("Invalid number; must be ten digits")

mob.focus()

**return** **false**

}

**var** data = dob.split("/");

// using ISO 8601 Date String

**if** (isNaN(Date.parse(data[2] + "-" + data[1] + "-" + data[0]))) {

window.alert("Invalid date; Enter the date in format dd/mm/yy")

dob.focus()

**return** **false**;

}

**if** (male==**false** && female==**false**){

window.alert("Check atleast one")

**return** **false**;

}

**if** (hobby1==**false** && hobby2==**false** && hobby3==**false** && hobby4==**false**){

window.alert("Atleast one hobby is required")

**return** **false**

}

}

1. Create web.xml file and configure it as follows

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

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://xmlns.jcp.org/xml/ns/javaee"* xsi:schemaLocation=*"http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd"* id=*"WebApp\_ID"* version=*"3.1"*>

<display-name>RegForm</display-name>

<welcome-file-list>

<welcome-file>index.jsp</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>action</servlet-name>

<servlet-class>org.apache.struts.action.ActionServlet</servlet-class>

<init-param>

<param-name>config</param-name>

<param-value>/WEB-INF/struts-config.xml</param-value>

</init-param>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>action</servlet-name>

<url-pattern>\*.do</url-pattern>

</servlet-mapping>

</web-app>

1. Create a Java Bean Class for storing form inputs.

Employee.java

**package** beans;

**import** javax.servlet.http.HttpServletRequest;

**import** org.apache.struts.action.ActionErrors;

**import** org.apache.struts.action.ActionForm;

**import** org.apache.struts.action.ActionMapping;

**import** org.apache.struts.action.ActionMessage;

**public** **class** Employee **extends** ActionForm {

String fname,lname, email, address, dob, mob, gender, hobbies[];

**public** String getFname() {

**return** fname;

}

**public** **void** setFname(String fname) {

**this**.fname = fname;

}

**public** String getLname() {

**return** lname;

}

**public** **void** setLname(String lname) {

**this**.lname = lname;

}

**public** String getEmail() {

**return** email;

}

**public** **void** setEmail(String email) {

**this**.email = email;

}

**public** String getAddress() {

**return** address;

}

**public** **void** setAddress(String address) {

**this**.address = address;

}

**public** String getDob() {

**return** dob;

}

**public** **void** setDob(String dob) {

**this**.dob = dob;

}

**public** String getMob() {

**return** mob;

}

**public** **void** setMob(String mob) {

**this**.mob = mob;

}

**public** String getGender() {

**return** gender;

}

**public** **void** setGender(String gender) {

**this**.gender = gender;

}

**public** String[] getHobbies() {

**return** hobbies;

}

**public** **void** setHobbies(String hobbies[]) {

**this**.hobbies = hobbies;

}

@Override

**public** ActionErrors validate(ActionMapping mapping, HttpServletRequest request) {

// **TODO** Auto-generated method stub

ActionErrors ae= **new** ActionErrors();

**if** (fname==**null** || fname.length()<1)

ae.add("fname", **new** ActionMessage("error.firstname.required"));

**if** (lname==**null** || lname.length()<1)

ae.add("lname", **new** ActionMessage("error.lastname.required"));

**if** (email==**null** || email.length()<1)

ae.add("fname", **new** ActionMessage("error.email.required"));

**if** (dob==**null** || dob.length()<1)

ae.add("fname", **new** ActionMessage("error.dob.required"));

**if** (mob==**null** || mob.length()<1)

ae.add("fname", **new** ActionMessage("error.mob.required"));

**if** (gender.equals(""))

ae.add("fname", **new** ActionMessage("error.gender.required"));

**if** (hobbies.length<1)

ae.add("fname", **new** ActionMessage("error.hobbies.required"));

**return** ae;

}

}

1. Create a Data Access Object(DAO) class for handling SQL operations and making database connections

UserResDAO

**package** beans;

**import** java.sql.\*;

**public** **class** UserResDAO {

**public** **void** insertData(String fname, String lname, String email, String address,

String gender, String mob, String dob, String[] hobbies) **throws** Exception {

Class.*forName*("org.mysql.jdbc.Driver");

Connection con= DriverManager.*getConnection*("jdbc:mysql://localhost/User", "root", "welcome");

**try**{

**try**{

Statement st= con.createStatement();

**int** value = st.executeUpdate("INSERT INTO user\_details(FNAME,LNAME, EMAIL, ADDRESS, MOB, DOB, GENDER, HOBBY1, HOBBY2, HOBBY3, HOBBY4) "

+ "VALUES('"

+ fname

+"','"

+ lname

+ "','"

+ email

+ "','"

+ address

+ "','"

+mob

+"','"

+dob

+"'',"

+gender

+"','"

+hobbies[0]

+"','"

+hobbies[1]

+"','"

+hobbies[2]

+"','"

+hobbies[3]

+"')");

System.***out***.println("+1 row effected"+ value);

}**catch** (SQLException ex){

System.***out***.println("SQL statement is not executed"+ ex);

}

con.close();

}**catch** (Exception e){

e.printStackTrace();

}

}

}

1. Create a controller class for finally executing the SQL statement and storing the same in the database.

UserController.java

**package** beans;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** javax.servlet.http.HttpSession;

**import** org.apache.struts.action.Action;

**import** org.apache.struts.action.ActionForm;

**import** org.apache.struts.action.ActionForward;

**import** org.apache.struts.action.ActionMapping;

**public** **class** UserController **extends** Action {

@Override

**public** ActionForward execute(ActionMapping mapping, ActionForm form, HttpServletRequest request,

HttpServletResponse response) **throws** Exception {

// **TODO** Auto-generated method stub

HttpSession session= request.getSession(**true**);

Employee UserForm= (Employee) form;

String fname= UserForm.getFname();

String lname= UserForm.getLname();

String email= UserForm.getEmail();

String mob= UserForm.getMob();

String dob= UserForm.getDob();

String gender= UserForm.getGender();

String address= UserForm.getAddress();

String hobbies[]= UserForm.getHobbies();

UserResDAO dao= **new** UserResDAO();

dao.insertData(fname, lname, email, address, gender, mob, dob, hobbies);

session.setAttribute("fname", fname);

session.setAttribute("lname", lname);

session.setAttribute("email", email);

session.setAttribute("mob", mob);

session.setAttribute("address", address);

session.setAttribute("dob", dob);

session.setAttribute("gender", gender);

session.setAttribute("hobbies", hobbies);

**if** (fname.equals("") || lname.equals("") || email.equals("") || address.equals("")||

mob.equals("") || dob.equals("") || gender.equals("") || hobbies.length<1)

**return** mapping.findForward("error");

**return** mapping.findForward("success");

}

}

1. Configure the struts-config.xml file as follows.

<?xml version=*"1.0"* encoding=*"ISO-8859-1"* ?>

<!DOCTYPE struts-config PUBLIC

"-//Apache Software Foundation//DTD Struts Configuration 1.1//EN"

"http://jakarta.apache.org/struts/dtds/struts-config\_1\_1.dtd">

<struts-config>

<form-beans>

<form-bean name=*"RF"* type=*"beans.Employee"*/>

</form-beans>

<global-forwards>

</global-forwards>

<action-mappings>

<action path=*"/reg"* name=*"RF"* input=*"/index.jsp"* type=*"beans.UserController"*>

<forward name=*"success"* path=*"/success.jsp"*/>

</action>

</action-mappings>

<message-resources parameter=*"beans/Messages"*/>

</struts-config>