

Homework 1 – Registration Servlet

In this assignment you are to implement a Registration Servlet. This Servlet will allow Users to create a new account on your Website.

Location

Your working servlet should be located at (remember to replace XX with your account):

<http://cs3.calstatela.edu:8080/cs320stuXX/Homework1/Register>

Required Form Fields

Your form should consist of the following four (5) HTML input elements:

1. Name

The name field will collect the User's full name. The name of the field should be **name**, and you should verify that at least two names (first and last) are entered. If more than two names are entered you should consider the last word as the last name, and all other words as the first name. The type of the name field should be **text**.

2. E-mail Address

The email address field will collect the User's e-mail address. The type of the email address field should be **text**. The email address field should be named **email**.

3. Password

While creating a new account, the User is expected to create a password to protect their account from being accessed by unauthorized Users. The password field will collect the User's password. The field should be named **password1**, and it should be of type **password**.

4. Re-Enter Password

Since the type of the password field is **password**, the User will be unable to see what they are typing. To ensure that they entered exactly what they intended, your form should prompt them a second time for their password. This name of this field should be **password2**, and it should be type **password**.

5. Submit Button

Lastly, your form should contain a button to submit all form data to the Server. The value of the button should read **Register**.

Validation

Your form should submit all form data back to itself. That is to say, the **action** attribute of your form should be: **Register**.

Upon submission, you should validate all form fields using the following criteria:

Name

- The name field must contain, at a minimum, two words.
- Words are defined as groups of alphabets
- Words are separated by one or more white space character
- If the name field is invalid, you should display an error message indicating that the submitted value is incorrect.
 - The error message should appear immediately below the input field, not at the top of the page.

Email

- The email field must contain a valid e-mail address.
- A string of alphabets and numbers does not constitute a valid e-mail address.
- Example inputs:
 - *Abcd* – This is not a valid e-mail address
 - *john@doe.com* - This is a valid e-mail address
- If the email field is invalid, you should display an error message indicating that the submitted value is incorrect.
 - The error message should appear immediately below the input field, not at the top of the page.

Password

- The password field must contain a number and one of the following special characters: `!@#$%^&*`
- The values of password1 and password2 must be identical.
- Example inputs:
 - *1!* – This is a valid password
 - *a@* - This is not a valid password (it does not contain a number)
 - *12345* – This is not a valid password (it does not contain a special character)
- If the password field is invalid, you should display an error message indicating that the submitted value is incorrect.
 - The error message should appear immediately below the input field, not at the top of the page.

Sticky Form

If the form does not validate, you should redisplay the form keeping all valid inputs populated in each valid form field. That is to say, your form should be “sticky”. If a

value is invalid, you should not keep the value. This applies to the password fields, too!

Successful Registration

If all submitted form fields validate successfully, you should display a welcome message that reads, "Welcome, <First Name>!" *Note, you should replace <First Name> with the User's first name.*

Below the welcome message should be an anchor tag that links to:

<http://cs3.calstatela.edu:8080/cs320stuXX/Homework2/Login>

Remember to replace XX with your account number.

The text of the anchor should read, "Login".

Grading

If your form elements are not correctly named, you will lose a significant amount of points.

Name Field

- Validation – 20pts
- Sticky – 20pts

E-mail Field

- Validation – 25pts
- Sticky – 20pts

Password Field

- Validation – 25pts
- Sticky – 20pts

Re-Enter Password Field

- Sticky – 20pts

Submit Button

- Value – 10pts

Successful Registration

- Welcome Message - 20pts
- Hyperlink – 20pts

Total: 200

Important Notes

- You must submit all source code (Java, JSP, HTML, XML, CSS, images) to CSNS to receive credit. Failure to do so will result in a grade of zero (0) for this assignment.
- You must submit an Assignments.html page to CSNS that contains a link to your working CS3 demo. Failure to do so will result in a grade of zero (0) for this assignment.
- You must not modify your CS3 demo after the due date/time, unless you have received your grade for the assignment. Modifying your after the deadline, but before receiving your grade, will be considered cheating.
- Your demo must work on CS3 or you will not receive credit for the assignment.
- You must deploy your working demo to the specified URL or you will not receive credit.
- Lastly, you must not use any client-side validation. This includes, but is not limited to:
 - JavaScript
 - Use of “required” attributes within the input tags themselves.
 - All validation must occur on the server side to receive credit.