

Released on: Nov 15, 2015

Due on: Dec 5, 2015 - 23:59

Value: 7%

Objective

The objective of assignment 3 is to enhance your knowledge and skills to HTML5, CSS3, JavaScript and DOM through a task of building up a HTML form of registering user to a company website and performing client-side validations.

Requirements

1. Create an HTML5 document (**assign3.html**), CSS3 file (**a3.css**) and a JavaScript file (**a3.js**) under a folder called **assign3** of your Zenit account. Download the [a3-top-level-domain-names.js](#) file; then upload it to the same folder on Zenit and embed the JavaScript file into the assign3.html file.
2. In assign3.html, create a form (name: **form1**) to accept user's input for creating user account. Set the title of the page with "**Sign Up**". Your assignment 3 page appearance should **mimic** the [screenshot from this link](#).
3. In the form, create the following form controls/fields as specified:
 - Field "**Email**": type of email, autofocus and required. User needs the email address for logging in the website.
 - Field "**Password**": type of password, required.
 - Field "**Retype Password**": type of password, required.
 - Field "**Country**": select element with options (dropdown menu), all country in the world must be on the list, required. Country names will be showed on the dropdown list; while the values for the options will be 2-letter country codes. The country names and country codes are available from the JavaScript object array **countries** in the provided file **top-level-domain-names.js**. You need to use the object array and HTML DOM to create the field "**Country**". Make **Canada** selected by default.
 - Fields "**First Name**" and "**Last Name**": text, required, arranged in the same line.
 - Field "**Job Title**": text, the user's current job title, required.
 - Field "**Phone Number**": type of tel, user needs to enter his/her phone number, required. This field must show the telephone input format using placeholder.
 - Field "**Your Company**": text, the company that the user currently works for, required.
 - Field "**Address**": text, mail address, required.
 - Field "**Address2**": text, second part for address.
 - Field "**City**": text, city of the address, required.

- Field "**Province**":
 - When the user selects "Canada" in the Country field, the "**Province**" field should have the type of select with options (dropdown menu). Required.
The items for the options are "-Select-" (as the top one) as well as the 10 provinces and 3 territories of Canada, required. The full names of provinces and territories will be showed up in the sequence of **Alberta, British Columbia, Manitoba, New Brunswick, Newfoundland and Labrador, Northwest Territories, Nova Scotia, Nunavut, Ontario, Prince Edward Island, Quebec, Saskatchewan** and **Yukon**; while the values for option elements must use standard 2-letter abbreviations.
 - When the user selects a country other than "Canada" in the Country field, the "**Province**" field should have the type of text, required.
- Field "**ZIP/Postal Code**": text, the ZIP or Postal code of the address, required.
- Field "**How did you hear about us?**": type of radio button. A fieldset element is used to group these radio buttons. Give appropriate values for these radio buttons.
- The last field: checkbox, optional, if the user allow to send email on Senada Products, Service and Events.
- In addition, the form must have both submit and reset buttons with the values of **Create Account** and **Reset**.

4. The form is required to use **post** method and submit to the post tester:

<https://formpost.azurewebsites.net/home/test>. The autocomplete feature should be turned on.

5. All form fields/controls should have id and name attributes; each of the form fields/controls should have a label tag apply to it if possible. Required fields should be lead by * sign in red color.

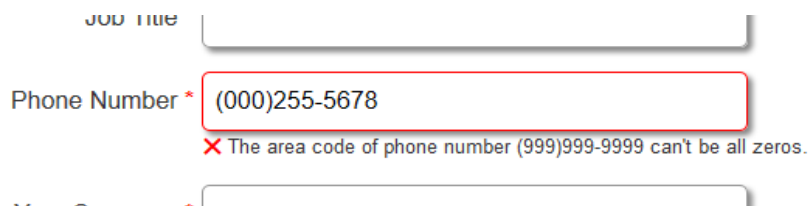
6. You are required to use CSS to style and mimic the page, including aligning form fields, font color/size, background, border, shadow, and etc.. HTML tables are not allowed for aligning form fields in the assignment. When a field is focused, the field border should be showed as green color; while user's input is invalid, the border should be changed to red color.

7. Some form fields need validation using JavaScript. But Regula Expression is not allowed in the assignment. The validation rules are as following:

- 1) Field "**First Name**" and "**Last Name**"
 - Must be present (cannot be empty string after trimmed)
 - Allowable characters
 - ✓ alphabetic characters lower case (a-z) upper case (A-Z)
 - ✓ apostrophe (')
 - ✓ hyphen (-)
 - ✓ Must have at least 1 alphabetic character (a-z) (A-Z)
- 2) Fields "**Password**" and "**Retype Password**"
 - Password must contain both upper and lower case letters
 - At least 1 number
 - At least 8 characters long

- The value of the 2 fields must be identical unless one of them is an empty string.
- 3) Field “**Phone Number**”
- Must be present
 - Must be in the format of (999)999-9999 (‘9’ here means digit only)
 - The area code (**999**)999-9999 can't be all zeros
 - The actual phone number (999)**999-9999** can't be all zeros
- 4) Field “**Province**”: this field needs to be validated only when “Canada” is selected in the “Country” field and the “Province” is displayed as a dropdown menu.
- Must select one of these provinces or territories (not the “-Select-” is selected).

8. For the fields that require validation, your code should validate the field upon user make change to the field. Show only one error message for a field at a time. Appropriate messages must be showed for the invalid field. For example:



The screenshot shows a web form with two input fields. The first field is labeled 'JOD Title' and is empty. The second field is labeled 'Phone Number *' and contains the text '(000)255-5678'. A red border highlights the input field, and a red error message is displayed below it: '✖ The area code of phone number (999)999-9999 can't be all zeros.' Below the error message is a third input field, which is also empty.

9. All field input must be valid before sending the form to server.

10. Your assignment must pass **W3C HTML validation**, **CSS validation** and **be free of JS run-time Exceptions**. But no JSLint check is required for JS file.

Evaluation

The evaluation for this assignment will be based on adherence to all of the requirements outlined above including:

- Adherence to the coding standards.
- Quality of the JavaScript technical features/usage/function (syntax, readability, documentation and indentation).
- The content.
- The overall appearance, organization and presentation.

Submission

Leave your work on zenit account under the assign 3 folder after completion; the instructor will check your work after the due date. No explicit submission, such as email and printout.

Leave your work untouched after completion/submission. You need to realize that every modification to the HTML, CSS or JavaScript files (while “saving”) is changing the date stamp of your work.

Late submissions will be penalized a 10% for the first day and a 5% for each day after that, up to a maximum of 5 days in total. Submission after 5 days will not be accepted.