Released on: July 18, 2015

Due on: Aug 8, 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 signing up for a job search 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. Download **top-level-domain-names.js** and store it in the same folder.

2. In assign3.html, create a form (name: **signup**) 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 **signup**, the user input information will be grouped in to four sections: *Basic information*, *Education*, *Work Experience* and *Attachments*. Add the following elements to each group in the form:

   a) Basic Information
   - 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 "**First Name**": text, user needs to input his/her full name, required.
   - Field "**Last Name**": text, user needs to input his/her full name, required.
   - Field "**Phone Number**": type of tel, user needs to enter his/her phone number, required. This field should show the telephone input format using placeholder.
   - Field "**Address**": text, home address, required.
   - Field "**City**": text, city of the address, required.
   - Field "**Province**": type of text with an element of datalist - the 10 provinces and 3 territories of Canada, required. The full names of provinces and territories will be showed up in the datalist <mark>and also be the values sent to server; while the values for option elements must use standard 2-letter abbreviations</mark>.
     The ten provinces are *Alberta*, *British Columbia*, *Manitoba*, *New Brunswick*, *Newfoundland* and *Labrador*, *Nova Scotia*, *Ontario*, *Prince Edward Island*, *Quebec*, and *Saskatchewan*. The three territories are *Northwest Territories*, *Nunavut*, and *Yukon*.
   - Field "**Country**": select element with options, 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

object array **country_list** in the provided file **top-level-domain-names.js.** You need to use HTML DOM and the object array to create the field "**Country**". Make **Canada** selected by default.

b) Education
- Field "**School**": text, the school that user/student is attending or the last attended, required.
- Field "**Program**": text, program in above school for the student, required.
- Field "**Education Level**": dropdown list using select and option elements, required. The **Education** Level should choose from the items of *–select–, Technical Diploma*, *College Diploma*, *Bachelor's Degree*, *Master's Degree* and *Others*. The option tag for the **"–select– "** should has the value of "0" <mark>and or</mark> an empty string.
- Field "**Education Status**": type of radio, required. The value should choose from the list of *Graduated*, *Current enrolled*, *Did not graduate*. When user clicks on the text beside a radio button, the button should change to checked if the button is in unchecked state.
- Field "**Graduation Date**": type of date, the date of graduation or anticipated graduation date, optional.

c) Work Experience
- Field "**Employer**": text, the current or last employer of the student, optional.
- Field "**Job Title**": text, the job title of the student in the above company.
- Field "**Job Duty**": textarea, description of job duties or responsibilities, optional.
- Field "**Worked Year**": type of number, the number of years for the student worked for his/her above employer, optional. The number user inputs will be in the range of 1 to 20.

d) Attachments
- Add an input with the type of file for attaching files to the application. There is no need to check the MINE type, but user can upload multiple files at a time.
- In addition, form **signup** must have both submit and reset buttons with the values of **Apply** and **Reset**.

4. The **signup** 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 the id and name attributes; each of the form field/control should have a label tag apply to it if possible. Required fields should be lead by * sign with red color.

6. You are required to use CSS to style the page and align form fields. HTML tables are not allowed for aligning form fields in the assignment.

7. Some form fields need validation using JavaScript. But Regular 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 characters (a-z) (A-Z)

2) Fields "Password" and "Retype Password"
   o Password must contain both upper and lower case characters
   o At least 1 number
   o At least 8 characters long
   o The value of the 2 fields must be identical unless one of them is an empty string.
3) Field "Phone Number"
   o Must be present
   o Must be in the format of 999-999-9999 ('9' here means digit only)
   o The area code **999**-999-9999 can't be all zeros
   o The actual phone number 999-**999**-**9999** can't be all zeros
4) Field "Education Status"
   o Must select one of these menu options

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

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.

Advanced

Complete the following tasks:

- Use JavaScript & DOM to add additional work history to the corresponding sections. In the history section, a ⊞ button should be provided for adding more sets of fields work experience to the form.
- Validate the **Email** field to ensure that the last (right) part of the email field must be a dot ('.') following by a valid top level domain name. If the top level domain name of an email entered by user is among the **domain_name** in the **domain_list** object array or among the **country_code** of the **country_list** object array in the **top-level-domain-names.js** file, the domain name is valid.

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.
- Note: the advance part is worth only about 8% out of the whole marks of the assignment 3.

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.