UNIVERSITY OF REGINA
Department of Computer Science

CS 215 – Web & Database Programming
Winter 2022

Assignment #3: Form Validation

Due: Monday February 28, 2022 by 11:55 PM

This is the third in the series of assignments for building a **collaborative note-taking** application. In the first two of these assignments, you designed and built an interface mock-up of the site and wrote some client-side code. For this one, you will use JavaScript to implement the form validation features. In the subsequent assignments, you will design and implement the database (Assignment #4), implement the back-end programming in PHP (Assignment #5), and implement AJAX-based updating (Assignment #6). Because the assignments build upon each other, you may wish to make modifications to your previous assignment submission before starting this assignment.

An important part of building an online application is verifying that the users have provided input in a format that is expected. Doing this on the client side allows the feedback to be provided quickly, and without the need to reload the Web page. For this assignment, you are **required to use DOM2 event registration, event handling functions, and provide all feedback by manipulating the DOM.** You are not permitted to use third-party libraries that simplify or abstract the JavaScript coding.

This assignment is divided into three different components:

A. Form Validation

In the previous assignment, you created four pages that each had some kind of a form on them. For each of these, your task is to implement JavaScript-based form validation. The following is a list of the pages, the form elements that are required, and the type of validation to be preformed.

1. Login Page
   - email address (valid email address format)
   - password (6 characters or longer, no spaces)

2. Sign-up Page
   - email address (valid email address format)
   - screenname (no spaces or other non-word characters)
   - avatar image (non-empty)
   - password (6 characters long, at least one non-letter character)
   - verify password (matches password)

3. Create Note Page
   - note name (non-blank; 256 characters or less)

4. View/Contribute Note Page
   - contribution (non-blank; 1500 characters or less)

For each of these form validation elements, you should highlight the portion of the form that contains the error, and add informative instructions to help the user to correct their mistake. As much as possible, you should write your validation functions to be reusable.

B. Dynamic Character Counter

Given the limitations on the number of characters that can be saved in a contribution (1500), it is also beneficial to give the user some feedback on the number of characters in their current contribution, as well as the number of characters left. This should operate dynamically as the user types in the field on the View Note Page. Should the character limit be exceeded, the user should be permitted to continue typing, but it should be made apparent that the limit has been exceeded.

Note that since we haven't implemented the back-end of the application, you do not need to save any of the data yet.

Since these pages will each be modified, you should ensure that they are still HTML5 compliant (following the syntax rules of XTHML), use CSS, and that there is a proper separation of the specification of the content from the specification of the presentation rules and the specification of the JavaScript code.

Grading Scheme

This assignment will be graded out of 10 marks, based on the following criteria:

1 mark: Form validation on Login Page.
2 mark: Form validation on Sign-up Page.
1 mark: Form validation on Create Note Page.
1 mark: Form validation on View/Contribute Note Page.
2 mark: Dynamic character counter on View/Contribute Note Page.
2 mark: Proper use of DOM document manipulation and DOM2 event registration.
1 mark: Valid HTML 5 that follows XHTML syntax rules.

Submissions

All the files for this assignment must be posted to your personal website on the Department of Computer Science servers (a separate directory for this assignment, with a link on your home page), as well as uploaded to UR Courses (a single zip file of your website directory).

A simple submission log file should be provided that includes your name, student number, class number, the URL of your web application, and a listing of the files you have submitted along with a short explanation of the purpose of each file.

Failure to provide these support documents will result in delays in the grading of your assignment and possibly a deduction in your grade.

Late submissions (up to 12 hours) will be accepted with an automatic 20% grade deduction (-2 marks). If there are exceptional circumstances that kept you from submitting your assignment on-time, you should consult with your instructor as soon as you are able to do so. See the syllabus for more details on the late policy for this class.