COMP 4610 GUI I

HW4

Using the jQuery Plugin/UI with Your Dynamic Table

PART 1: Validation Plugin

What This Part Is About

As we have discussed, it's good for a GUI program to anticipate and compensate for user errors, but it's even better if the program is designed to catch errors as soon as the user makes them and guide the user toward correcting them. Remember the three pillars of good error messages:

- identify **where** the error occurred — direct the user to the precise **location** of the error
- identify **why** the error occurred — explain exactly **what** the problem is
- suggest **how to correct** the error — provide information to **guide** the user to correcting his/her entry

In our previous assignment, you were to do this using either "straight" JavaScript or jQuery by processing events such as blur, focus, and submit. For this assignment you are required to do the validation using the jQuery Validation plugin (http://plugins.jquery.com). This plugin will allow you to do more powerful validation and make your application more responsive to users. (As you know, jQuery is a JavaScript library built entirely in JavaScript, so it is *possible* to do everything that the jQuery Validation plugin does using "straight" JavaScript, but it is much, much more work.)

What You Are To Do

1. Begin by copying your program for the previous assignment — or at least your main HTML page — to a new file so that you can modify it without destroying your previous work.
2. Modify your page to validate all data entered by the user using the jQuery Validation plugin (http://plugins.jquery.com) or found here (https://jqueryvalidation.org/) and prevent the form from being submitted (or the table from being regenerated) if the user's entry contains an error. If there is no error, regenerate the table as in the previous assignment.
3. Explore the various jQuery Validation plugin options carefully. Read the text, study examples below, and search for additional examples online. Customize the error messages displayed by the plugin so that they make sense in the context of your application. Customize where the error messages are placed to maintain complete control over how your page looks.
4. Test your page thoroughly. Make sure that you handle each possibility.

5. Have a friend run your application and try to "break" it. Plug any "holes" that your friend finds in your application and/or its validation scheme.

How You Will Be Graded

This part of the assignment will be graded on a 20-point system with points awarded as follows. Please note that the lists of features provided below are not meant to be exhaustive. They are merely representative of the types of things we are looking for in each grading category. As always, 20% of your grade is for documentation.

Criteria (numbers in parentheses are the points that can be earned for that bullet item)	***Possible Points***
Program Integrity / Design • (4) validation is accomplished using the jQuery Validation plugin • (4) error messages are precise and relevant and helpful to the user • (4) error messages are positioned reasonably and direct the user to the error location • (2) error messages suggest how to correct the error • (2) page does not reload if there is an error in the entered parameters	16
Source Code Documentation and Formatting • user name and pertinent contact information appear in all source files • ***all*** files contain adequate explanatory documentation that is meaningful and does not merely echo code • ***all*** files are properly indented and formatted with adequate white space for readability • ***any sources used are cited in comments embedded within code*** • **any missing documents/link**	4

PART 2: jQuery UI Slider and Tab Widgets

What This Part Is About

This assignment introduces you to the jQuery User Interface (UI) library. The purpose of doing this is to explore yet another JavaScript library to see its similarities and differences as compared to the library that we have already looked at: the jQuery Validator plugin. The more of these libraries you look at, the better understanding you will have about how they are built. And the better you understand that, the better you will be able to construct an industry-standard library yourself.

This part will look at two jQuery UI components: the slider and the tabbed interface. You could use the HTML5 slider, and I know that some of you have already programmed your own tabbed interface for previous assignments, either from scratch or using a different JavaScript library. However for this assignment you must use the components in the jQuery UI library.

Two chapters of Dan Wellman's book on the jQuery UI library below are for you to read and use as reference material for this assignment. These chapters are not from the latest edition -- they reference version 1.8 of the jQuery UI library, not version 1.10 as in Wellman's latest book -- but they will suffice for this assignment.

- Chapter 3: Using the Tabs Widget
- Chapter 6: The Slider Widget

You should use the latest jQuery UI library version, which is downloadable from http://jqueryui.com/download. And of course, you can also reference one of the JQuery UI CDNs if you'd prefer. Here are some of the CDN links available. Note that the library requires both JavaScript and CSS files.

- https://code.jquery.com/ui/jquery-ui-git.js -- uncompressed
- https://code.jquery.com/ui/1.11.4/jquery-ui.min.js -- minified
    - themes (CSS files): black-tie | blitzer | cupertino | dark-hive | dot-luv | eggplant | excite-bike | flick | hot-sneaks | humanity | le-frog |mint-choc | overcast | pepper-grinder | redmond | smoothness | south-street | start | sunny | swanky-purse | trontastic | ui-darkness |ui-lightness | vader
- https://ajax.googleapis.com/ajax/libs/jqueryui/1.11.4/jquery-ui.min.js -- minified JavaScript
- https://ajax.googleapis.com/ajax/libs/jqueryui/1.11.4/themes/smoothness/jquery-ui.css -- required CSS file

What You Are To Do

1. Read the two chapters for which PDF files are provided above.

2. Begin your development by copying the files for your part 1 to a new files (or a new directory) so that you can modify them without destroying your previous work. Do not remove the jQuery Validation plugin code! This part is an extension of the part 1. Everything that you did in the previous part 1 is still useful for this one.

3. Modify your page to do two things. These tasks are the crux of this assignment.

    a. Add jQuery UI sliders for each of your text input fields. Manipulating a slider should change the value in the corresponding text input field dynamically. That is, moving the slider should instantly change the text input field value. Likewise, typing into the text input field should change the value indicated by the slider.

Note: This is known as "two-way binding" between two form fields or two widgets, which is often accomplished using other JavaScript libraries such as AngularJS. But for this assignment, you are to implement the two-way binding yourself.

In addition, your table should update dynamically when either the slider is changed or a new text value is entered.

b. The second major modification is to implement a jQuery UI tabbed interface. Each time you create a new table, display it in a new tab and label that tab with the four parameters used to create it. Thus, you should have a single tab in which you enter parameters, and individual tabs for each table that gets generated. Think about how you want to implement this on your page, as you may not want to have the tabs in the standard location or format.

Provide a way to delete individual tabs and then provide a way to delete multiple tabs at the same time. This will take a little thinking to decide the best design for implementing this functionality. You might put the deletion controls on your data entry page, or you might put them on a separate "layout editing" page. There are many valid ways to do this. The ultimate design decision is up to you.

4. Test your page thoroughly. Try to anticipate all the errors that a user might make, whether intentional or unintentional. Make sure that you handle each possibility.

5. Have a friend run your application and try to "break" it. Plug any "holes" that your friend finds in your application and/or its validation scheme.

How You Will Be Graded

This part of the assignment will be graded on a 30-point system with points awarded as follows. Please note that the lists of features provided below are not meant to be exhaustive. They are merely representative of the types of things we are looking for in each grading category. As always, 20% of your grade is for documentation.

Criteria (numbers in parentheses are the points that can be earned for that bullet item)	***Possible Points***
Program Integrity / Design • (6) page exhibits two-way binding between each slider and its corresponding text field • (6) table updates dynamically when either the slider is changed or a new text value is entered • (6) page implements the jQuery UI tabbed interface • (3) individual tabs can be deleted	24

• (3) multiple tabs can be deleted simultaneously	
Source Code Documentation and Formatting • user name and pertinent contact information appear in all source files • ***all*** files contain adequate explanatory documentation that is meaningful and does not merely echo code • ***all*** files are properly organized (folders), indented and formatted with adequate white space for readability • ***any*** *sources used are cited in comments embedded within code* • no missing documents/link	6

Submitting Your Assignment for Grading

Please submit on Blackboard:

1. Readme file:
    a. GitHub URL where your application resides (for part 1 and part 2, separately)
    b. Link to your Github repository (for part 1 and part 2, separately)
2. All related code as a zip (for part 1 and part 2, separately)

** This assignment is based on 91.461 by Prof. Heines.*