

CSCI 4131 – Fall 2021

Internet Programming Assignment 2

Due Date: *Friday September 24th at 11:59pm*
Late Submission Deadline: *Sunday, September 26th at 5:59am*

1. Description

The objective of this assignment is to expose you to client-side JavaScript, Cascading Style Sheets (CSS), the Document Object Model, and jQuery. You will refactor and build upon your first homework assignment to accomplish this. Specifically, when you complete this your website will consist of the following 3 pages:

1. A Schedule Page (You should add your own events(s), and their info/picture)
2. A Form Input Page
3. A Widgets Page

This assignment description is 11 pages long, and you should complete it on your own.

The following tasks are required by this assignment

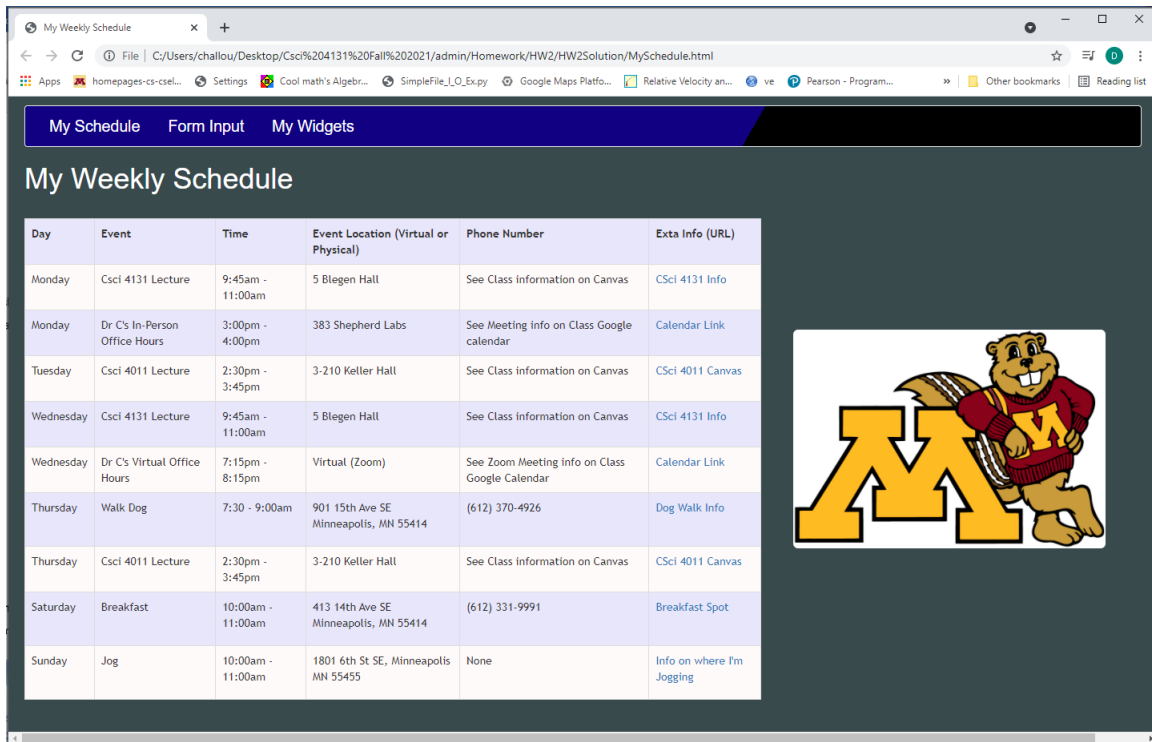
- i.) The navigation bar should be styled and updated to enable navigation between all three pages of your updated website.
- ii.) Update your Schedule page (***MySchedule.html***) as follows:
 - a. Each event should contain a thumbnail image related to the event in some way. When the mouse hovers over the event's row, the thumbnail should appear, and this image is shown enlarged next to the table. *Use your best judgment in selecting the pictures and do not use copyrighted material. Note: you can use pictures you create or take.*
- iii.) Create a new page named ***MyForm.html*** consisting of an eventinput form that will be used to add events to your table in later assignments.
 - a. The form should use HTML5 input elements and/or JavaScript to perform basic validation on the form's input fields for an entry in the table.
 - b. Upon successfully filling out the form, the form can be successfully submitted (i.e., submitted without displaying an error) to a URL provided, which will display the data entered in the form.
- iv.) Update your widgets page (***MyWidgets.html***) as follows
 - a. Add a Twitter **Profile Timeline** (that should replace the singular tweet on your page from assignment 1).
 - b. Add an Instagram post.
 - c. Add a password strength validator written solely in JavaScript (it cannot use jQuery) that behaves the same as the jQuery example we provide.
 - d. Organize the widgets side by side.
- v.) Style all your web pages with CSS. Specifically, create and use a single external style file, named ***style.css*** to style the pages on your Website (with one exception

specified in the Constraints section on page 9). Your web pages should not contain any inline or embedded styling. Your styling should be accomplished using CSS commands (i.e., CSS code) that are specified in your style.css file.

2. Required Functionality

Updates to the Schedule Page

Once the Schedule page is loaded into the browser, it should display the events information and the image of the event as shown in the screenshots below.



Day	Event	Time	Event Location (Virtual or Physical)	Phone Number	Extra Info (URL)
Monday	Csci 4131 Lecture	9:45am - 11:00am	5 Blegen Hall	See Class Information on Canvas	CSci 4131 Info
Monday	Dr C's In-Person Office Hours	3:00pm - 4:00pm	383 Shepherd Labs	See Meeting info on Class Google calendar	Calendar Link
Tuesday	Csci 4011 Lecture	2:30pm - 3:45pm	3-210 Keller Hall	See Class Information on Canvas	CSci 4011 Canvas
Wednesday	Csci 4131 Lecture	9:45am - 11:00am	5 Blegen Hall	See Class Information on Canvas	CSci 4131 Info
Wednesday	Dr C's Virtual Office Hours	7:15pm - 8:15pm	Virtual (Zoom)	See Zoom Meeting info on Class Google Calendar	Calendar Link
Thursday	Walk Dog	7:30 - 9:00am	901 15th Ave SE Minneapolis, MN 55414	(612) 370-4926	Dog Walk Info
Thursday	Csci 4011 Lecture	2:30pm - 3:45pm	3-210 Keller Hall	See Class Information on Canvas	CSci 4011 Canvas
Saturday	Breakfast	10:00am - 11:00am	413 14th Ave SE Minneapolis, MN 55414	(612) 331-9991	Breakfast Spot
Sunday	Jog	10:00am - 11:00am	1801 6th St SE, Minneapolis MN 55455	None	Info on where I'm Jogging




Figure 1. Event information displayed when page is initially displayed

- i) The rows of your Schedule table should be displayed in alternating colors in a manner similar as displayed in figure 1. Note, you can choose the colors, they don't have to be the same as in figure 1, but they should **enhance** the readability of your events.
- ii) The placeholder image shown (provided with the assignment) should be displayed as pictured next to your Schedule table. The table and picture should be able to fit on a reasonably sized screen.
- iii) When you hover over a row in the Schedule table with a pointing device (e.g., a mouse or your finger on a touch screen), a small (thumbnail) image of the location of the corresponding event should be displayed and the larger version of the image should be displayed next to the table. The larger image should persist even on mouse-out.

Specifically, as depicted in Figure 2, a small thumbnail image is displayed in each event's *Name* cell, and a larger version of the image is displayed next to the table. If the pointing device is moved to another row on the Schedule table, a thumbnail and large image of that event should be displayed. If the mouse moves off the table, the last large image displayed should remain, but the thumbnail image should disappear. When hovering from one row to another, the large image should maintain the same size.

Monday	Csci 4131 Lecture	9:45am - 11:00am	Virtual (Zoom)	See Zoom Meeting info on Class Canvas site	Csci 4131 Info
Monday	Csci 2081 Lecture	1:25pm - 2:15pm	Virtual (Zoom)	See Zoom Meeting info on Class Canvas site	Csci 2081 Info
Tuesday	Virtual Office Hours	12:00 - 1:00pm	Virtual (Zoom)	See Zoom Meeting info on Class Canvas site	Csci 2081 Info
Wednesday	Csci 4131 Lecture	9:45am - 11:00am	Virtual (Zoom)	See Zoom Meeting info on Class Canvas site	Csci 4131 Info
Wednesday	Csci 2081 Lecture	1:25pm - 2:15pm	Virtual (Zoom)	See Zoom Meeting info on Class Canvas site	Csci 2081 Info
Thursday	Walk Dog	7:30 - 9:00am	901 15th Ave SE Minneapolis, MN 55414	(612) 370-4926	Dog Walk Info
Friday	Csci 2081 Lecture	1:25pm - 2:15pm	Virtual (Zoom)	See Zoom Meeting info on Class Canvas site	Csci 2081 Info



Figure 2. Events page as displayed when pointing device hovers over a row of event information on event table

Add a Form Page:

Using HTML, CSS and JavaScript as necessary, create and add a Web page with a form for adding new events to your Schedule table (we will not implement its functionality in this assignment).

The form should use HTML-5 input elements wherever possible to obtain the following **mandatory** input:

Event Name; Day of Week; Start Time; Stop Time; Phone Number; Location; Extra Info; a uniform resource locator (URL) specifying information about the event, and the Submit button (Note, the form should be submitted using “**POST**” method). Optionally, you can also add a Clear button (not shown).

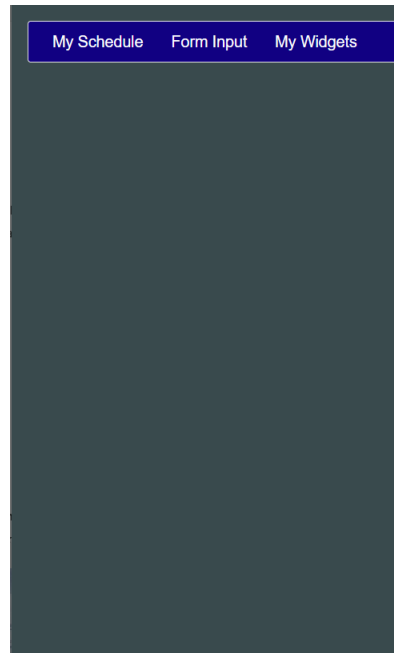


Figure 4. An example what your form might look like (the final styling is up to you).

On the form page, you must ensure all data entered on the form is syntactically legal input (using HTML-5 elements and/or JavaScript and regular Expressions). Specifically,

- i) All the fields are required to be filled.
Upon the submission, an error message should be displayed if any field is left blank.

Figure 5: Example error message displayed if there is an unfilled field.

- ii) The Event Name should accept only alphanumeric characters. Acceptable Alphanumeric characters are specified in the following ranges: number: 0-9, lower case characters: a-z, upper-case characters: A-Z and spaces.

Upon (or before) submitting the form, an error message should be displayed if non-alphanumeric characters are included in the event name fields.

Figure 6: Example error message displayed if there is an error in the event name input to the form.

- iii) “Day of Week” input is the HTML dropdown list as shown in Figure 7.

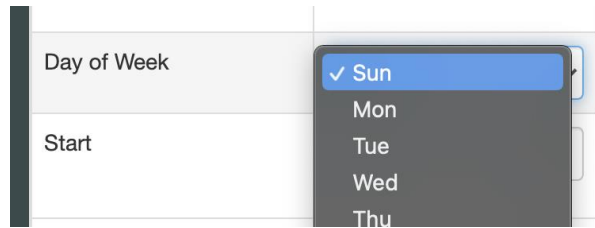


Figure 7: “Day of Week” Dropdown List

- iv) Finally, the URL should be syntactically validated to ensure it is formed as a syntactically legal URL (it may not really exist, but the form should be correct). An error should be displayed if a syntactically incorrect URL is entered.

Once the form is filled out with legal input (as pictured in Figure 8 below) and submitted (by left-clicking when your mouse is on the submit button), an HTML page with a table of your input data should be displayed, indicating that the form was successfully submitted (as pictured in Figure 9 on the next page).

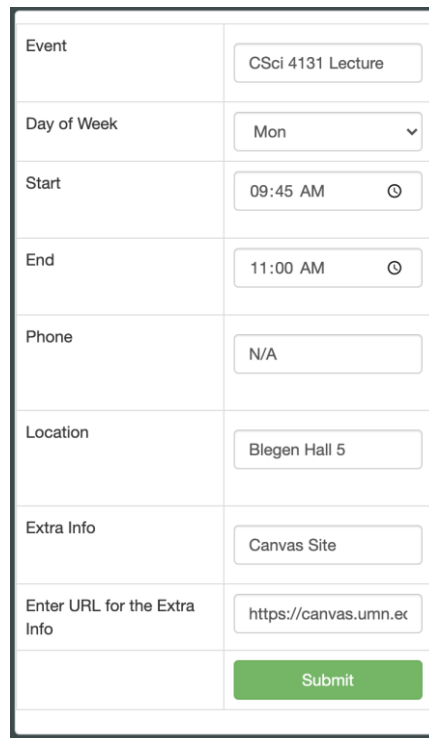


Figure 8: A form that has been successfully filled out by the user – there are no errors. When an error-free form is submitted, the action specified by the form’s action attribute is executed.

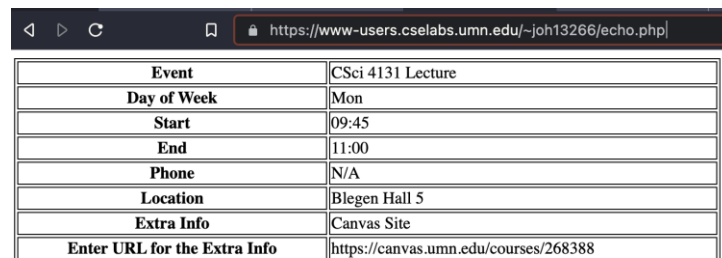
Specifically, the form should be submitted to the following server script we created with URL:

<https://www-users.cselabs.umn.edu/~joh13266/echo.php>. To achieve this, the action attribute on your form should specify the URL of the server script (above) - see https://www.w3schools.com/tags/att_form_action.asp for a discussion on how to accomplish this. **Note - you should not develop or deploy a server script for this assignment, use the one we have provided at the URL specified above.**

The method of the form must be “post” and the id attribute of each input element in the form must be the same as specified in the table below:

Input Element	ID attribute
Event	event
Day	day
Start	start
End	end
Phone	phone
Location	location
Extra Info	info
URL	url

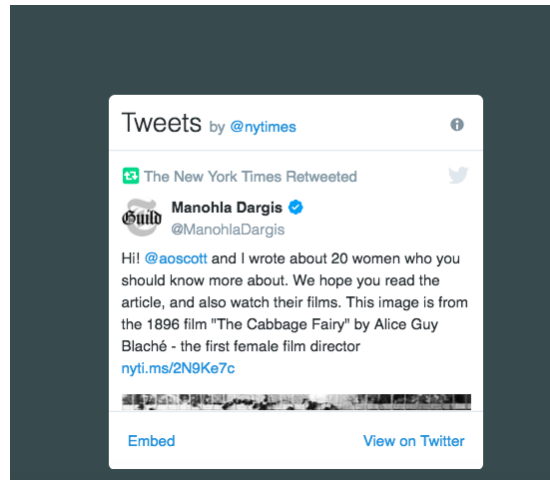
Make sure to double check that the information in the returned html table is the same as the input data you entered before submitting the form.



Event	CSci 4131 Lecture
Day of Week	Mon
Start	09:45
End	11:00
Phone	N/A
Location	Blegen Hall 5
Extra Info	Canvas Site
Enter URL for the Extra Info	https://canvas.umn.edu/courses/268388

Figure 9: Web page that is displayed upon successful submission of the form. Updates to the Widgets Page

1. **First**, replace the Twitter Tweet with a Twitter Timeline widget on the **My Widgets** Page, as pictured below in Figure 10. See the link <https://developer.twitter.com/en/docs/twitter-for-websites/timelines/overview.html> to get details about how to accomplish this.



Figure

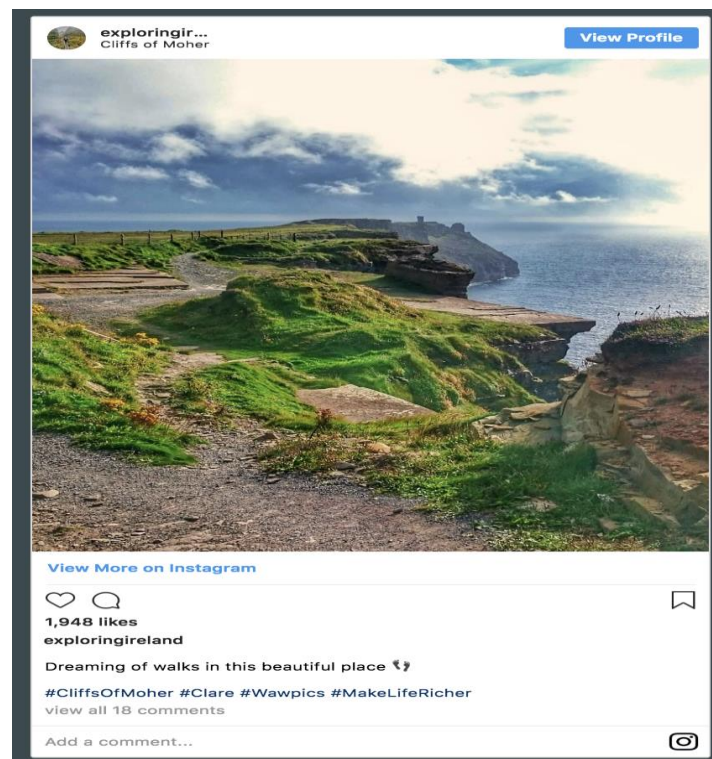
10: A twitter stream – this should replace the tweet from assignment 1

2. **Second**, embed an Instagram widget as pictured in Figure 11 on the next page onto the My Widgets page using following link as reference.

<https://www.instagram.com/developer/embedding/>

Note – If only the Instagram icon is showing, you should add `https` to the source link you obtain from Instagram (see:

<https://stackoverflow.com/questions/37322148/why-is-instagram-embed-code-only-showing-an-instagram-icon-not-the-image> for an explanation).



Figure

11: An Instagram widget should be added to your widgets page

3. **Third**, add a password strength check Widget on the My Widgets page. This JavaScript widget will check the strength of a password entered by the user. Use the jQuery example given in the link to help you write plain JavaScript code without using the jQuery library (**in other words, you must not use jQuery to implement this widget, you need to re-write it in plain JavaScript**).

Your JavaScript code should display the strength of the password as you type it by showing a progress bar layout as illustrated in Figures 12 (below) and 13 (on the next page).

<https://www.formget.com/password-strength-checker-in-jquery/>

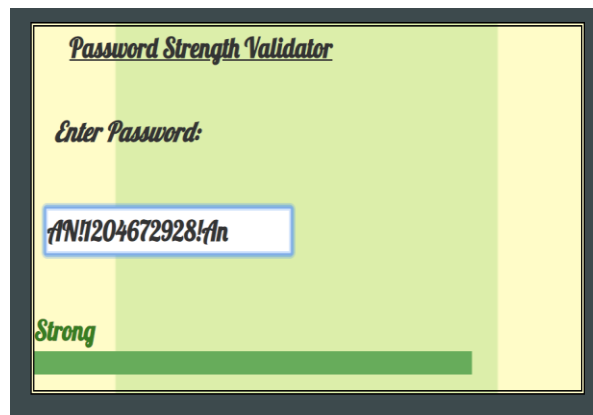


Figure 12: Example of the password checking widget indicating a strong password

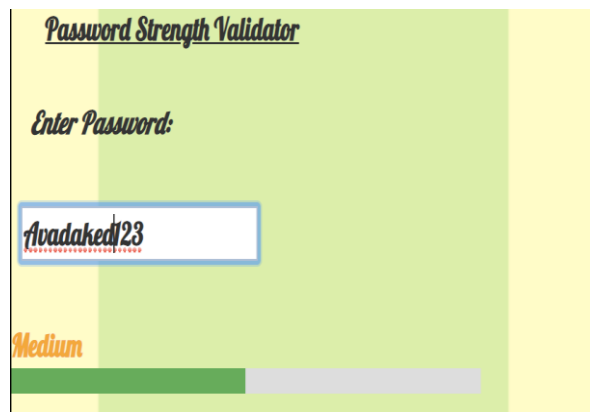
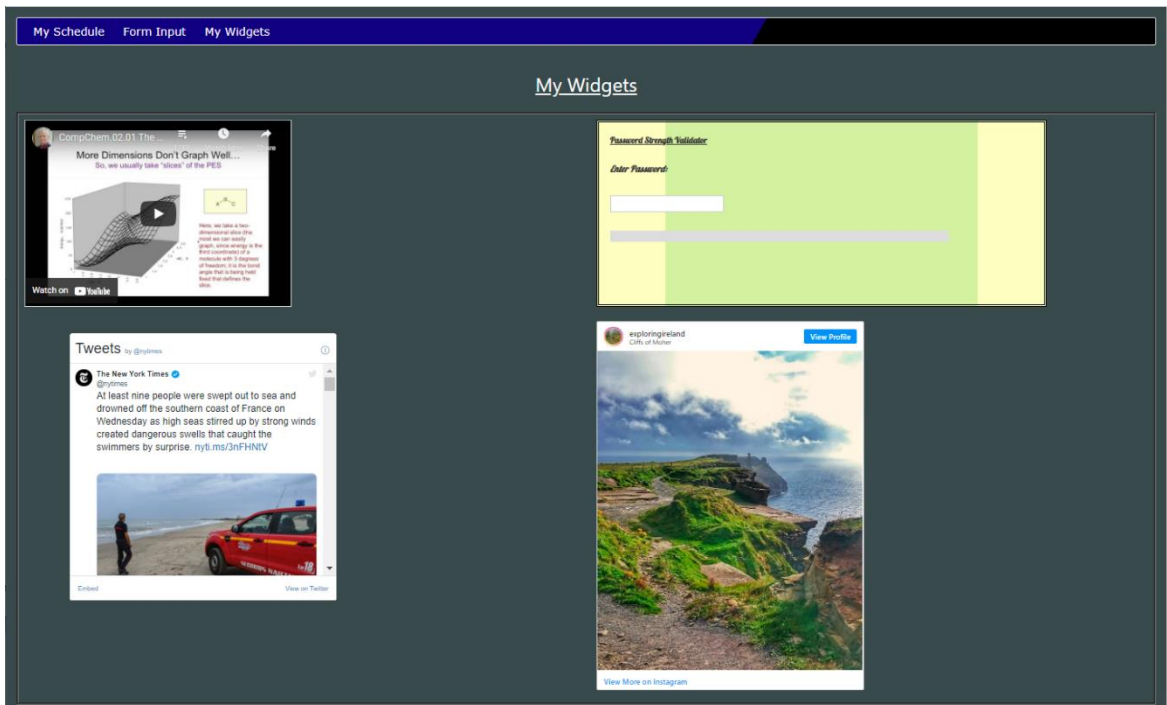


Figure 13: Example of the password checking widget indicating a medium strength password

Finally, you should style your widgets page, so the widgets are displayed side by side and displayed toward the center of the page.



F

figure
14:
Widge
ts as
displa
yed on
the
widget
s page

Hint: Side-by-side organization of the widgets can be achieved with grid layout.
https://www.w3schools.com/css/css_grid.asp

Constraints:

You must develop and use your own CSS styling for all items that are displayed. Do not use any style sheet (file) other than the one you have not developed yourself – with the exception of the style sheet that accompanies the password checking widget. Also, do not use embedded or inline styling in your final deliverable.

Violating this Constraint is a 15-point penalty.

Grading Criteria

- a. The navigation bar should be styled and updated to enable navigation between all three pages of your updated website. Note, the navigation bar styling can differ from the styling illustrated in this writeup **(5 points)**

- b. Update your Schedule page as follows:

A thumbnail image relating to the event should be displayed when your mouse hovers over the corresponding row in the table. **(10 points)**

A larger pop-up image of the thumbnail image should be displayed to the center right of the events table, image should be displayed at a reasonable size. **(15 points)**

When the mouse moves off the table, the last large image displayed should remain, but the thumbnail image should disappear. When images change, the size of the large image stays consistent. **(5 points)**

- c. Create a new page named ***MyForm.html*** consisting of an event input form that will be used to add event in later assignments. **(5 points)**
 - a. The form should use HTML5 input elements and/or JavaScript to perform basic validation on the form's input files from the events as specified in the required functionality discussion above **(10 points)**
 - b. Upon successfully entering data into the form, it can be submitted without displaying an error to the URL provided.
The page returned by the server should be a table displaying the information entered into the form. **(10 points)**
- d. Update your widgets page according to the requirements discussed previously as follows:
 - a. Add a Twitter **Timeline** (that should replace the tweet on your page from assignment 1). **(5 points)**
 - b. Add an Instagram widget. **(5 points)**
 - c. **Add the password strength validator with the styling that accompanies the widget to your widgets page (5 points)**
 - d. **BONUS** – rewrite the password strength validator so it uses JavaScript (it cannot use the jQuery provided or **any** jQuery) – and the resulting widget should behave the same as the jQuery in the example provided **(10 points - BONUS ITEM)**
 - e. The widgets are displayed side-by-side as shown in Figure 14. **(10 points)**
- e. Style all your web pages with CSS. Specifically, create and use a single external style file, entitled **style.css** to style the pages on your Website. Your web pages should not contain any inline or embedded styling. Your styling should be accomplished using CSS commands (i.e., CSS code) that are specified in your **style.css** file. **(10 points)**
- f. With the exception of the embedded widgets on the updated widget page, your HTML and CSS should files pass the w3 organizations validators (<http://validator.w3.org> and <https://jigsaw.w3.org/css-validator>) without errors. Warnings are accepted. **(5 points)**

SUBMISSION INSTRUCTIONS:

Your submission should be packaged in a tar file or zip file. When opened, it must create a directory named: '<Your UMN x.500 ID>' containing all of your files (HTML, CSS, Pictures, and any additional external JavaScript files (If used). You will be penalized 10 points if you do not do this correctly.

Submit your homework via the class Canvas item named Homework 2 Submission Link in the week 3 module on the class Canvas site.

Additional:

We have provided some images you can use with the assignment in the images folder with the assignment on Canvas. You are responsible for legally obtaining any other images you use.