

CMSI 370-01
INTERACTION DESIGN
Fall 2012

Assignment I204 Feedback

Chris Whiting

1e — Your color scheme for password strength is simple and straightforward, making use of the red-yellow-green cultural affordance to indicate password strength. Although the logic for determining the final color has some flaws (see below), the design itself is fairly clear. What is missing affordance-wise is some indication that your password strength field is different from a “vanilla” password field. Although this is not absolutely necessary (after all, the user sees the distinct behavior once he/she starts typing), it would still help communicate the user interface’s expected behavior better. (|)

3a — You clearly demonstrate user interface construction knowledge from the perspective of standalone web pages. What is specifically missing in this assignment, with regard to user interface construction, is the reusability aspect of being a “widget.” As written, your code would convert *every* div on a web page into a password strength field—not exactly a “widget” approach. (Bootstrap and jQuery UI show some good examples of how to design and implement widgets) (|)

3b — Your work shows a basic understanding of event-driven programming, but with some ways to go for full proficiency. One of your miscues, for example, is attaching your primary event handler from the wrong user interface element (which contributes to the confusing behavior that you have reported). Further, the approach of processing the password on a key-by-key basis, instead of looking at the entire password string itself (which is always available), speaks to some further gaps in understanding what events to handle in what situations. (|)

3c — Your source files are organized cleanly along MVC lines (HTML/CSS/JavaScript), showing that you at least knew not to “break” the structure already seen in sample code. (+)

3d — Although the approach is probably not the best for this particular widget, your key-by-key event handler does show some proficiency with low-level event handling. The convoluted logic (thus resulting in the occasionally not-what-you-expected color choices) speaks to some need for further practice though. (|)

4a — As we noted when we met in my office, your code has a few functionality gaps. The big issue really is the already-mentioned mismatch in your key-by-key approach vs. the “whole password string” one. Making that switch alone will clean up a lot of your logic. There are also a few inline comments in your JavaScript code, so check those out. In general, you have the beginnings set up, but not quite fully formed yet. (|)

4b — Proper widget structuring and reusability was not achieved in this version as already mentioned, and that does detract from your separation of concerns and MVC. (|)

4c — Your HTML and CSS are pretty readable, with good spacing and indentation, but your JavaScript is less so. Some of your expressions are too tight, and indentation can be inconsistent. This includes your *comment* indentation—I know they are comments and ultimately ignored, but you should still indent them in a manner that is appropriate to their location. You do this to maintain respect for the structure of the code, which in turn improves readability. (|)

4d — Your work shows use of some external resources and documentation, but not so much as to achieve completely reusable widget status, or even completely correct behavior. (|)

4e — Your commit log shows excellent phrasing and messages. (+)

4f — Submitted on time. (+)

CMSI 370-01
INTERACTION DESIGN
Fall 2012

Assignment I204 Feedback

Chris Whiting

(updated feedback based on 12/13/2012 commits; only reevaluated proficiencies are shown)

3a — You addressed the specific problem indicated previously, which was the conversion of every div on a web page into a password strength field, but you fell into that trap *again* by adding your password field event handler to every input on the web page. This is either a continued indication of misunderstanding, or a careless oversight, or lack of thoroughness (i.e., you did not test your code on a web page with additional div and input elements). Whatever the reason, it does not justify an increase in proficiency. (|)

3b — You use a more appropriate event now (input), and process the password string more cleanly. (+)

3d — Your password-processing logic is cleaner now, and yields better results. Hope you can carry this additional experience forward in future projects that require low-level event handling. (+)

4a — Your code's functionality has certainly improved, but the quality of the code itself still does not communicate "highly proficient programmer" to me. See the inline comments for additional (occasionally glaring) miscues that keep this proficiency from going over the top. Note that the main knock is on the code quality in general, and not the event handling functionality, which is why it hits *4a* but not *3d*. (|)

4b — Proper widget structuring and reusability is still not achieved, so this remains the same. (|)

4c — Your JavaScript spacing and indentation is slightly better than before, but enough to take you over the top. There is one particular faux pas that, really, should not happen at your level. Plus many spacing choices remain too tight while others can use additional linebreaks to improve readability. (|)