

## **Homework 5: Forms and Events (5 points plus up to 2 bonus points)**

### **Homework due Feb 14, 6AM.**

Each task is worth one point, with each subtask worth an equal fraction of the point.

*To get full credit on this assignment, everything must be contained in **one HTML page, one JavaScript page, and one CSS page**. Yes, you could create a multi-page setup to get around some of the requirements here, but the point of this exercise is to give you practice in responding to events while also giving you more practice with dynamic DOM manipulation.*

1. In an **HTML** page (or in JavaScript referenced from the HTML page), create a form that includes the following, laid out in such a way so that the user sees the connection between them (read on to the next parts to understand the connection).
  - a. A group of three radio buttons labeled "Log In", "New Account", and "Continue as Guest". "Log In" should be selected by default.
  - b. A regular textfield labeled as "Username"
  - c. A password-type textfield labeled as "Password"
  - d. A "Submit" button that is not visible by default)
2. When the user clicks on a radio button, adjust the fields that are shown using any method (such as field hiding/showing via JavaScript CSS manipulation or dynamic field creation/removal through JavaScript DOM manipulation), as follows:
  - a. If the user switches to "New Account", the fields should change to "New Username" "Password" and "Verify Password" (You can re-use the fields from Step A, or you can hide/show other fields, but have the fields remain in the same spot on the screen so they don't jump around and disorient the user).
  - b. If the user switches to "Continue as Guest", the fields should all disappear, leaving only the "Submit" button, which should become visible without disturbing the layout of the rest of the page.
  - c. If the user switches back to "Log In", the fields specified in #1 should reappear.
3. Add the following responses to form data entry:
  - a. If the user has typed something in all required fields (2 fields for the "Log In" screen and 3 fields for the "New Account" screen, show the "Submit" button.
  - b. If any of the fields are emptied (by the user deleting everything), hide the "Submit" button again. Make sure that the "Submit" button appears in the same place for all three cases.
4. In only the "New User" case, if the user has just clicked or tabbed away from a "Password" field, check the following:
  - a. If only one of the password fields is still blank, do nothing

- b. If both password fields have something filled in, compare the values in the two fields. If the values are not the same, put a red warning next to the password fields (in some sensible place) that says "Passwords must match!"
  - c. While the "Passwords must match!" warning is up, each time the user types a character in one of the password fields, re-do the comparison between the two Password field values. If the two match, then clear the "Passwords must match!" warning.
- 5. If the user presses the "Submit" button (which should only be visible when all required data is filled in!), do the following:
  - a. If the user has selected "Log In", then clear away all the form fields and replace the form with a greeting on the page (formatted however you want) that says "Welcome back, *username*!" (with username replaced by whatever the person entered)
  - b. If the user has selected "New User", then clear away all the form fields and replace the form with a message that says "Welcome, *username*!" Then check to see if the user's password is longer than 5 characters. If it is not, then add a message that says "CAUTION: Your password may not be very secure."
  - c. If the user has selected "Continue as Guest", then clear away all the form fields and replace the form with a message that says "Welcome, Guest!"
- 6. BONUS QUESTION: Add this prompt to item 5b: "Would you like to change your password now?" Then add a button or a hyperlink that, when clicked, presents a new form (replacing the greeting message) that contains the following:
  - a. Three password fields, appropriately labeled: One for the old password, one for the new password, and a third to verify the new password.
  - b. Follow the same parameters described above for password checking (make sure they match and warn the user if they don't) and showing the submit button (when all fields are filled in).
  - c. When the user clicks the submit button (which should only show when all fields are filled in), check the "old password" against the password that the user entered when creating the account. If the old passwords are the same, remove the form fields and put up a message saying "Your password has been changed." If the old passwords are not the same, then put up a red message next to the "Old Password" field saying "Wrong Old Password"
- 7. BONUS QUESTION: Add this prompt to item 5c: "Would you like to create an account now?" Then add a button or hyperlink that, when clicked, brings back up the original login/new account screen, but with the "New Account" radio button selected by default instead of the "Log In" button. This should use the same functions and objects that you used in the original screen and not be a complete duplication, so plan your conditional logic accordingly.