SOFE 4850U

Fall 2022

Assignment 1: Implementing an HTML/JavaScript Application

Due Date: Oct. 31st, 2022, 11:59 pm

This problem set asks you to build a small user interface that searches a list of words using the set of core technologies that drive most modern web applications, namely HTML, CSS, and JavaScript. We provide some backend code that does the heavy lifting (loading the word list and searching it). We also specify the design of the user interface. Your job is to implement it using HTML, CSS, and JavaScript.

To do this assignment, you will need to know how to:

- write HTML: create correct HTML pages that display in modern browsers
- use HTML form elements (like <button, <input>, and <select> to create a web application.
- use CSS to layout and style HTML elements
- use JavaScript (and jQuery) to add event handlers and bindings to respond to user input;

We provide you with the following:

- ASSIGNMET_ASSETS.zip: a zip file containing the code you will be editing for this problem set.

You can import this zip file directly into your IDE workspace or use whatever text editor you would like. In the root folder of the project are the following files:

- jquery-1.8.min.js: the code for jQuery, the JavaScript library you will be using for this assignment.
- words-small.json: a dictionary of 850 common English words in JSON notation.
- words-large.json: a dictionary of 45,407 words taken from the standard Linux /usr/share/dict/ words, in JSON notation.
- js: a backend class that represents a list of words and provides operations for filtering the list using a search pattern.
- word-finder.html: a skeleton file for your user interface.

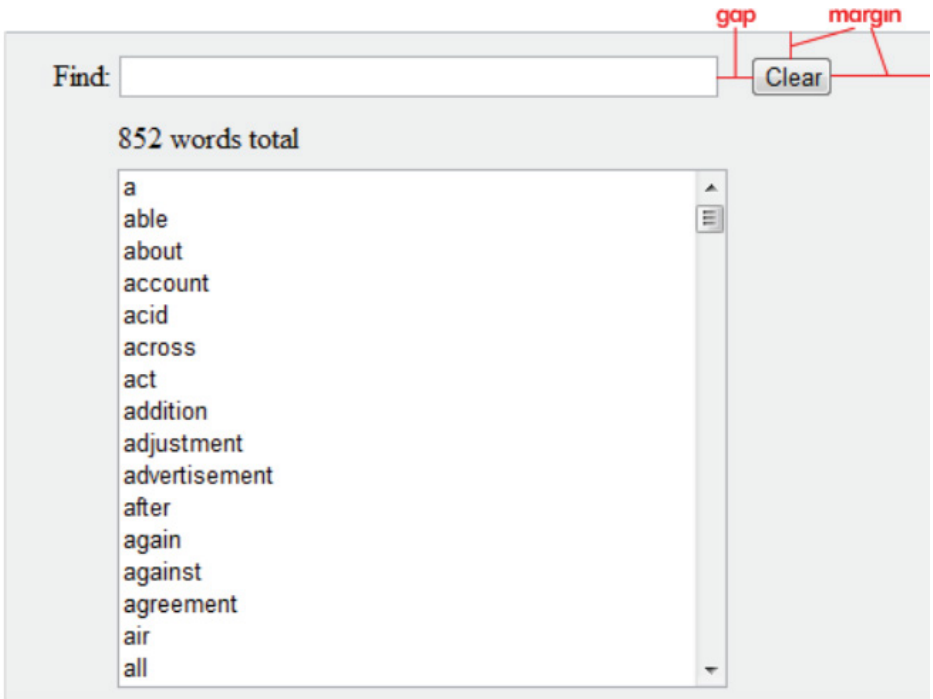
Feel free to change these files as you see fit.

We highly recommend installing the appropriate plugins for web development support if you use VS Code or similar IDEs.

Instructions

Basic Layout (30%)

First, build an interface that looks like this. There should be margins between and around the elements, as shown. No behaviour is required to earn the points for this part, just a static, fixed-size layout. The <select> widget (which is showing words in the figure) should have a handful of words in it. You need to use HTML form elements whenever appropriate.

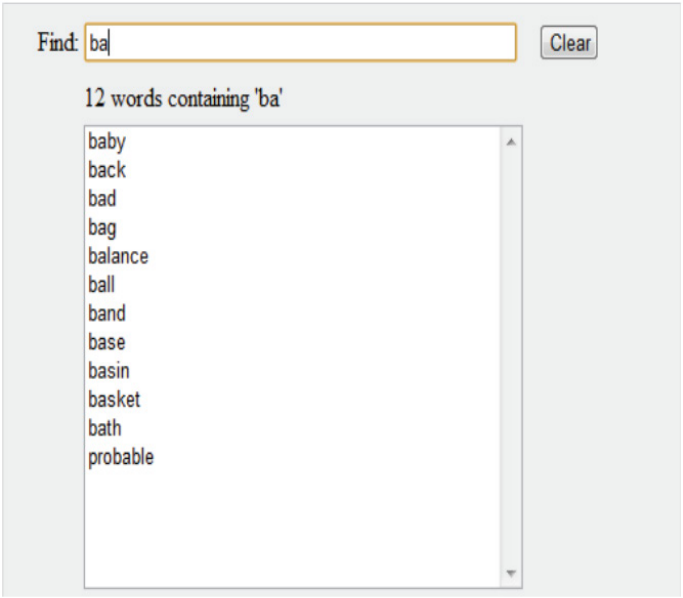


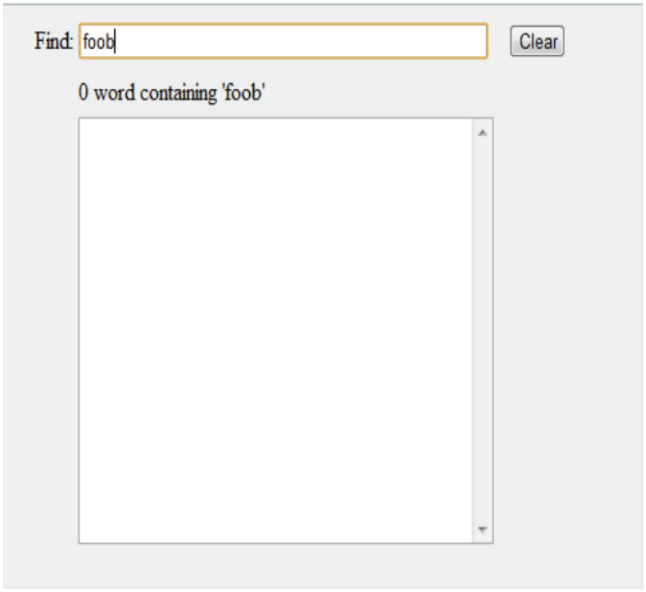
Displaying Words (20%)

Improve your interface so that when the program is first to run, the select element displays the small dictionary we gave you, as shown in the figure. The label above the select element should display the number of words in the list, as shown.

Searching (40%)

The Find input field contains the user's query. When the query is blank, the list box displays the entire word list, as shown above. Whenever the query changes, the list box immediately updates to display all words that contain the query text.



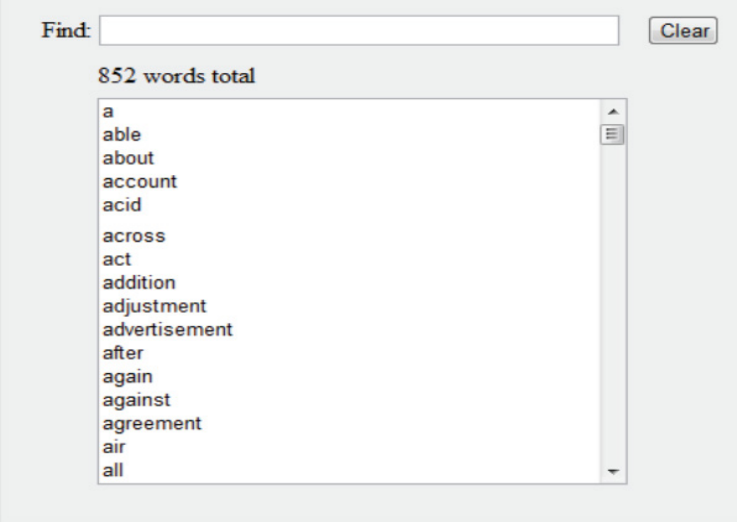


The list box should constantly update as the user types. Pressing Enter should not be necessary. (Hint: jQuery's event handling API makes this particularly simple, see the documentation here.) Be careful about which keyboard-related events you decide to listen to. If none of the words contain the query, the list box should be empty.

The Clear button should clear the query field, restoring the list box to displaying all words again.

Resizing (10%)

The window should be resizable, with all extra space going to the query field and the word list. You may need to use some JavaScript to set the height of the dynamically



What to Hand In

Package your completed assignment as a zip file that contains all your HTML, JS, and CSS files (including the JSON data).