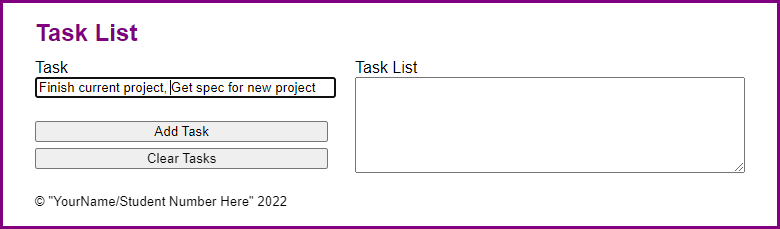
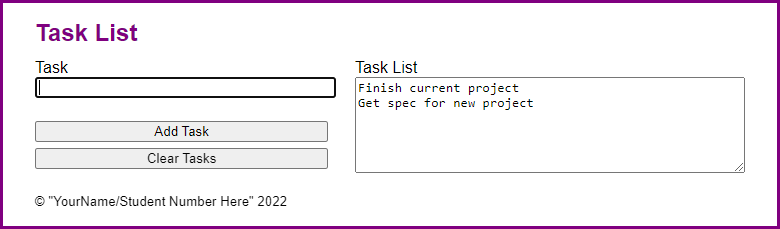
**Assignment 4-1 Instructions: Allow multiple task entries in the Task List application**

In this assignment, you’ll make an enhancement that allows you to enter multiple tasks separated by commas in a single entry.

Here is the enhanced application, with multiple tasks about to be entered:



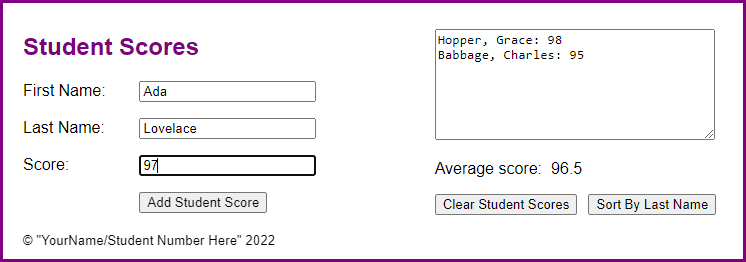
And here is the application after the multiple tasks have been entered:



1. Open the starter application(zipped) attached to this Assignment folder:
   1. Evaluations\Assignments\Assignment 4\Part1-task\_list.zip
2. Run the application and add two tasks, separated by a comma. Note that the tasks are stored as one task, exactly as you entered it.
3. In the JavaScript file, find the click() event method of the Add Task button. Then, find the code that adds the task entered by the user to the tasks array. Replace that code with code that works for one or more tasks in an entry.
   1. To do that, you can use the split() method of the String object to convert the user’s entry into an array. Then, you can loop through that array to add the new tasks to the tasks array.
4. In index.html, change the text in the <small> tag to your name and student number.

**Assignment 4-2 Instructions: Develop the Student Scores application**

In this assignment, you’ll develop an application that tracks student’s scores, tallies the average of the entered scores, and sorts the entered students by last name.



1. Open the starter application(zipped) attached to this Assignment folder:
   1. Evaluations\Assignments\Assignment 4\Part2-scores.zip
2. In the JavaScript file, note the start of a displayScores() function. In the ready event handler, note the starts of the handlers for the click events of the Add Student Score, Clear Student Scores, and Sort By last Name buttons. Note that the handler for the Add Student Score button validates the data. For valid data, it clears the add form and sets the focus on its first field. Also, the handler for the Clear Student Scores button ends by clearing the display area and setting the focus on the first name field.
3. Code two arrays outside of these functions, one for score values and the other for strings that display the students’ names and scores.
4. In the displayScores() function, add the code that calculates the average score of all the scores in the first array and displays it in the label with the id of “avr\_score”. Then, add the code that gets the students’ names and scores in the second array and displays them in the text area.
5. In the click event handler for the Add Student Score button, use the push() method to save the score in the first array and to save the name and score string (as shown in the text area) in the second array. Then, call the displayScores() function to redisplay the updated data.
6. In the click event handler for the Clear Student Scores button, add code that clears both arrays.
7. In the click event handler for the Sort By Last Name button, add code that sorts the students by last name and then re-displays the score information.
8. In index.html, change the text in the <small> tag to your name and student number.