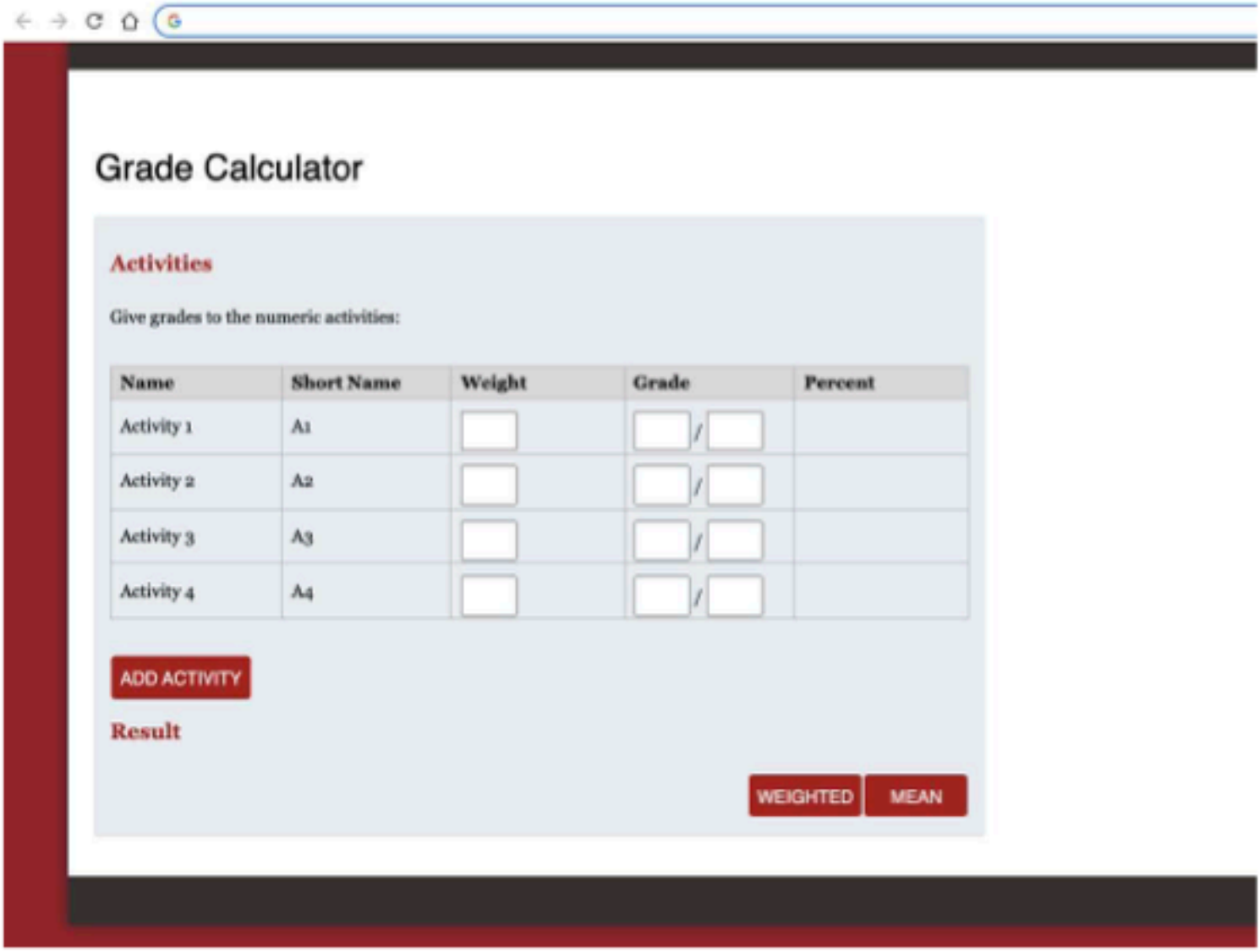
asn1

New Attempt

Due Jan 30 by 11:59pm **Points** 22 **Submitting** a file upload **File Types** txt

Part A - HTML/CSS:

Create two a file called “calculator.html” and “calculator.css” to recreate (as close as possible) to the following picture (copied from coursys – and shown in a browser window):



You may make any reasonable guesses on fonts, colors and measurements. You may add additional HTML elements for the next part, but we will only look for the elements in the previous picture.

Part B – Javascript

Create a file called “calculator.js” that will calculate allow users to choose to produce a calculated "Mean of grade" or “Weighted grades”. The result should be displayed under the “Result" label in the above picture. The Percentage column (the percentage of the corresponding activity) should be updated as the user is typing the into the “Grade” Column. The user may click on the "add activity" button to display more grade items.

Mean of grades

The sum of all grades divided by the total number of grades

A1 70/100, A2 20/80, A3 10/10:

(0.7 + 0.25 + 1.0)/3 = 0.65 -> 65/100

Weighted grades

Each grade item can be given a weight, which is then used in the arithmetic mean aggregation to influence the importance of each item in the overall mean. In simple terms, the category "total" will be equal to the sum of the scores in each grade item, these scores being multiplied by the grade items' weights, and that sum being finally divided by the sum of the weights, as shown in this example.

A1 70/100 weight 10, A2 20/80 weight 5, A3 10/10 weight 3:

(0.7*10 + 0.25*5 + 1.0*3)/18 = 0.625 -> 62.5/100

If inputs are omitted for any of the assignments, you can consider it as 0/0 **OR** perform the calculation without the omitted item. Either way, for usability, you should alert the user.

Hint: You may want to consider Javascript functions

onclick()

getElementsByTagName()

getElementsByName()

Usability and Effort:

A small portion of the grade will be dedicated to usability and effort, this is learnability and like-ability. Existence of bugs, lack of error checking may result in a lower usability score.

Constraints:

In this assignment, you may not use frontend frameworks or libraries such as Angular, React, Bootstrap, or jQuery. You will have many opportunities to use these (if you'd like) in the future :)

Marking Scheme:

Part A –
6 marks: The colors, fonts, and spacing should follow the picture **as much as possible**. For example, a shade of red would be a reasonable guess at the background color of the page; a shade of blue would not.

Part B –
12 marks: The correct values should be calculated and all described functions should be working.
4 marks: Effort, Usability.

Project:
1 marks: Running on Heroku
1 marks: project pushed onto github or gitlab (in addition to the git link provided by Heroku) - We must be able to find your working app at the webapp link, and should be able to clone your project at the git link.

Submission:

Your webpage could either be stored in the "public" folder of your heroku webspace or in a view with an empty controller. As with all subsequent assignments/project iterations, you should submit **a text file** with the following information onto canvas.

1. The URL of your **webapp**, this is where we'll be able to play around with your application. It is your responsibility to ensure that you have the correct URL. If the TAs cannot find it, he/she cannot mark it.
2. Your **git link,** this is where we can view your code, your progress and time of submission.

Some Rubric

Criteria	Ratings		Pts
Adding new form elements and making correct calculations	**6 to >0.0 pts** **Full Marks**	**0 pts** **No Marks**	6 pts
Proper calculation for both buttons	**7 to >0.0 pts** **Full Marks**	**0 pts** **No Marks**	7 pts
Usability	**4 to >0.0 pts** **Full Marks**	**0 pts** **No Marks**	4 pts
Deployment on Heroku and Git link	**3 to >0.0 pts** **Full Marks**	**0 pts** **No Marks**	3 pts
Similar HTML & CSS	**2 pts** **Full Marks**	**0 pts** **No Marks**	2 pts
			Total Points: 22