Implement a counter with count up and count down button. [Link demo](https://bit.ly/c4t-counter).

Steps suggestion:

1. In HTML file, create the UI structure
2. In JS file, create a variable to represent the current count (let count = 0 for example)
3. Register to count up (+) and count down (-) buttons, when users click a button, update the count variable accordingly
4. After update the count variable, update the UI (DOM) for users to see

-------------------

Implement a timer, with time out entered and can be stopped by users. [Link demo](https://bit.ly/c4t-timer).

Step suggestion:

1. Learn how to stop an interval in JS, Google: 'JS clearInverval function'
2. Create UI structure in HTML file
3. In JS, create a variable to represent the current time (in seconds)
4. Assume the time out is always 5 seconds
5. Register start button click event, when users click it, set the current time to 5 seconds, start a timer using setInterval
6. In setInterval callback, reduce current time by 1
7. When the current time is reduced to 0, stop the timer by calling clearInterval
8. Register stop button click event, when users click it. Stop the timer
9. Now go back to step 4, remove the *'time out is 5 seconds'* assumption, when users click on start button, get the time from input and assign it to the current time

---------------------------------

Implement an offline random quote app. [Link demo](https://bit.ly/c4t-quote).

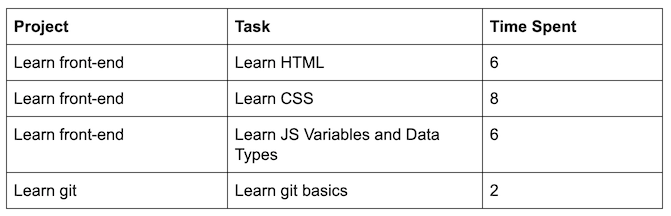
The quote data to randomize from is from [this](https://raw.githubusercontent.com/edtechkidsvn/quotes/master/data.json).

Step suggestion:

1. Create UI structure in HTML file
2. In JS, copy the data from the provided link, paste it to JS code and assign it to a variable / constant
3. Randomly pick a quote from the data
4. Use DOM to show the quote content and author to users
5. Register to reload button's click event. When users click on this button, randomly pick a quote and show it to users again

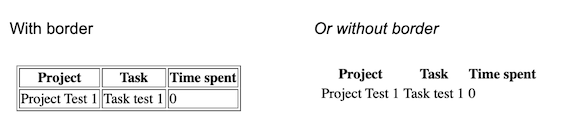
-----------------------------------

Take a look at the following Timesheet

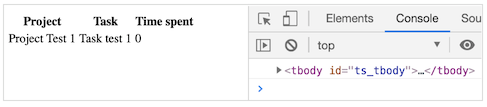


1) In JS, Initialize a variable named timeSheetData to store the data above, log/print it out

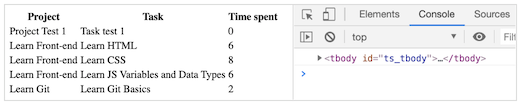
2) In HTML, create a <table> with the same headers as the above timesheet. If you have not worked with table before, watch the *‘HTML - List and Table’* video or see this [quick example](https://codepen.io/mindxvn/pen/vYYyNyW).



3) Get (Read) the table body from DOM



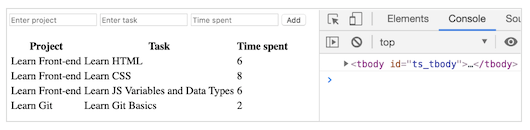
4) Insert ALL of the data in timeSheetData into the table body, each task in a row



The first, placeholder, row now can be removed



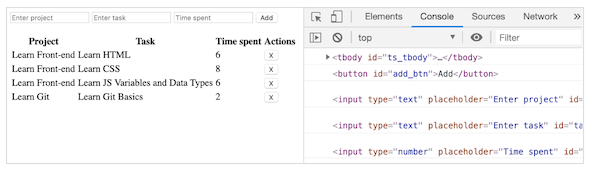
5) Update the UI (HTML) as follows:



6) Perform add new line when ‘Add’ button is clicked (*note: Instead of only inserting new HTML into the table body, you should update timeSheetData first, then use timeSheetData to update the table*)

[Link demo](https://youtu.be/MMZs061ahAE?t=4)

7) Add new columns name Actions, with each line, add a delete button in the newly added column



8) When users click on the delete button, remove the respective line

***Hint:***

Declare removeLine function with on parameter: index of the item to delete

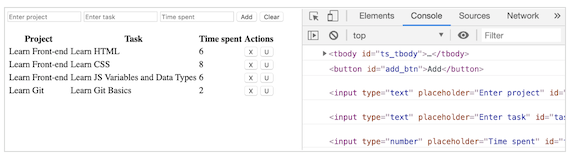
*First solution: Call removeLine inside onClick value of button: onClick=”removeLine(1)”*

*Second solution: Add a custom property to each button, for example:*

*data\_index=”1”. Then use in event callback e.target to get the index of the line in which the button is clicked. Finally, use removeLine()*

[Link demo](https://youtu.be/5N3qaH5FvvM)

9) Add Update, Clear buttons and U button in each line



10) Implement update

[Link demo](https://youtu.be/HTLK8gaNrJ0?t=4)

-----------------------------------

*Note: This exercise is OPTIONAL*

Implement a freaking math game. [Link demo](https://bit.ly/c4t-fmath). *Note: The settings button is not required*

Step suggestions:

1. Create UI structure in HTML file
2. Learn how to create a quiz, save the right answer (this quiz is correct or incorrect) for later, *ask mentor if you get stuck*.
3. Using DOM, show the quiz to users
4. Register the correct (v), incorrect (x) buttons, when users click on one of these button, check with the right answer
5. Learn how to play a sound, google 'JS Play audio' and play the incorrect or correct sound based on user's result
6. Update score based on user's result and use DOM to show it to them then show users a new quiz
7. Use setInterval to start counting down right after a new quiz is shown, when count down to 0, play the incorrect sound, reset the score, and show them to users
8. Add CSS to present your result better