Lab 5 HTML, CSS, and JavaScript

Given: Thursday August 30, 2018
Deadline: Thursday September 6, 2018

What you need to have done within lab hours

1. **Click me**
   Develop the web app that counts and displays the number of clicks on the button “Click me”

Fig1. Clicking the button for 1 time

Fig 2. Clicking the button for 2 times

Fig 3. Clicking the button for 3 times

What can be done after lab hours

Guess the number game

Develop the web app that a simple guess the number type game. It should choose a random number between 1 and 100, then challenge the player to guess the number in 10 turns. After each turn the player should be told if they are right or wrong — and, if they are wrong, whether the guess was too low or too high. It should also tell the player what numbers they previously guessed. The game will end once the player guesses correctly, or once they run out of turns. When the game ends, the player should be given an option to start playing again.

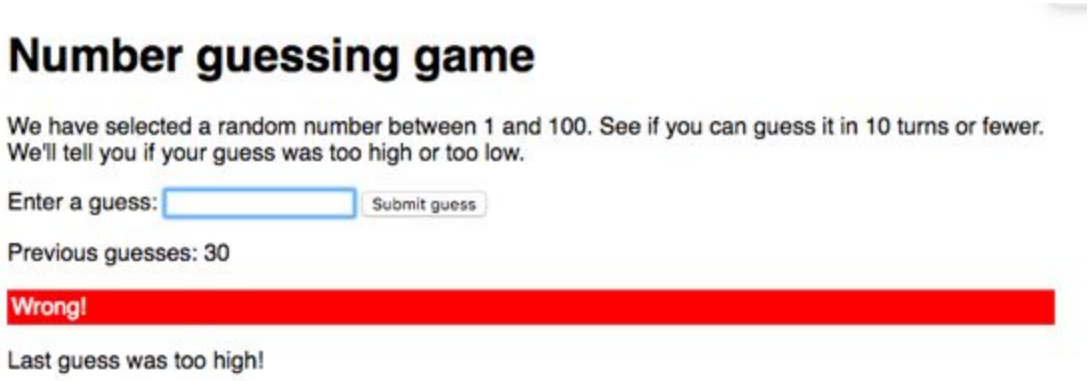

Fig 4. The number guessing name interface

Fig 5. The number guessing game when the user can guess the number correctly

Fig 6. The number guessing game when the user fails to guess the number correctly

Simple Calculator

Develop a web app with the title as "Simple calculator that has the interface and the behavior similar as the Fig. 7 and Fig. 8.

The styles of the web app as specified

1) The font family is "sans-serif"
2) The input that displays the number has the type as text, height as 2em, font size as 12 pt, width 96%, background color as black, text color as white, and text-align as right
3) The input that has the button has the width as 3.5 em, background color as orange, no border, padding as 20 px, text-align as center, font-size 12 pt, margin 2 px, and border radius as 50%. But for the button number '0' has the width as 7em
4) The background-color of the number with the values of red, green, blue as 189, 183, 183

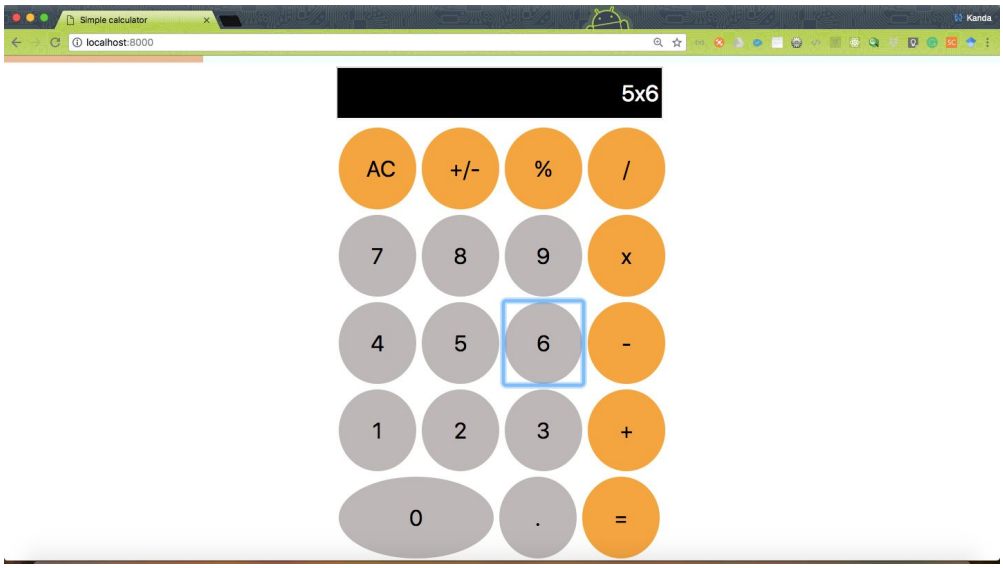

Fig 7. The simple calculator with the input as “5x6”

The behaviors of the simple calculator as specified

1) When the user clicks any button except the button ‘=’, the app displays the number or the symbol of that button.
2) When the user clicks the button ‘=’, the app calculates the previous numbers and symbol.

As shown in Fig. 7, when the user clicks the number “5”, “x”, and “6”, the app displays “5x6” as shown in the input text box. Then when the user clicks the button ‘=’, the app computes the result fo 5x6 as 30 as shown in Fig. 8

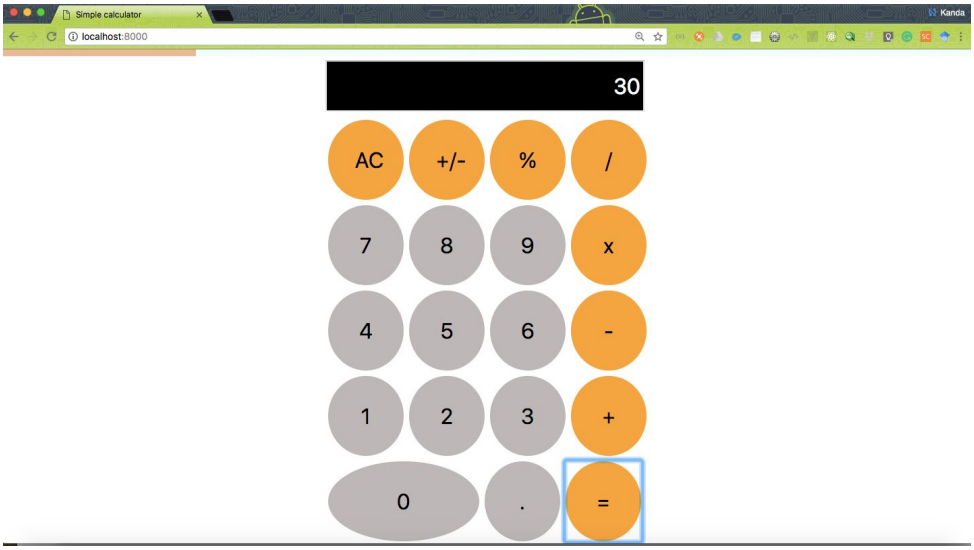

Fig 8. The simple calculator with the output of the pressing buttons “5”, “x”, “6”, and “=”

When the user presses “AC”, the app resets the text display to be an empty string

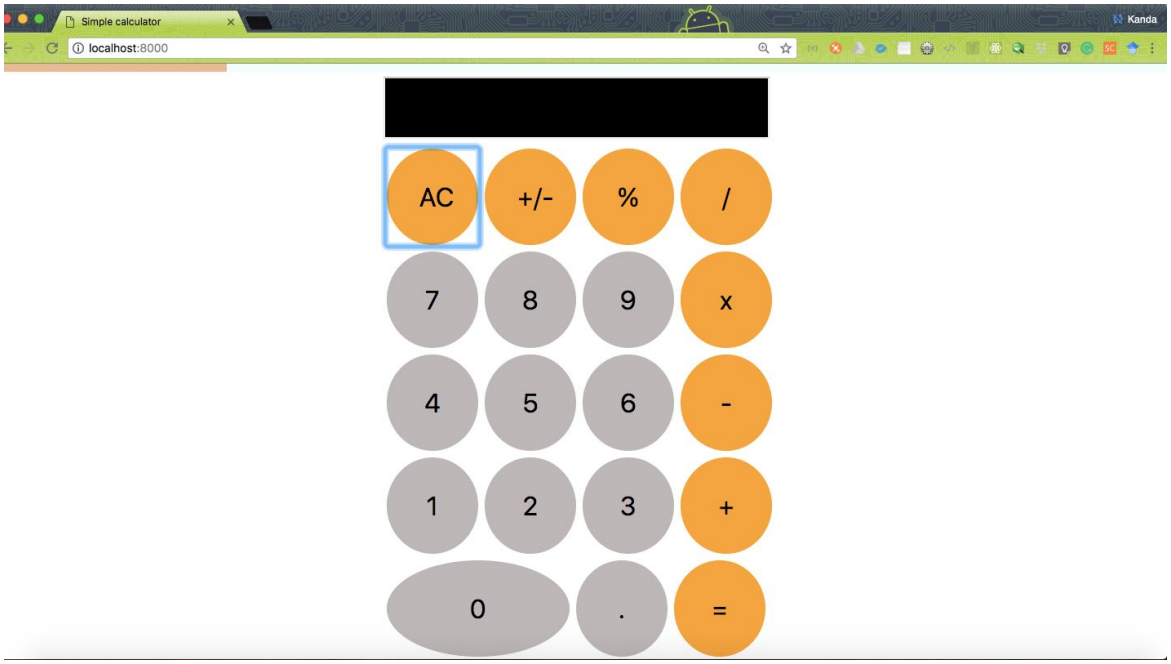

Fig 9. The simple calculator with the display reset after the user presses “AC”

When the user presses “+/-”, the app switches a positive number to a negative number but if the user presses “+/-” to a negative number, the app sets the number to be a positive number. For example, when a user presses “58” and “+/-”, the app displays “-58” as shown in Fig. 10

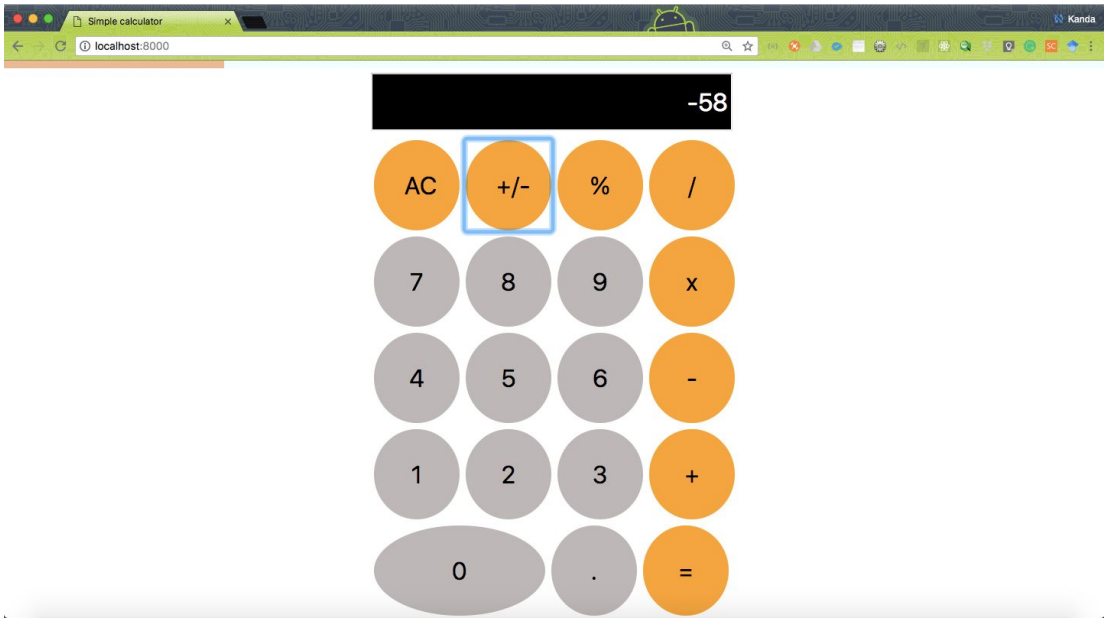

Fig. 10 The simple calculator after the user presses “58” and “+/-”

However, if the user presses “+/-” to the number “-58”, then the app displays the number as “58” as shown in Fig. 11

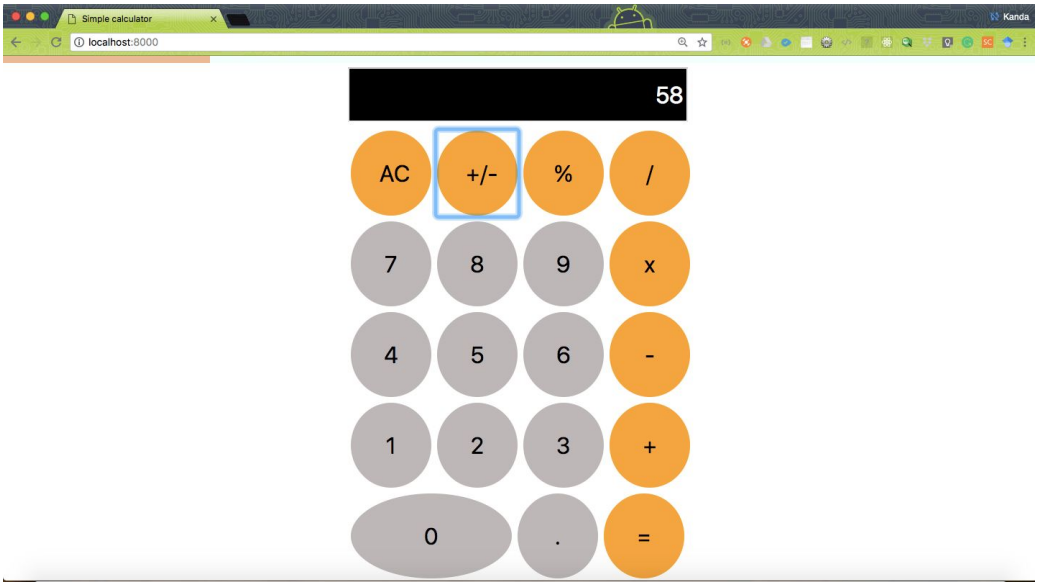

Fig. 10 The simple calculator after the user presses “58”, “+/-”, and “+/-”

For other symbols, “%” stands for the remainder operator, “/” stands for division operator, “x” stands for the multiplication operator, “+” stands for addition operator, and “-” stands for subtraction operator.

You may want to test both “AC”, “+/”, and “%” by pressing the “50”, “+/-”, “%”, “8”, and “=”, you then should get the result as -2 as shown in Fig. 11 and Fig. 12

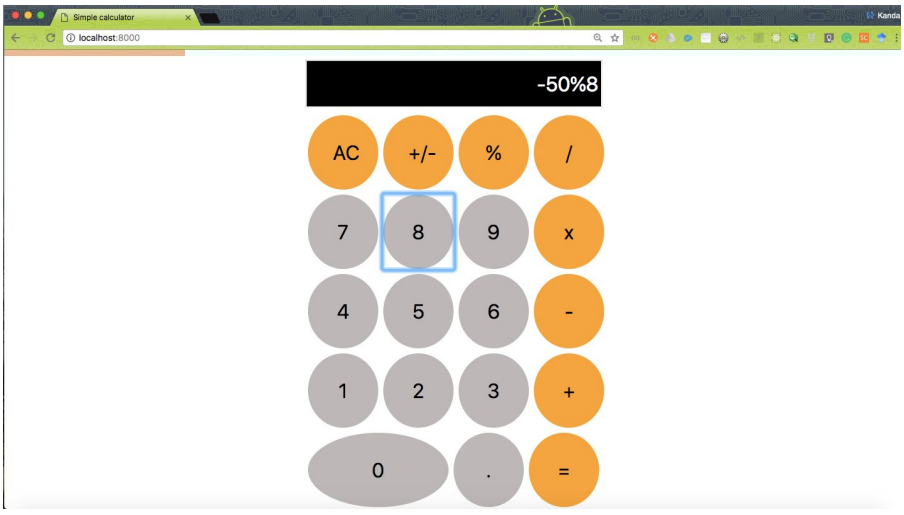

Fig. 11 The simple calculator when the user presses “50”, “-”, “%”, and “8”

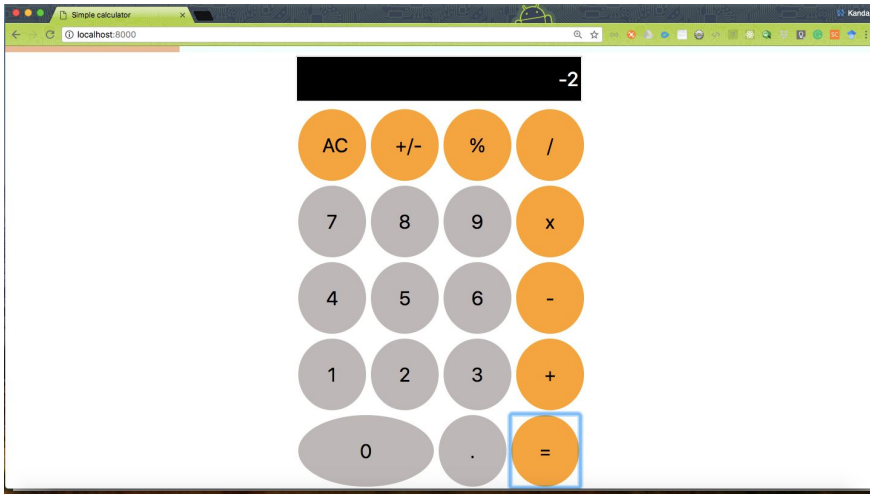

Fig. 12 The simple calculator when the user presses “50”, “-”, “%”, “8”, and “=”