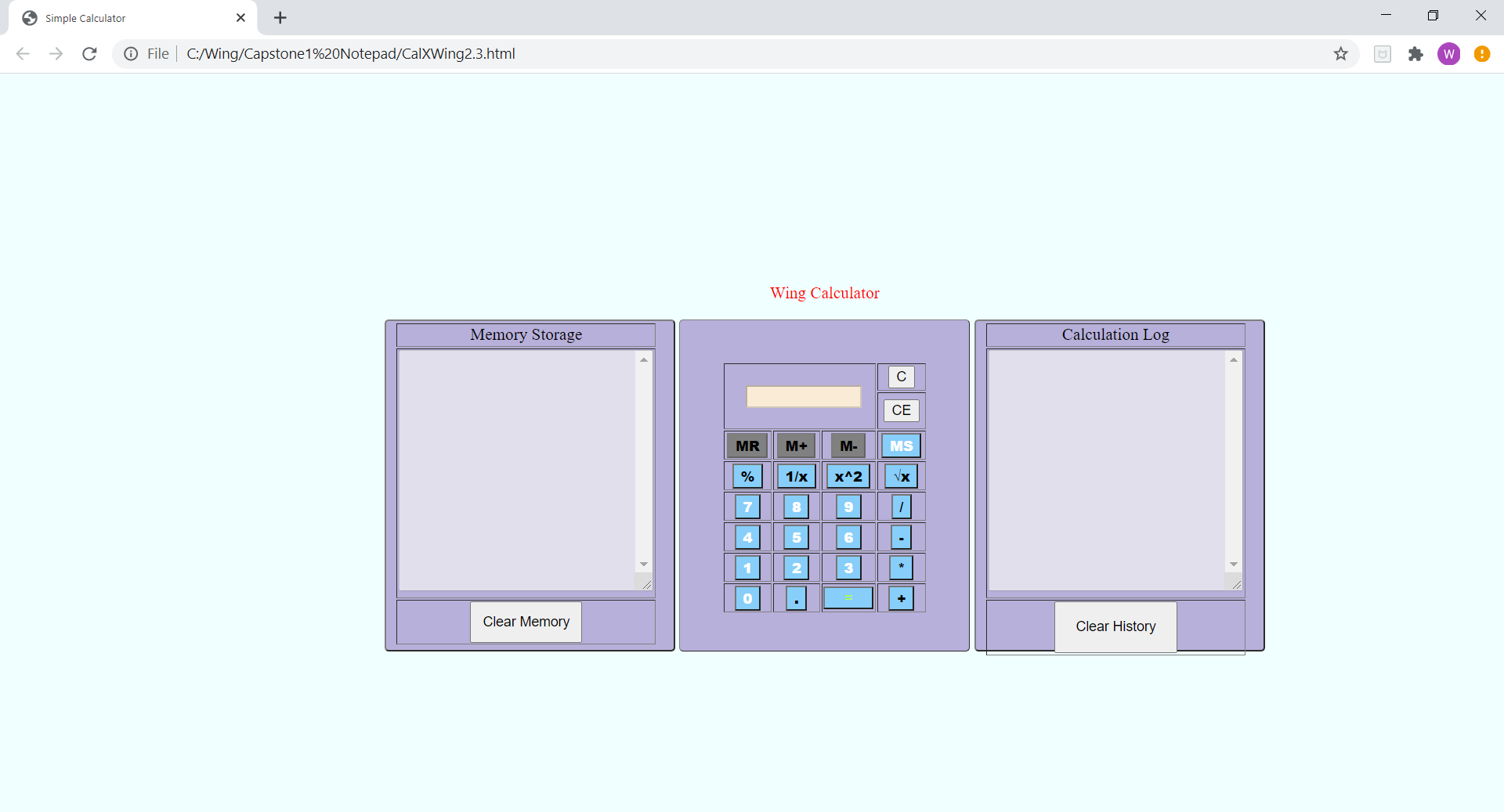
Documentation for Calculator Application (TOH CHUN WING)

1. Requirement:
   1. Develop a display of a “calculator” with the following functional requirements:
      1. There should be an input box for user to enter numbers and string for evaluation.
      2. There should be a “Log” text box to display the string input and its result. The box must be scrollable to see all past history.
      3. There should also be another text box to display the results stacked on a memory space. This box must also be scrollable.
      4. Evaluate a string of numbers with the following operators. The string input and results must be displayed in the Log” box.
         1. ‘+’(addition);
         2. ‘-‘(subtraction);
         3. ‘\*’(multiplication);
         4. ‘/’(division)
      5. Perform the following operations on a single value input. The input and result must be displayed in the Log box.
         1. Square root
         2. Square (number \* number)
         3. Inversion (1 / number)
         4. Percentage (present the percentage of 1 number over another)
      6. It should also have a “memory” function that can perform the following and the result displayed in the memory display text box. It should also display the latest memory function in the Log box.
         1. Store input into a memory (subsequent numbers stored should be stacked , meaning it should not overwrite the previous input stored. The stored number should be displayed
         2. Add input value to the last memory value and display the result as the current value
         3. Subtract input value from the last memory value and display the result as the current value
         4. Recall from memory the last value and display on input box.
      7. Two extra buttons, one to clear the latest input and another to clear the latest input and the Log should be available
      8. There should also be an option to clear the “Log” and “Memory” display boxes.
      9. All inputs should be enabled from the buttons provided in the calculator or the keyboard.
   2. A simple design will look like photo below but the layout and color is optional as long as it performs the function as stated.



1. Design
   1. We will adopt a similar look with our Brand Name at the top of the calculator.
   2. A centralised text box would be used for display of input and results at the top of the calculator
   3. The buttons to clear the input and log/input should be on the right of the input box.
   4. The next row of buttons would be the memory function buttons that will span the breadth of the input and the clear function buttons
   5. Below these memory buttons will be the buttons
      1. Percentage (%) button
      2. Invert button
      3. Square function button and the
      4. Square root function button
   6. The following rows of buttons below would be
      1. numeric numbers 7-9 and operator divide (/)
      2. numeric numbers 4-6 and operator divide (-)
      3. numeric numbers 1-3 and operator divide (\*)
   7. The bottom last row would be numeric numbers 0, decimal (.), result (=) and operator plus (+)
   8. The display box for the memory data would be on the left side of the calculator and having the same size of the calculator. It must be made scrollable.
   9. The display box for the Log data would be on the right side of the calculator and having the same size of the calculator. It must be made scrollable.
   10. As the input is clicked (keyed) on, the number/operator is appended to the text string so far and displayed.
   11. When the result (=) is clicked on, only the calculated result is freshly displayed in the text box. If it is a memory function, the requirement rule above 1(vi) should be followed
2. Development
   1. We will use html for this calculator application with in-line CSS and scripts.
   2. Within the “head” we have added the following <scripts> in order:
      1. Function “dis()” which is used to display the input in the text box and it’s called by the input click on numeric and operator function buttons.
      2. Function “solve() which is used to evaluate the string of numbers/operators input.
         1. Assign the string of input to “x”
         2. Assign the evaluated result to “result”
         3. If the result is infinite (i.e. divide by 0) send windows alert to user (“Cannot divide by zero”) and clear the display using the clr() function
         4. Display the string and results in the “Log” area.
      3. An “addEventListener() group is added to capture keyboard input.
      4. Function clr() which is called by bthe “C” button is used to clear the input text box.
      5. Function clrAll() which is called by the “CE” button and the “Clear History” button which then calls the function clr() to clear input text box and also clears the “Log” history.
      6. Function xSq() is called by the "x^2" button and evaluates the input value by using the method Math.pow(curVal, 2). 2nd attribute is for power of 2.
      7. Function sqRoot() is called by the "√x" button and evaluates the input value by using the method Math.sqrt(curVal). 2nd attribute is for square root of val.
      8. Function invX() is called by the “1/X” button and evaluates the input to 1/x.
      9. Function percent() is called by the “%” button and present the result by multiplying the result by 100 and displaying as result %.
      10. Function mStore is called by “MS” button to store the input value into local storage. At start, this button has the button name “MS” set to white. At the first click (used) the button name “MS” is set to black, the buttons “MR”, “M+”, “M-“ are enabled and the background color set to lightskyblue from previous gray.
      11. Function mRec() is called by the “MR” button to get the last value from the memory (local storage) and display on the input text box.
      12. Function mPlus() is called by the “M+” button and it adds the input value to the last value in the memory and display the results in the memory text area. It also display the equation “x + y = z” in memory” where x is the latest input, y is the last stored value and z is the addition result.
      13. Function mSub() is called by the “M-” button and it subtracts the input value from the last value in the memory and display the results in the memory text area. It also display the equation “y - x = z” in memory” where x is the latest input, y is the last stored value and z is the subtraction result.
      14. Function mClr() is called by the “Clear Memory” button sitting at the bottom of the Memory Storage display text area.
   3. After the Scripts, I coded the CSS <style> as follows:
      1. .box is used to describe the layout and colors of the boxes (calculator, memory and log display)
      2. .title is for the tile
      3. .numButton is used to described buttons that are in the class numButton which is basically for the numeric buttons
      4. .symButton is used to described buttons that are in the class numButton which is basically for the operator buttons
      5. .input is for the input text box
   4. In the body section, we are creating tables for the Memory display text area, the calculator, and the History log text area. The table is created as –is described in the design section.
      1. The numeric buttons are classed as “numButton”, having values same as the number, and calling the relevant function as onclick (as described above).
      2. The operator and symbols buttons are classed in “symButton”, having values same as the number, and calling the relevant function as onclick (as described above).
3. Developer Test Cases for verification of the requirements are done as follows and passed.
   1. String of numbers with mixed operators (e.g. 3+4\*12 = 51)
   2. 6 MS (first MS changes the other memory buttons) and stores value in memory
   3. Subsequent MS stack into memory
   4. Square, square root, percentage and invert tested correctly (e.g. ¾ percentage is 75%, inversion of 5 is 1/5=0.2, square root of 81=9, square 5 = 25)
   5. C and CE buttons worked according to specs.
   6. M+ adds latest value to memory and result is displayed on memory and the equation string is display on the log
   7. M- works as well as the M+
   8. MR recalled the last value stored in the memory.
4. Future Enhancement
   1. Beautify the buttons and lay out
   2. Include more “scientific” calculation like Sine, Cosine, Tangent