Full Stack Development Home Assignment

The following coding exercise consists of several questions. The questions become more advanced as you progress through the exercise. You may choose which questions to answer and are allowed to skip questions.

Note:

1. Please write the code in C# or .NET Core.
2. Use Angular X for the GUI.
3. Please use SOLID design principles.
4. You can use external libraries, but you do not have to.
5. Create Unit tests for all assignment steps below.
6. Create at least 3 Unit tests for your BL (beside assignment steps).
7. This is not a Boolean true/false test. The test does not check if the program is running or if the result is correct. Instead, the test result considers a multitude of coding aspects, such as callable units, correct separation of components (in a client-server model), easy to maintain code and the difficulty level of the questions.
8. The weight breakdown for the exam score is as such:
    - Server side: 50%
    - Client side: 40%
    - Testing: 10%
9. After submitting your answers, you will be invited to an interview in which you will be asked to explain the code.

The goal: To write a program that receives an arithmetic expression (calculator) as the user input and outputs the result.

- The calculation must be done on the server side.
- Use only your custom calculation logic.
- Write on external txt file the principals and reasons behind your logic calculator implementation and make sure the text file is included in the project zip file.
- Please note that we won't be able to accept answers using existing .NET mathematical evaluators (i.e. DataTable) or online/3rd-party solutions.

Step 1:
Develop a home page with your name and email address in the middle of the page.
Then, add a button that when clicked on opens a webpage with the calculator user interface.

****Keep a history of all the calculations and their results and display that calculations history next to the calculator.**

Step 2:
The program receives an arithmetic expression consisting of two single digits and one of the four basic arithmetic operations. You can assume the input is valid.

Examples:

Input	Output
1+2	3
8-2	6
8+5	13

Step 3:
Add support for integers.

Examples:

Input	Output
12+45	57
123+35	178

Step 4:
Calculate the result of a series of arithmetic operations according to the order of operations.

Examples:

Input	Output
2+2*2	6
12+5*7/4-2	18.75

Step 5:
Add support for decimal fractions.

Examples:

Input	Output
3.5+12.3	15.8
0.5*8	4

Good luck!