Random Math Game

The purpose of the project is to create a math game that would review elementary level arithmetic(addition, subtraction, multiplication, and possibly division). The design of the game needs to be very user friendly so it's obvious what buttons to push and what needs to be entered to go from problem to problem. Each new problem should be initiated with the push of a button. (Not one button per problem.) Each problem should be a combination of random numbers. The user should be notified if their answer is correct or not and their overall score should be kept. In order to add complexity, the game could have things like levels, modes, multiplayer options, badges, high score…

Each level needs to be completed and turned in before the next level is done.

C-Level

- Create a Start Button.
    - When clicked it should show buttons that represent an Easy level (later there will be more buttons)
    - It should also show addition and subtraction buttons.
- Easy/Medium Buttons
    - When these buttons are pressed variables will be set for use in other functions. These variables should be global
        - For example level = "easy"
- Addition and Subtraction Buttons
    - When these buttons are pressed variables will be set for use in other functions. These variables should be global
        - For example operation = "plus"
    - A button should then appear that the user will click on to ask the actual questions.
    -
- Question button
    - If you would like to use prompts and alerts
        - Random numbers should be chosen in the range associated with the level. For example
            - If(level=="easy")
                - Num1 = Math.random…..
            - Else if (level = "hard")
                - Num1 = Math.random…..
            - It is easier to use the random number function that was made in class and is shown in the examples.
        - A math question will appear in a prompt. For example:
            - If (operation = "plus)
                - Guess = prompt("What is " + num1 + " + " + num2 + "?")

    - If you would like to use text boxes (required later) The only difference would be that you would either:
        - Need a separate button to ask the question and a separate button to check your answer
        - Or
        - Need a button to ask the question and check the answer when you press the enter key
    - The user should see if they are correct or incorrect (paragraph is better than alerts.)
    - The total correct questions should be displayed somewhere in a nice way.
- Badges
    - After the user gets 5 addition questions correct a badge (image) should appear that indicates they completed the level. (Don't worry if your picture isn't that nice. It could just be a picture of a plus sign.)
    - Another badge should be earned after subtraction.
    - To do this have the pictures there already and simply hide them using css. Then using a condition show the image. For example
        - If(countCorrect ==5){
            - Doc…getEle….("plus").style.visibility = "visible"

B-Level : ALL aspects of C level should work in with these adds.

- Levels
    - Add a multiplication level
    - Add the ability to go beyond the 5 questions. For instance, after 5 addition questions, the random numbers will have a higher range to make it more difficult.
- End of Game/Level
    - Each Level should have an ending point. This occurs after the mastery badge for that level is reached. After this, that level will not be available anymore.
    - After all 3 mastery badges are earned the game will end.
- Badges
    - The badge should change after each set of 5 questions. Make it look a bit fancier until after 15 questions they get the mastery badge for that level.

A – Level - Ideas

- Levels
    - A new level will be available that mixes all 3 operations. Again, the difficulty level should increase and badges should be given.
- Badges
    - At least 2 special badges should be awarded. One of them should be for being perfect through each level.
    - The other can be something creative
- Other

    - Ask for the user's name and display the high score for each level with that players name.
    - No prompts or alerts are allowed at the A level
- 2 Player mode
    - In this mode there are 2 players that are competing. There should be something that tells the players when to go.
    - A random player should go first. If that player misses a question the computer tells the other player to go. You can use any combination of levels that you wish to make this 2 player mode work, or you can make a new level just for 2 player mode. Try not to just copy and paste code. Think of abstraction
    - At the end the winning player should be known. How a winner is determined is up to you. Again try to be creative on how the winner is determined. This helps to decide between an A-, A, or A+.
- CSS and User Friendly
    - At the A level there should be some basic CSS that makes the screen looks good.
    - The program should also be very playable. This means that buttons are labeled nicely, show up only when I should be able to click on them. Use visible or enabled.
    - The code itself should have good names. This means that variables are named things that tell me what the variable is for. Functions have names that describe what it does. Etc…