Capstone Project: Raid Static Manager

Code Quality Review

General

- Does the code work? Does it perform its intended function, the logic is correct etc.

  Yes, all function, and logic works well.

- Is all the code easily understood?

  Yes

- Does it conform to your agreed coding conventions? These will usually cover location of braces, variable and function names, line length, indentations, formatting, and comments.

  Yes, all variable names and function names even file names all understandable.

- Is there any redundant or duplicate code?

  Yes, there are some duplicate code, that caused by open sources in .CSS file.

- Is the code as modular as possible?

  Yes, all different functions are separated by different form.

- Can any global variables be replaced?

  Yes, they didn't affect each other.

- Is there any commented out code?

  No

- Do loops have a set length and correct termination conditions?

  Yes,

- Do the names used in the program convey intent?

  Yes

Performance

- Are there any obvious optimizations that will improve performance?

  Yes, UI design code.

- Can any of the code be replaced with library or built-in functions?

  Not sure

- Can any logging or debugging code be removed?

  Yes, some code can be removed without changing functionality.

Security

- Are all data inputs checked (for the correct type, length, format, and range) and encoded?

  NO, some of low priority data are allowed bad input.

- Where third-party utilities are used, are returning errors being caught?

  Yes.

- Are output values checked and encoded?

  No

- Are invalid parameter values handled?

  Not all the invalid parameter.

Documentation

- Do comments exist and describe the intent of the code?

  Yes

- Are all functions commented?

  Not all

- Is any unusual behavior or edge-case handling described?

  No

- Is the use and function of third-party libraries documented?

  Yes

- Are data structures and units of measurement explained?

  Yes

- Is there any incomplete code? If so, should it be removed or flagged with a suitable marker like 'TODO'?

No

Testing

- Is the code testable? The code should be structured so that it doesn't add too many or hide dependencies, is unable to initialize objects, test frameworks can use methods etc.

  Yes, all code works well.

- Do tests exist, and are they comprehensive?

  Yes

- Do unit tests actually test that the code is performing the intended functionality?

  Yes