**Functionality**: A function is a block of code that does a specific task. Determine whether the code functionality does what the code was designed to do.

**Design**: Check for the following:

* Are there code files such as images, scripts, and user interfaces and are they systematically organized?
* Does the code have classes and are they properly identified?
* Is the code logic divided among classes and do the classes interact with each other? Is the code logic separated from user interface code?

**Repeated Codes**: are codes being repeated throughout the program? Repeated code should be used as a single function for a project and as a sub-module if repeated across different projects. For repeated functionality in multiple classes, inheritance could be used to reduce the number of functions.

**Run Time of Algorithm**: Check for the following:

1. Are there any nested loops? Can they be replaced with simpler functions?

2. Are any calculations being repeated? Could the code use memoization to reduce run time?

3. Are the data being accessed properly to reduce lookup cost?

4. Are there any irrelevant operations that could be dropped from the code?

**Hardcode**: Hard coding is the software development practice of embedding output or configuration data directly into the source code of a program. Hardcoding in software development should be avoided and constants and descriptive variables names should be used instead.

Check if:

* Are constants specific to a class being used?
* Are configurable constants part of configuration class?
* Are constants that are being shared between classes configurable or not configurable, and are the constants being used through a constant file?

**Error and Warning Handling:** Does the code have error handling included in it? Exceptions should handle invalid inputs, hardware constraints, and network issues among others.  Assertions should validate conditions to assist in detecting bugs and validate test cases.

**Comments**: comment is needed when it is unclear why a portion of a code exists. Comments enhance the readability of the code and help others to understand the code. Check to confirm if:

* Code is properly commented and the comments are concise and actionable,
* Specific categories of comments such as inline comment, block comment and documentation strings that are being used in the code.

**Consistency**: The programming style should be consistent throughout the code in terms of naming, spacing, and brackets. Check if naming of function, variable and class are consistent throughout the code.

**Safe Parallel Programming**: Check if the code contains parallel programming and confirm if there is any deadlocks and race conditioning.