Coding Standards Adherence

- Is the code consistent with the project's coding standards?
- Are naming conventions followed (variables, functions, classes)?
- Are there any hard-coded values that should be constants?

Readability

- Is the code easy to read and understand?
- Are comments used effectively to explain complex logic?
- Is the code appropriately indented and formatted?

Potential Bugs

- Are there any obvious logic errors in the code?
- Are all possible error cases handled (e.g., null checks, exceptions)?
- Are there any potential security vulnerabilities?

Performance Improvements

- Are there any inefficient algorithms or data structures that could be optimized?
- Are there unnecessary computations or memory allocations?
- Is there any redundant code that can be removed?

Functionality

- Does the code meet the functional requirements/specifications?
- Are all the edge cases and boundary conditions handled?
- Are input and output validations performed correctly?

Testing

- Are there corresponding unit tests for the new code?
- Do the tests cover all possible edge cases and error conditions?
- Are the tests passing and integrated into the CI pipeline?

Modularity and Reusability

- Is the code modular and encapsulated properly?
- Are there any opportunities for reusing existing code or libraries?
- Are functions/methods/classes single-purpose and small?

Maintainability

- Is the code structured in a way that future changes will be easy to implement?
- Are there any parts of the code that are overly complex and need refactoring?

- Is the codebase free from commented-out code and TODOs?

Documentation

- Is the code adequately documented with comments and docstrings?
- Are the functions/methods/classes documented with their purpose, parameters, and return values?
- Are there any external documents (e.g., design docs) that need updating?

Security

- Is sensitive data handled securely (e.g., encryption, proper storage)?
- Are authentication and authorization mechanisms correctly implemented?
- Are there any known security issues (e.g., SQL injection, XSS) in the code?

Compatibility

- Is the code compatible with the target environment and platform?
- Are there any dependencies that need to be updated or checked?
- Does the code integrate well with existing modules and libraries?

Compliance

- Does the code comply with relevant laws, regulations, and standards?
- Are there any licensing issues with third-party libraries?
- Are there any GDPR or other data protection compliance considerations?

Scalability

- Is the code designed to handle increased load and scale appropriately?
- Are there any bottlenecks that could affect scalability?
- Are there strategies in place for scaling (e.g., load balancing, caching)?

Error Handling and Logging

- Are errors handled gracefully with appropriate error messages?
- Is logging implemented correctly and at the appropriate levels (info, debug, error)?
- Are there any uncaught exceptions or missing error handling?

User Experience

- Is the user interface intuitive and user-friendly?
- Are there any usability issues that need to be addressed?
- Is the user feedback mechanism (e.g., notifications, alerts) implemented correctly?