

Definitions of Done

Code Completeness: The task is considered done when all the functions have been coded completely and are working as expected without throwing any errors.

Integration Testing: The new features must work correctly and seamlessly with the existing system.

Code Review: The code should be reviewed by at least one peer developer. Only after the code review, and necessary changes have been made and approved, the task can be marked as done.

Documentation: Proper documentation including function descriptions, variable descriptions, and comments for complex code snippets should be written. Without proper documentation, a task cannot be considered done.

Performance Check: The performance of the new features should be checked. If the new feature is making the system slower or if it's not scalable, then the task cannot be considered done.

Style Guides

Naming Conventions: Follow the naming conventions as per PEP 8. For example, use snake_case for variables and function names, PascalCase for class names.

Indentation: Use 4 spaces for indentation as recommended by PEP 8.

Line Length: Try to keep the line length under 79 characters, as recommended by PEP 8.

Comments: Use comments to explain complex code logic. Do not write unnecessary comments for simple and obvious code snippets.

Imports: Always import packages in a new line. Do not import multiple packages in one line.

Whitespace: Do not leave trailing whitespaces at the end of the lines. Also, avoid unnecessary whitespaces in expressions and statements.

Error Handling: Always handle possible exceptions in the code. Use custom exceptions if necessary.

Code Consistency: The style of the code should be consistent throughout.