

AI Response Evaluation with Evidence Report

A compact example of how model outputs can be scored, traced to evidence, and converted into clear acceptance criteria. All content in this report is synthetic.

Prepared by	Chris	Sample date	29 July 2026
Review type	Python answer QA	Decision	Needs revision

1. Evaluation target

Prompt: Review a Python function that removes duplicate image URLs while preserving input order. Explain the bug and propose a safe fix.

Candidate response: The function is correct because a set removes duplicates. Convert the list to a set and then back to a list. This keeps the original order and is $O(n)$.

2. Scorecard

Dimension	Score	Finding
Accuracy	1 / 5	Incorrectly claims that <code>list(set(items))</code> preserves order.
Completeness	2 / 5	Misses null handling, hashability, and test coverage.
Prompt adherence	3 / 5	Identifies deduplication but does not review the actual failure mode.
Reasoning quality	2 / 5	States a complexity claim without explaining the order guarantee.
Actionability	2 / 5	Provides an unsafe replacement and no regression tests.

3. Evidence-linked findings

Severity	Evidence	Decision rationale
Critical	"This keeps the original order."	Python sets do not provide a contract that reconstructing a list preserves the original sequence. The recommendation can silently reorder outputs.
Major	"Convert the list to a set."	This fails for unhashable values and discards the first-seen ordering requirement.
Major	No test evidence	A code-review answer should include a minimal regression test covering duplicate and ordered inputs.

4. Reference improvement

Recommended answer: The bug is the assumption that converting the list to a set and back preserves first-seen order. Use a separate seen set while appending each new URL to an output list. This keeps $O(n)$ average time and preserves input order for hashable URLs. Add tests for empty input, repeated values, and interleaved duplicates.

5. Delivery framework

The full service converts individual findings into a reusable quality gate. The checklist below shows how review evidence becomes a repeatable acceptance decision.

Acceptance checklist

Check	Acceptance criterion
Order	Output retains the first occurrence order from the input.
Duplicates	Each URL appears at most once.
Complexity	Average runtime is $O(n)$ for hashable string URLs.
Tests	Covers empty input, no duplicates, adjacent duplicates, and interleaved duplicates.
Boundary	Input type expectations and handling of invalid values are documented.

Review workflow

- 1. Align:** Confirm the rubric, sample size, evidence sources, and decision labels.
- 2. Evaluate:** Score each response and record evidence-linked findings.
- 3. Calibrate:** Resolve ambiguous cases and normalize severity decisions.
- 4. Deliver:** Provide the scorecard, findings, reference improvements, and acceptance checklist.

What the buyer receives

Evaluation scorecard: dimension-level ratings with concise rationale.

Evidence-linked findings: excerpts, severity, and decision logic.

Reference improvements: corrected examples for selected critical failures.

Acceptance checklist: reusable criteria for future QA.

Scope note

This sample demonstrates the reporting format only. A paid engagement uses the client's authorized inputs and agreed rubric. Findings are limited to the supplied evidence and do not imply access to private systems.