

Take-Home Assignment Review

Take-Home Review

Universal

Structured review of take-home submissions: code quality, testing, and documentation.

Evaluation Criteria

Requirements Completeness

Evaluates whether the candidate addressed all stated requirements and handled edge cases.

Rating: 1 2 3 4 5

Sample Questions:

- Walk me through how your solution addresses each requirement.
- What edge cases did you consider, and how did you handle them?
- Were there any requirements you intentionally skipped, and why?

✓ STRONG SIGNAL

All core requirements are met with attention to edge cases. Candidate prioritized effectively if time-constrained and documented what they would add with more time.

✗ WEAK SIGNAL

Core requirements are missing, obvious edge cases are ignored, or the submission feels rushed with no prioritization of what matters most.

Code Quality & Organization

Assesses the readability, structure, and maintainability of the submitted code.

Rating: 1 2 3 4 5

Sample Questions:

- Why did you organize the code this way?
- How would a new team member navigate this codebase?
- What naming conventions did you follow?

✓ STRONG SIGNAL

Code is well-organized with clear separation of concerns, meaningful naming, consistent style, and appropriate abstraction. It reads like production code, not a hackathon entry.

✗ WEAK SIGNAL

Code is a single file with no structure, variable names are cryptic, there's excessive duplication, or the abstraction level is inappropriate (either God objects or over-engineered).

Architecture & Design Decisions

Evaluates the thoughtfulness of architectural choices and the candidate's ability to justify them.

Rating: 1 2 3 4 5

Sample Questions:

- Why did you choose this architecture/pattern?
- What alternatives did you consider?
- How would this design scale if the requirements grew?

✓ **STRONG SIGNAL**

Candidate makes appropriate architectural choices for the problem size, avoids over-engineering, and can articulate trade-offs. The design allows for reasonable extension.

✗ **WEAK SIGNAL**

Architecture is either over-engineered for the scope (enterprise patterns for a CRUD app) or has no structure at all. Candidate cannot explain why they made the choices they did.

Testing Strategy

Evaluates the quality and coverage of tests submitted with the assignment.

Rating: 1 2 3 4 5

Sample Questions:

- What was your testing strategy for this project?
- How did you decide what to test?
- Are there areas you'd add more tests to with more time?

✓ **STRONG SIGNAL**

Meaningful tests cover the critical paths and edge cases, not just happy paths. Tests are well-organized, readable, and the testing strategy matches the problem complexity.

✗ **WEAK SIGNAL**

No tests at all, tests only cover trivial cases, or tests are tightly coupled to implementation details and would break with any refactoring.

Documentation & Setup

Assesses whether the project is easy to set up, run, and understand through documentation.

Rating: 1 2 3 4 5

Sample Questions:

- Walk me through how to run your project from scratch.
- What assumptions or trade-offs did you document?
- Is there anything I need to know before reviewing the code?

✓ **STRONG SIGNAL**

README includes clear setup instructions, documents assumptions and trade-offs, and explains design decisions. The project runs successfully following the instructions.

✗ **WEAK SIGNAL**

No README, setup instructions are missing or broken, or there's no documentation of the decisions made during implementation.

Red Flags

- Submission appears to be generated entirely by AI with no evidence of understanding
- Code does not compile or run despite claiming it does in the README
- Plagiarized from an existing open-source solution without attribution

Notes & Overall Recommendation

Strong Hire Hire No Hire Strong No Hire

Notes: _____

