

Homework

Three tasks to turn one-off code review into a repeatable AI-assisted audit workflow.

Finish the lab first.

Task 1 — False-Positive Calibration

Objective

Measure the AI's accuracy.

Instructions

1. Re-run your best audit prompt on one of the four lab files.
2. Manually verify each reported finding. Label it: TRUE POSITIVE / FALSE POSITIVE / UNCLEAR.
3. Compute $\text{precision} = \text{TP} / (\text{TP} + \text{FP})$. Compute the count of findings the AI MISSED that you caught manually.
4. Write a one-paragraph note on the FP pattern — what class of code confuses the AI most?

Deliverable

A labelled table and the precision/miss numbers.

Task 2 — Patch & Verify

Objective

Close the loop: take an AI-identified bug, patch it, and validate the patch.

Instructions

1. Pick the highest-severity finding from any lab file.
2. Write a unified diff patch that fixes the root cause (not just the symptom).
3. Ask the AI to re-audit the patched file. Does it still flag the finding? Does the patch introduce new issues?

Deliverable

Original file, patch, post-patch audit output.

Task 3 — Semgrep Rule Authoring

Objective

Codify one finding as a reusable static-analysis rule.

Instructions

1. Pick one vulnerability pattern from the lab (e.g., unsafe `eval()`, hard-coded secret).
2. Write a Semgrep rule YAML that would detect this pattern in any codebase.
3. Test it mentally against the lab file: would it match?

Deliverable

A `rule.yml` file plus a one-paragraph rationale.