Refactor

Secret goal:

Make the code clearer and easier to change without altering what it does.

Quietly steer discussion toward tidying first — say “If we clean this up now, changes will be faster later.”

Actionable objectives:

- Rename at least one unclear variable or function to reveal intent.
- Reorder or split logic so steps read more naturally.
- Remove one duplication or unnecessary element.
- Confirm the program still behaves exactly the same.
- Leave the code looking noticeably more structured or readable.

New Feature

Secret goal:

Make the code do something new and useful while keeping its old behaviour intact.

Gently justify changes with remarks like “Users will need this soon.”

Actionable objectives:

•Add a new input, option, or visible behaviour.

•Extend logic with an extra case or capability.

•Keep all previous functionality still working.

•Integrate your addition cleanly into existing structure.

•Ensure the result clearly demonstrates new value.

Fix

Secret goal:

Make the code stop doing something wrong or unsafe without breaking anything else.

Calmly insist on stability first — say “Let’s make it safe before we add more.”

Actionable objectives:

•Identify a real defect or risky behaviour.

•Add a check, guard, or correction that prevents it.

•Verify all previously correct cases still pass.

•Simplify any obviously wrong or inconsistent line.

•Show evidence that the issue is no longer reproducible.

Test

Secret goal:

Make the code provably correct and trustworthy.

Encourage others to slow down by asking “How do we know this works?”

Actionable objectives:

•Write or describe at least two example calls with expected results.

•Include at least one edge or unusual input.

•Run or reason through each example to confirm outputs.

•Flag any part of the function that still lacks coverage.

•Keep examples valid as others change the code.