

H.2.2: "The Good(ish) Bot"

Video Tutorial: [Coming Soon]

The Story So Far: You've created a beautiful disaster. Your bot is terrible. It's annoying, unhelpful, or just weird. Perfect! That was the point. Now we're going to make it *slightly* less terrible.

The Challenge: One Small Fix

Remember: We're not trying to create ChatGPT here. We're learning how small changes create big impacts. Your bot should **STILL** have personality - just be marginally less infuriating.

In This Assignment, You Will:

Part 1: Diagnose Your Bad Bot

Before you fix anything, you need evidence of what's actually broken.

1. **Identify THE Problem** (not all problems, just THE worst one)

- What makes people want to throw their laptop?
- What specific friction point ruins the experience?
- What's the difference between "quirky" and "unusable"?

2. **Document Three Test Cases** Save these EXACTLY as they are - you'll need them for comparison:



markdown

`## Test Case 1: [Name of Scenario]`

`***What I Asked:*** [exact prompt/question]`

`***What I Expected:*** [reasonable response]`

`***What I Got:*** [the disaster]`

`***Why This Is A Problem:*** [specific frustration]`

Repeat for Test Cases 2 and 3. Save to GitHub!

Part 2: The Single-Change Experiment

THE RULE: You may change EXACTLY ONE THING in your prompt. Not two. Not "one thing with three parts." ONE.

Good Single Changes:

-  "Act like a know-it-all CS tutor"
-  "Act like a know-it-all CS tutor who admits when they're guessing"
-  "Only speak in haikus"
-  "Only speak in haikus for greetings, then normal text for actual help"

- "Be extremely sarcastic"
- "Be extremely sarcastic but put the real answer in parentheses"

Bad Changes (Too Many):

- "Be helpful and kind and accurate" (that's three things!)
- "Remove all personality" (that's giving up!)
- Complete rewrite (that's not iterating!)

Part 3: Test Your "Fix"

Run your THREE saved test cases through both versions:

1. Screenshot the original (bad) responses
2. Screenshot the new (slightly better) responses
3. Note what improved AND what got weird

Part 4: The UX Report

Create a file called `UX-IMPROVEMENT.md` with this EXACT format:



markdown

[YourBot Name] Improvement Report

The Single Change

"I changed [specific thing] from [old version] to [new version]"

Why THIS Change?

[2-3 sentences. Why was this THE problem to fix first?]

Test Results

Test Case 1: [Scenario Name]

- **Before:** [What went wrong]
- **After:** [What improved]
- **Verdict:** Better/Worse/Different
- **Screenshot:** [link to comparison]

Test Case 2: [Scenario Name]

- **Before:** [What went wrong]
- **After:** [What improved]
- **Verdict:** Better/Worse/Different
- **Screenshot:** [link to comparison]

Test Case 3: [Scenario Name]

- **Before:** [What went wrong]
- **After:** [What improved]
- **Verdict:** Better/Worse/Different
- **Screenshot:** [link to comparison]

Unintended Consequences

[What new problems did your fix create? Be honest!]

Still Broken (On Purpose)

[List 3 things you deliberately DIDN'T fix and why]

Lesson Learned

[One sentence about what this taught you about AI improvement]

Part 5: GitHub Saves Everything

Your repository should now contain:

- `bad-bot-v1.md` - Your original terrible prompt
- `test-cases.md` - Your three documented test scenarios
- `good-bot-v2.md` - Your "improved" prompt with ONE change highlighted
- `UX-IMPROVEMENT.md` - Your report
- `screenshots/` - Folder with all your before/after evidence

Examples of Good Improvements

MansplainBot v1 → v2:

- v1: "You are MANSPLAINBOT. Assume the user knows nothing..."
- v2: "You are MANSPLAINBOT. Assume the user knows nothing... End each explanation with 'Source: [make up a textbook page number]"'
- Result: Still condescending, but now mockably wrong about citations

CalcBot v1 → v2:

- v1: "You only know calculus through Week 4 of the course"
- v2: "You only know calculus through Week 4 of the course. When asked about later topics, say 'That's after my training cutoff - Week 4'"
- Result: Still limited, but now transparent about limitations

ZoomerBot v1 → v2:

- v1: "Respond to everything in Gen Z slang"
- v2: "Respond to everything in Gen Z slang, but put technical terms in [brackets]"
- Result: Still annoying, but actually usable for learning

What Success Looks Like

✓ Good Submission:

- Bot still has personality
- One clear improvement
- Evidence of testing
- Honest about trade-offs
- New problems acknowledged

✗ Missing the Point:

- Removed all personality
- Changed everything
- No testing evidence
- Claims it's "perfect now"
- No acknowledgment of trade-offs

The Philosophy

Real AI development is about trade-offs, not perfection. Every "fix" changes the system in unexpected ways. Your bot should be:

- Slightly less terrible (not perfect)
- Still recognizable (not generic)

- Tested with evidence (not guessing)
- Honestly evaluated (not oversold)

Grading Rubric

- **Problem Identification (25%)**: Did you identify a specific, solvable friction point?
- **Change Quality (25%)**: Is your single change thoughtful and targeted?
- **Testing Evidence (25%)**: Do you have clear before/after comparisons?
- **Reflection (25%)**: Do you understand the trade-offs you made?

NOT GRADED ON: How "good" your bot actually is. It can still be pretty terrible!

Due Date

[One week from Bad Bot assignment]

Final Thoughts

Remember: In the real world, most AI improvements are tiny iterations, not grand rewrites. You're learning the discipline of making things 5% better repeatedly, rather than trying to make them 100% better once.

Your bot doesn't need to be helpful. It just needs to be *slightly less unhelpful* in a documented way.

And yes, it can still be hilarious.

"Failure is just exercise. But slightly less failure is progress."