Hussein Bourgi

Maids.cc

Systematic Debugging

# 1 – Solution:

1. **Start by looking at what's changed**:

* Since the issue popped up right after we updated the discount logic, that’s my first suspect. I’d check the new code or files that handle the discount calculation.

1. **Recreate the problem**:

* I’d try applying the SAVE20 code under different conditions—like different user types (member vs. non-member), different order totals, etc.—in a test environment. This helps me figure out exactly when it goes wrong (10% discount, 40%, or no discount) and when it works fine.

1. **Dig into the details**:

* Add some printouts/logs to see what's actually happening inside the discount function. I’d check things like:
* What percentage is the code setting for SAVE20?
* Is the membership logic overriding it?
* Are there any unexpected rounding or calculation steps?
* If I can, I’d run the debugger and watch the values in real time.

1. **Trace the logic step by step**:

* I’d literally walk through the code that determines the discount percentage. Maybe it does one thing for members, then another for non-members, and then something else for people who have certain order totals. I’m looking for places where SAVE20 is supposed to be 20% but somehow another discount is taking over.

1. **Test a fix**:

* Once I spot where the calculation is messing up, I’d adjust the code (for example, ensuring SAVE20 is never overwritten by other conditions).
* Then I’d try the exact scenarios where people previously got the wrong discounts. If it all shows 20% now, that’s a good sign.

1. **Double-check and watch in real usage**:

* Before putting the fix live, I’d have someone else in QA test it in our staging environment.
* After we’re confident, we’d release it and then keep an eye on user feedback to make sure the problem is truly solved.