

Week 2 — Validation Output

Goal

Add input validation and user-friendly error messaging.

Required Outputs

- Validation rules list (required fields, type checks, range checks)
- At least 3 invalid input tests documented
- UI shows errors without crashing
- Trail logs validation status (pass/fail)

Evidence to Attach

- Screenshots: 3 invalid input cases with errors shown
- Screenshot: trail includes a validation step

Reflection (Write short answers)

- Which validation rule caused the most issues?
- How did you prevent crashes?

Weekly Sign-off

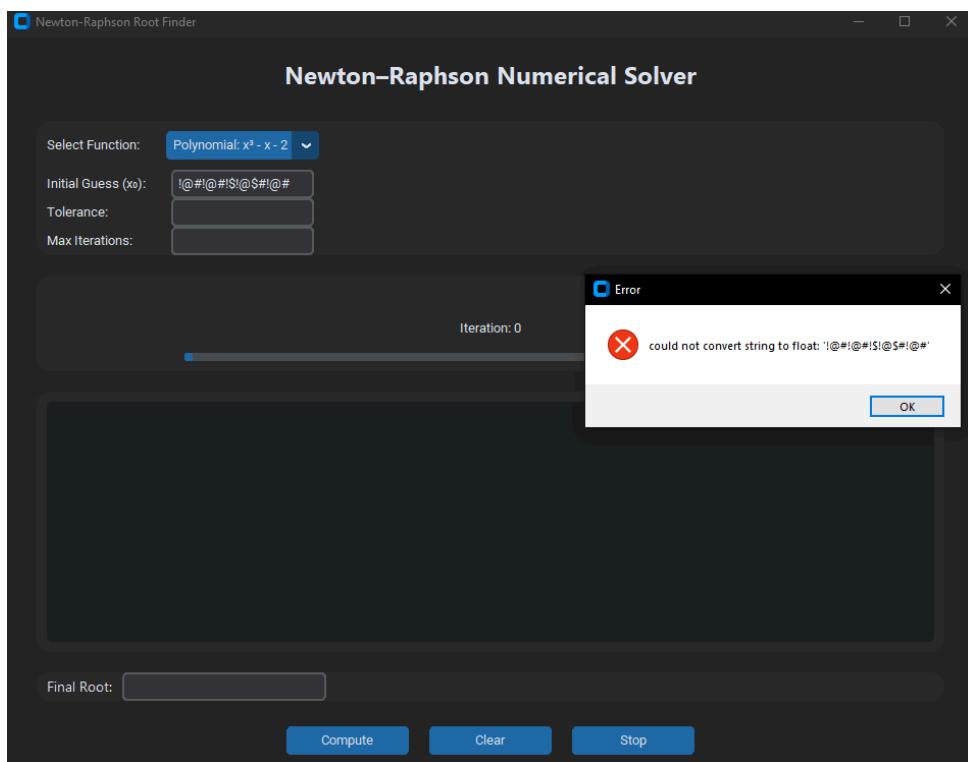
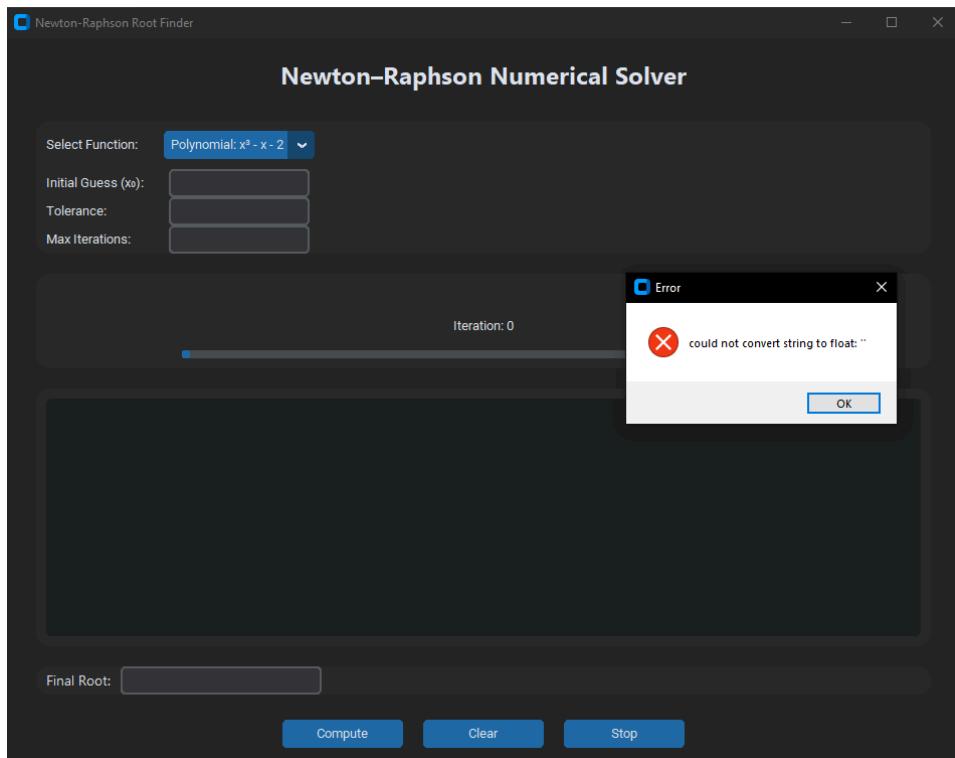
Group Leader Name/Signature: _____ Date: _____

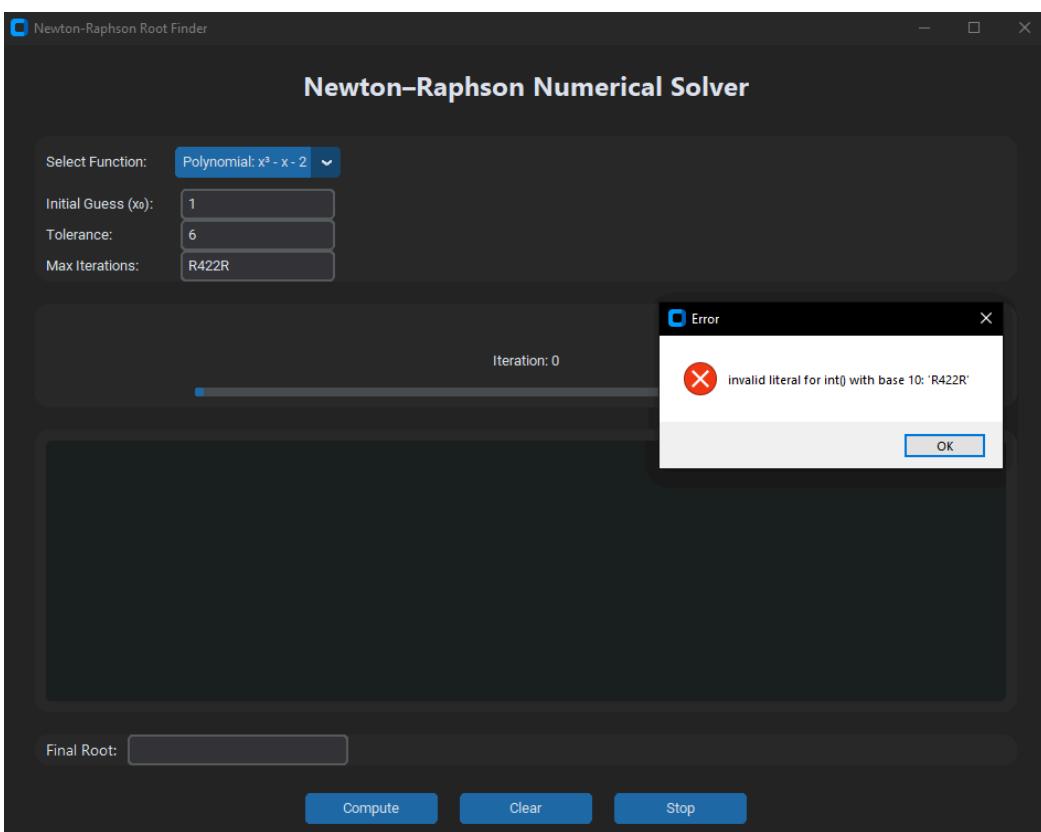
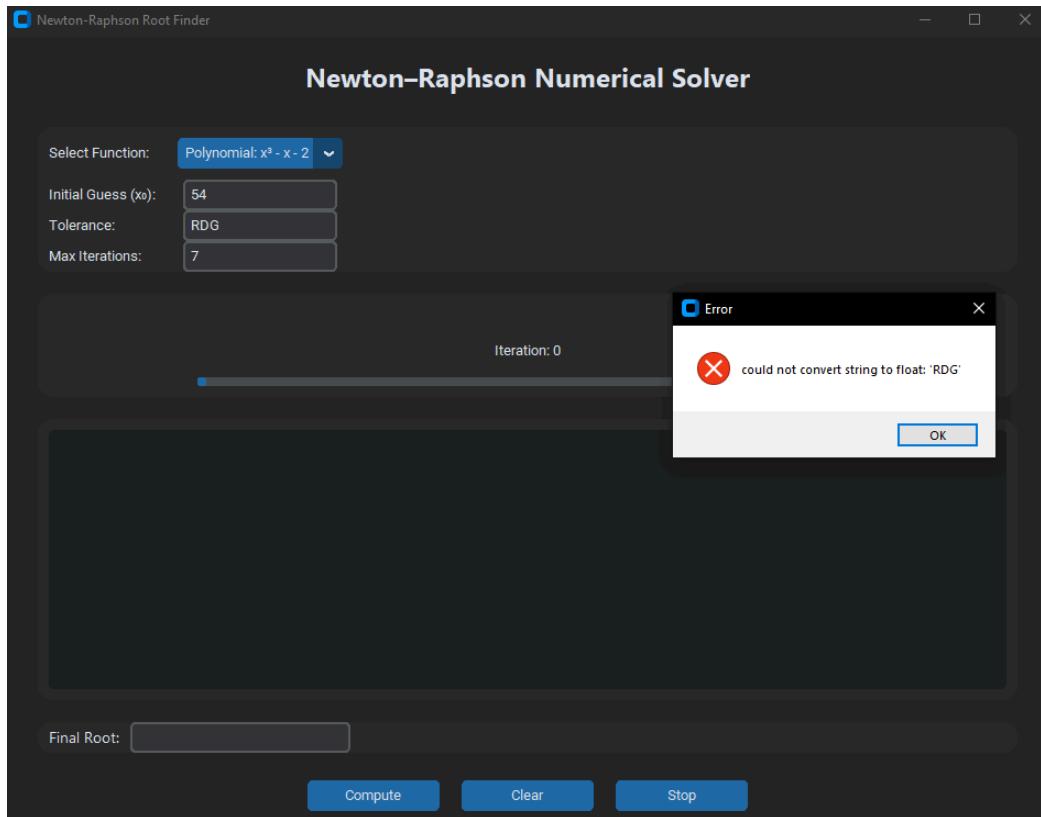
Member Name/Signature: _____ Date: _____

Member Name/Signature: _____ Date: _____

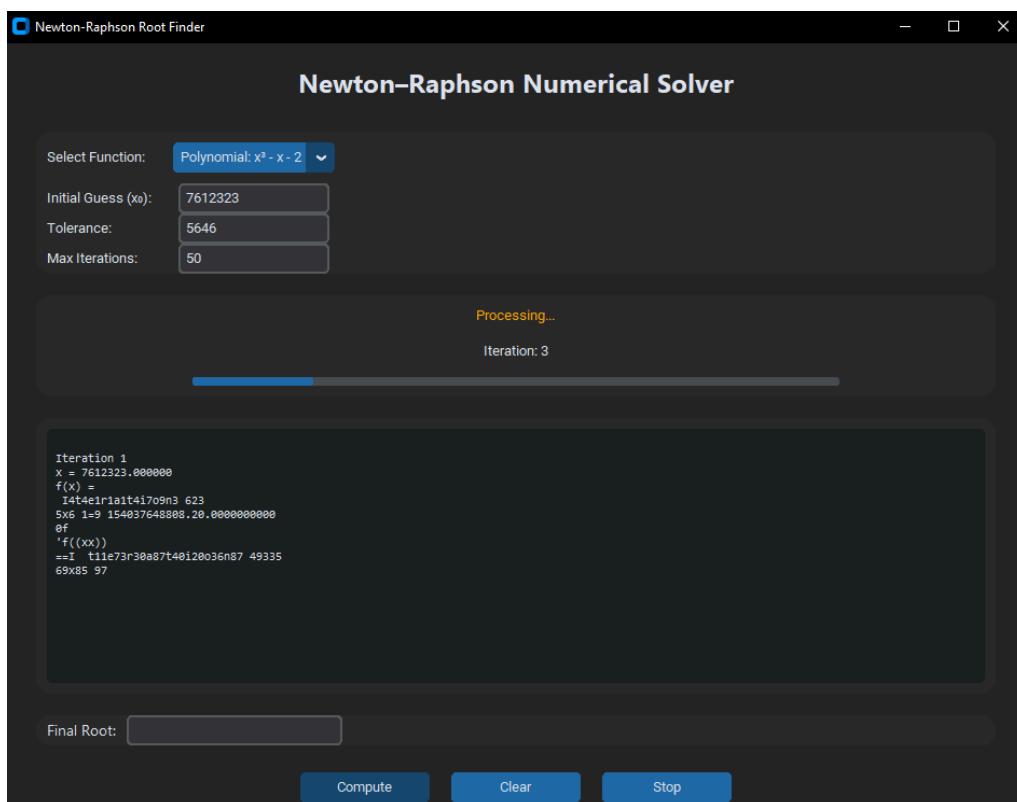
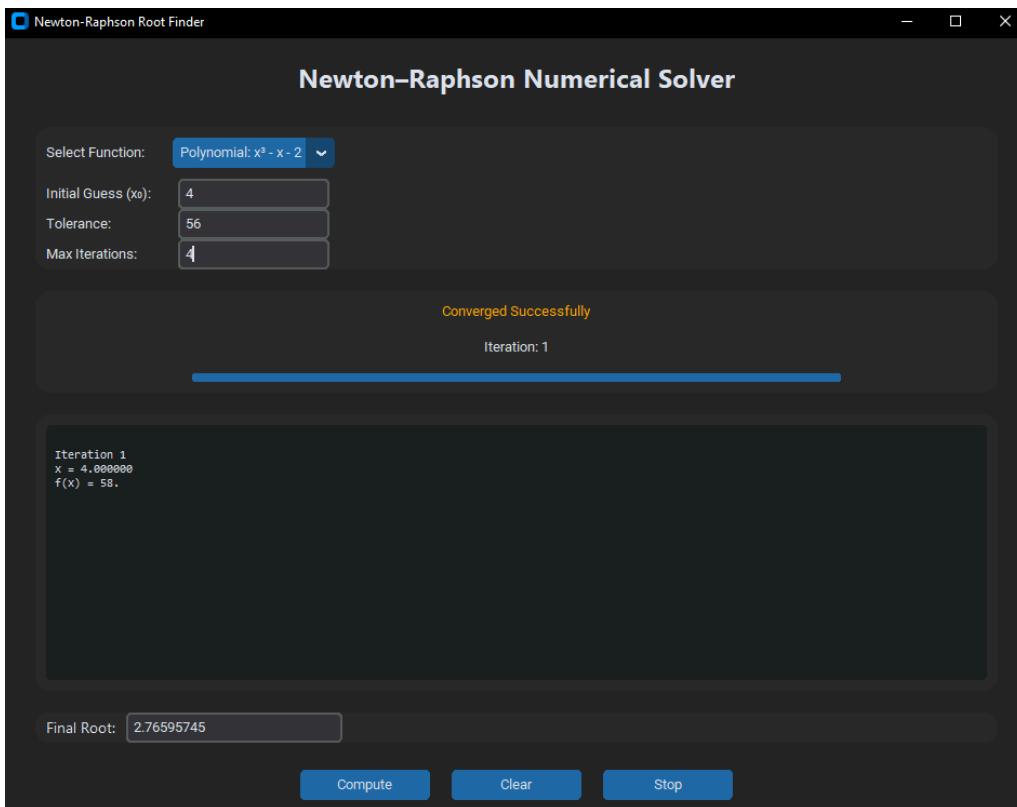
EVIDENCES

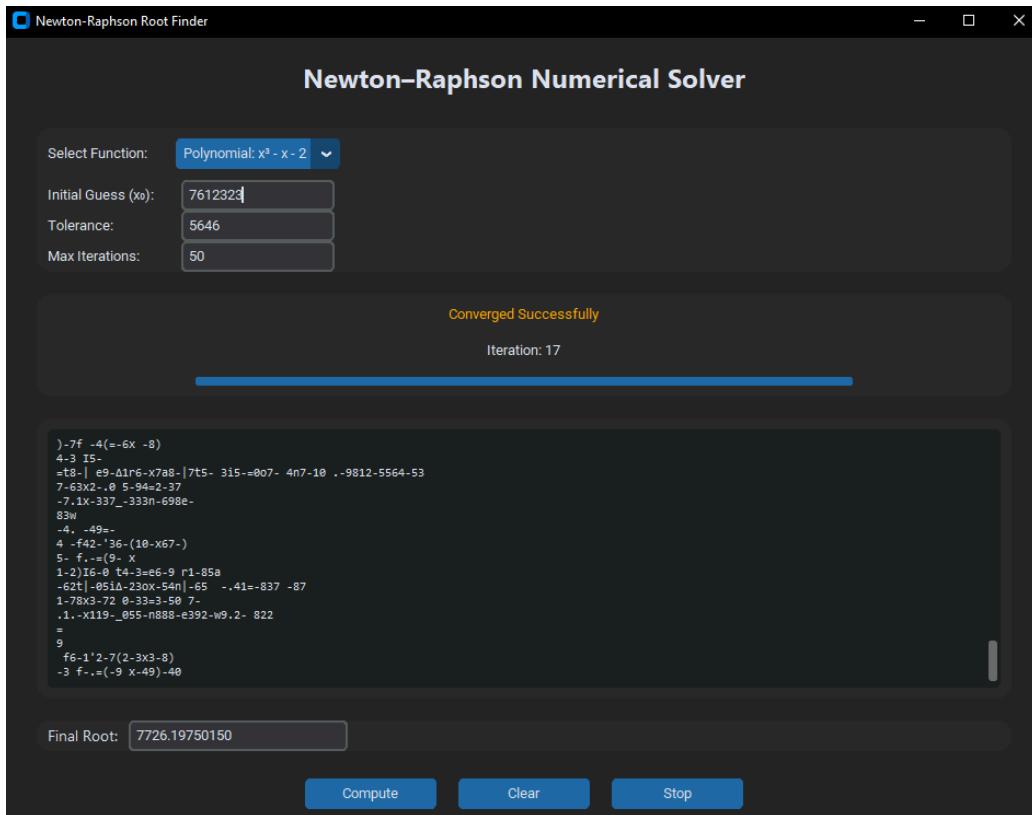
Invalid Input Cases with Errors Shown





Trail includes a Validation Step





Reflection

Which validation rule caused the most issues?

The validation rule that caused the most issues was checking that the inputs are numeric, specifically the initial guess, tolerance, and maximum iterations. If a user typed letters, left a field empty, or entered invalid characters, converting the input to float or int would raise a ValueError, which could crash the program if not handled.

How did you prevent crashes?

Crashes were prevented by wrapping the computation in a try-except block. Any invalid input or runtime error (like derivative being zero) triggers the except block, which uses messagebox.showerror() to display a clear error message to the user. This ensures the app does not crash and provides feedback on what went wrong.

Weekly Sign-off

Group Leader Name/Signature: CHARLES RUSSEL O. SAN JUAN Date: 2/26/2026

Member Name/Signature: CHRISTINA GOMBA Date: 2/26/2026

Member Name/Signature: JOHN MICHAEL TULALIAN Date: 2/26/2026