

Make the error useful

Locate the problem. Explain the effect. Offer a way forward.

1. Identify the actual state

Is the input invalid, access missing, the service unavailable, or submission uncertain? These need different messages.

2. Say what happened in familiar words

Name the relevant field or action. Replace a generic failure with something the user can recognise.

3. Explain the impact on the task

Say whether the person can continue. If you cannot confirm whether submission succeeded, say that instead of guessing.

4. Provide a supported recovery route

Point to the field, access request, saved draft, original task or status check. Preserve valid input where the feature supports it.

5. Check truth and privacy

Do not expose private record details, stack traces or internal identifiers. Do not claim an alert was accepted when it was only sent.

6. Put the message where it can be used

Associate it with the relevant field or task. Give an error summary when several issues need attention; make it usable with a keyboard and assistive technology.

Example: "Enter a description of 500 characters or fewer. Your other answers are still here." Use the second sentence only if the form preserves those answers.

Rewrite one message

Use this worksheet with an error from your own product.

Original message / where it appears

What happened, and what can the system verify?

What does this mean for the user's task?

What supported action can the user take next?

Rewrite: situation + impact + next action

Check: specific, truthful, understandable, safe and actionable. Test the recovery route.