

UX Copywriting — Workbook

This workbook turns the course into reps you can show. Each section mirrors one course module with hands-on exercises, fill-in worksheets, and checklists you run against real screens. Pick one product, app, or website you can edit - your own, a side project, or one you use daily - and carry it through every section. You will finish with an audited flow, a voice-and-tone system, a rewritten error and onboarding set, and a test plan that proves your copy reduced friction.

What UX Copy Is and the Jobs It Does

Learn to see interface copy as a design material and audit any screen against the four jobs: orient, label, instruct, and reassure.

Exercise: The Four-Jobs Screen Audit

Pick one real screen from your chosen product (a form, a settings page, or a dialog). List every piece of text on it. For each line, tag which of the four jobs it is doing - orient, label, instruct, or reassure - then find the job no line is doing. That missing job is usually your biggest fix.

- List every text element on the screen, top to bottom.

- Tag each one: orient, label, instruct, or reassure.

- Which of the four jobs is missing or done weakly?

- Write the one line that would add the missing job.

Worksheet: Marketing Copy or UX Copy?

For one flow in your product, decide whether each touchpoint is persuading a not-yet-committed reader (marketing) or helping a committed user complete a task (UX), and note how that changes the writing. The touchpoint (e.g. landing-page hero, signup button, in-app empty state, error message)

Is the reader deciding or doing?

Marketing or UX copy?

What the writing goal should be here (persuade vs. remove friction)

One way the current copy fights its goal

Exercise: Cut the Cost of a Word

Find three padded strings in your product - error messages, helper text, or onboarding lines are good hunting grounds. Rewrite each to be clearer and shorter. Run both versions through the Hemingway Editor and record the reading grade for each.

- Original string 1, then your rewrite, then the grade level of each.

-
- Original string 2, then your rewrite, then the grade level of each.
-
- Original string 3, then your rewrite, then the grade level of each.
-
- Which filler words did you delete most often (please, just, simply, in order to)?
-

Checklist: Clarity-First Gut Check

- A first-time user could understand each line with no re-reading.
- Every button names an action and outcome, not Submit or OK.
- I led with the meaningful word so a scanner catches it first.
- I replaced at least three vague words with a concrete word or number.
- No line requires the reader to decode jargon or a pun mid-task.
- Each screen does all four jobs: orient, label, instruct, reassure.

Voice, Tone, and Writing for the Moment

Define how your product sounds, map tone to the user's emotional state, and make your language accessible and translation-ready.

Worksheet: Build Your Voice Chart

Define your product's voice in one page. Choose three to four genuinely distinctive traits - avoid friendly and clear, which fit everyone. For each, write a do, a do-not, and a short before-and-after so writers can see it, not just read it.

Trait 1, with its do and its do-not

Trait 1 example: a weak line, then the on-voice rewrite

Trait 2, with its do and its do-not

Trait 3, with its do and its do-not

Where this chart will live so the whole team writes from it

Exercise: One Message, Three Tones

Take a single event in your product (a save, an upload, a signup) and write it three ways while keeping your voice constant: a routine version, a first-time/onboarding version, and a recovered-from-friction version.

Name the emotional state before writing each.

- Routine moment - user is focused and wants speed. Write the line.
-
- First-time moment - user is curious but unsure. Write the line.
-
- After friction - user was frustrated and it finally worked. Write the line.
-
- Note what stayed the same (voice) and what changed (tone).
-

Worksheet: Tone-by-Situation Map

Fill in the tone your product should take in each kind of moment, then write one example line per row. Keep this map next to you when you write so tone becomes a quick decision, not a guess.

Routine moment - how the user feels, the tone to use, and an example line

High-stakes moment (deleting, paying, errors) - feeling, tone, example

Success moment - feeling, tone, example

First-time/onboarding moment - feeling, tone, example

Our fault (downtime, bug) - feeling, tone, example

Checklist: Inclusive and Localization-Ready Check

- [] Link and button text makes sense out of context (Learn more, not Click here).
- [] Errors are not signaled by color alone - paired with a word and an icon.
- [] No directional instructions like the button on the right.
- [] No idioms, puns, or slang that break in translation or slow non-native readers.
- [] Sentences are written whole, not glued from fragments.
- [] Layout leaves 30 to 50 percent extra room for text expansion.
- [] Most UI text uses sentence case.

The Microcopy Toolkit: Buttons, Errors, and States

Rewrite the highest-leverage interface elements - calls to action, form fields, error messages, empty states, loading, and confirmations.

Exercise: Rewrite Five Buttons

Find five generic or vague buttons in your product (Submit, OK, Continue, Send, Yes). Rewrite each so the label names the action and its outcome in the user's words. For any confirmation dialog, rewrite both buttons to be self-describing.

- Button 1: original, then a rewrite that names the action and outcome.

- Button 2: original, then rewrite.

- A confirmation dialog: rewrite both buttons (e.g. Delete project / Keep project).

- Which rewrite do you expect to lift conversion most, and why?

Worksheet: Error Message Rebuild

Take two real error messages from your product and rebuild each using the three-part structure. Place the rebuilt message next to the field that caused it and phrase the fix as an action the user can take now.

Original error message 1

What happened, in plain words

Why (only if it helps the user act)

How to fix it - the action, linked where possible

The final rebuilt message, with the word Error and any code removed

Original error message 2, then its rebuilt version using the same three parts

Exercise: Design an Empty State

Choose a screen in your product that a brand-new user sees before they have any data. Rewrite it from a dead end into a teaching moment using the explain, motivate, act pattern.

- Explain what belongs here.

- Motivate: say why it is useful in one line.

- Act: write the one primary button that fills it.

- Write the full empty state as it would appear on screen.

Checklist: States and Microcopy Check

- Every button names an action and outcome, not Submit/OK/Yes.
- Confirmation dialogs use self-describing buttons, never Yes/No.
- Each field has a persistent visible label, not a placeholder-only label.
- Errors say what happened, why if useful, and how to fix - with a next step.
- Empty states explain, motivate, and offer one action.
- Loading copy names what is happening; long waits set an honest expectation.
- Confirmations echo the specific result and offer the next step or undo.

Onboarding, Process, and Proving It Works

Write onboarding that reaches a first win, run a repeatable rewrite process, and prove your copy reduced friction with a test.

Worksheet: Onboarding to First Win

Map your product's onboarding around the single action that gives a new user their first real win (activation). Frame each step as a benefit, and cut any step that does not lead to the win. The one activation action that delivers the first win

Step 1 written as a benefit, not a feature

Step 2 written as a benefit, not a feature

Steps that can be cut because they do not lead to the win

The rewritten first screen: the payoff, an honest time cost, and one small ask

Exercise: Run the Four-Pass Rewrite

Take one flow you have drafted in this workbook (onboarding, an error set, or an empty state) and run all four editing passes in order. Note what each pass changed so the process becomes a habit.

- Clarity pass - what you fixed so every line is understood first time.

- Concision pass - the filler words and dead phrases you deleted.

- Voice and tone pass - where you matched the voice chart and the moment.

- Edge-case pass - long content, error/loading states, accessibility, translation room.

Worksheet: Copy Test Plan

Design a test that proves one copy change reduced friction. Pick a single friction metric, change only the copy, and decide how you will validate it qualitatively first and quantitatively second.
The copy change being tested (version A vs. version B)

The one friction metric (completion, click-through, error rate, drop-off, or support tickets)

Qualitative check first (cloze test, five-second test, or highlighter test) with ~5 users

How traffic will be split and the tool used (Optimizely, VWO, or built-in)

What result would make B the winner, and how you will avoid a small-sample false positive

Checklist: Ship-Ready Copy Check

- Onboarding drives to one first win, with each step framed as a benefit.
- Progress is shown (Step 2 of 4) and skipping is safe and obvious.
- I ran all four passes in order: clarity, concision, voice/tone, edge cases.
- Copy lives in the design (Figma), not a disconnected doc.
- Reading grade checked in Hemingway; spelling and grammar checked.
- I validated with about 5 users before shipping.
- An A/B test is set up on one metric with one variable changed.
- Results will be documented in a swipe file for the team.

Your Action Plan

1. Choose one product, app, or website you can actually edit and carry it through the whole workbook.
2. Audit one core screen against the four jobs - orient, label, instruct, reassure - and add the missing job.
3. Write a one-page voice chart with 3 to 4 distinctive traits, each with a do, a do-not, and an example.
4. Build a tone-by-situation map and pin it where you write so tone becomes a fast decision.
5. Rewrite your five worst buttons to name the action and outcome, and fix any Yes/No confirmation dialogs.
6. Rebuild your two worst error messages using the what-happened, why, how-to-fix structure.
7. Turn one empty state into a teaching moment with explain, motivate, and one action.
8. Rewrite onboarding to reach the first win, framing each step as a benefit and cutting the rest.
9. Run the four-pass rewrite on one full flow and store the copy in Figma with Ditto or Frontitude.
10. Set up one A/B test on a single friction metric, validate with ~5 users first, and log the result.

