SKILLSTACK SUITE

E-Learning Course Creation & Design — Lesson Workbook

All twelve modules in full: the idea, the method, the mistake to avoid, and the build task for each.

How to use this workbook

Twelve modules. Each one gives you the idea, the method, the mistake most people make, and a build task that produces a real artefact. Do the tasks in order — later modules assume the earlier artefacts exist. Reading without building will not get you the outcome.

Module 1 — Outcome definition

What you will learn. How to state the single change your course produces, and why every other decision depends on it.

A course is a promise that someone will be able to do something afterwards that they could not do before. Most creators skip straight to topics, which is why their modules sprawl — without a stated outcome there is no principle for deciding what to leave out. The transformation statement is the tool: one sentence, present-tense, describing a capability. "Understand instructional design" is not an outcome, because nobody can tell whether it happened. "Design and ship a 6-module course with a completion rate above 40%" is, because it either happened or it did not.

The method

- Write the sentence: "By the end, you will be able to ___."
- Test it against the falsifiability check — could an observer tell whether it happened?
- Name the starting point you are assuming; anyone below that line is not your learner.
- Write the one thing you are deliberately NOT teaching. This is what keeps scope honest.

Most common mistake. Writing an outcome about knowledge rather than capability. If the verb is "understand", "learn about" or "explore", rewrite it with a verb someone could film.

Build task. Write your transformation statement and your explicit non-goal. One sentence each.

Module 2 — Audience & prerequisite mapping

What you will learn. How to define the learner precisely enough that you can make content decisions without guessing.

Every lesson you write makes an assumption about what the learner already knows. When those assumptions are unstated they drift, and the course ends up simultaneously too basic in one module and impenetrable in the next. Writing the prerequisite down converts an invisible assumption into a design constraint you can check every lesson against.

The method

- Write a one-paragraph learner profile: role, context, why they are here now.
- List the five things you assume they already know.

- List the five things they are most likely to get wrong.
- Decide what you do for someone who arrives below the prerequisite — a pre-module, a reading list, or an honest "this is not for you yet".

Most common mistake. Writing for the version of yourself who already finished. Your learner does not have your context, and cannot ask you a question mid-lesson.

Build task. Write the learner profile and both five-item lists.

Module 3 — Backwards design

What you will learn. Why you design the final assessment before writing a single lesson.

Backwards design inverts the usual order: outcome, then evidence, then content. You build the thing that proves the outcome happened, and only then ask what someone needs in order to produce it. This is uncomfortable because it front-loads the hardest thinking, but it is the single biggest lever on completion rate — every module now has an obvious reason to exist, and anything that does not serve the assessment gets cut without agonising.

The method

- Design the final artefact the learner produces — not a quiz, a real output.
- Write the standard: what does a good version look like?
- List every capability required to produce it.
- Group those capabilities into modules. That grouping IS your module map.
- Delete anything on your original topic list that did not survive step 3.

Most common mistake. Designing modules first and bolting an assessment on at the end. You will discover half your content leads nowhere and be too invested to cut it.

Build task. Design your final assessment and build the module map backwards from it.

Module 4 — Lesson architecture

What you will learn. How to chunk content so working memory can actually hold it.

Working memory holds a handful of new elements at once. Exceed that and comprehension does not degrade gracefully — it collapses. This is why the 6-12 minute lesson is a real constraint rather than a stylistic preference: one idea per lesson, demonstrated once, practised immediately. Long lessons are almost always several lessons that have not been separated yet.

The method

- Break each module into lessons of one idea each.
- Target 6-12 minutes per lesson; anything over 15 gets split.
- Order lessons so each one uses the previous one’s output.
- Put a practice beat at the end of every lesson, not at the end of the module.

Most common mistake. Batching all practice into an end-of-module exercise. By then the learner has forgotten lesson one and the exercise feels like an exam.

Build task. Break one module into lessons and mark the practice beat in each.

Module 5 — Scripting & delivery

What you will learn. The four-part lesson pattern, and how to script without sounding scripted.

The reliable pattern is hook, context, demo, practice. The hook states the problem the lesson solves; context says why it matters and where it fits; the demo shows the thing being done; practice hands it over. Scripting matters most in the first thirty seconds and the last twenty — the middle can be bulleted beats you speak naturally, but openings and closes read badly when improvised.

The method

- Write the hook verbatim.
- Bullet the teaching beats — do not write full prose for the middle.
- Script the close verbatim, ending on the practice task.
- Read it aloud once and cut every sentence that does not advance the idea.

Most common mistake. Reading a full script word for word. It flattens delivery. Script the edges, speak the middle.

Build task. Script one complete lesson using the four-part pattern.

Module 6 — Production basics

What you will learn. Why audio outranks video, and the minimum setup that sounds professional.

Viewers forgive mediocre picture and abandon bad audio within seconds. A USB microphone close to your mouth in a soft-furnished room beats an expensive camera in an echoey one. Beyond audio, the second-biggest factor is consistency: the same framing, level and background every session, so lessons recorded weeks apart still feel like one course.

The method

- Get the microphone within a hand-span of your mouth.
- Record in the softest room you have; a wardrobe genuinely works.
- Light from the front, never from behind.
- Record a 10-second test and listen on headphones before every session.
- Write down your settings so you can reproduce them exactly next time.

Most common mistake. Upgrading the camera before fixing the room. Room acoustics dominate perceived quality.

Build task. Set up, record a 60-second test, and write down every setting.

Module 7 — Editing & packaging

What you will learn. A repeatable edit that takes minutes rather than hours.

Editing expands to fill the time available, so constrain it. The goal is not polish, it is removing dead air, mistakes and rambling. A consistent intro and outro, captions, and a readable thumbnail do more for perceived quality than frame-accurate cutting ever will — and captions materially increase completion because a large share of learners watch muted.

The method

- Cut only: silences over two seconds, restarts, and tangents.
- Add the same intro and outro to every lesson.
- Generate captions and correct the technical terms.

- Use a thumbnail with text readable at thumbnail size.
- Export with one preset and never adjust it per lesson.

Most common mistake. Frame-perfect editing. Nobody notices; you lose days.

Build task. Edit your recorded lesson end to end and time how long it took.

Module 8 — Workbooks & assets

What you will learn. How the downloadable turns a passive lesson into a completed task.

The workbook is where learning becomes doing. A good one is not a transcript — it is the blank version of the thing you demonstrated, with the prompts that get the learner to fill it in. If your workbook could be read without watching the lesson, it is a summary; if it cannot be completed without watching, it is a workbook.

The method

- For each lesson, identify the artefact the learner should produce.
- Build the blank template for that artefact.
- Add the three prompts that get them unstuck.
- Include one worked example, clearly marked as an example.

Most common mistake. Shipping slide transcripts as "workbooks". They create no action and no completion.

Build task. Build the workbook for your scripted lesson.

Module 9 — Platform setup

What you will learn. What actually differs between platforms, and how to choose without a month of research.

Platforms differ on four things that matter: drip versus open access, whether cohorts are supported, progress tracking, and what happens to your learner data if you leave. Everything else is presentation. Choose against those four and you can decide in an afternoon. Drip scheduling in particular is worth deciding deliberately — it raises completion for beginners and frustrates experienced learners who want to binge.

The method

- Decide drip vs open access based on who your learner is.
- Decide whether you need cohorts now or ever.
- Check what progress data you get and whether you can export it.
- Check the data-export path before you upload anything.
- Set up completion tracking on day one — retrofitting it loses your baseline.

Most common mistake. Choosing on landing-page aesthetics, then discovering you cannot export your learner list.

Build task. Set up your platform and drip module 1 to three beta learners.

Module 10 — Pricing & positioning

What you will learn. What justifies each price band, and how to avoid selling a $40 course at $400.

Price tracks the size of the outcome and the amount of support attached, not the number of videos. A self-study course with no feedback sits in one band; add structured feedback and it moves; add live cohort time and it moves again. The most common error is pricing on effort expended rather than value delivered — buyers do not compensate you for your production time.

The method

- Write down the outcome's value to the learner in their terms.
- Decide the support level you will actually sustain.
- Pick a band consistent with that support level, not with your effort.
- State plainly on the sales page what is and is not included.
- If you cannot defend the price in one sentence, it is wrong.

Most common mistake. Pricing high on the strength of content volume alone, with no support and no proof of outcome.

Build task. Write your price and the one-sentence defence of it.

Module 11 — Launch sequence

What you will learn. How to launch with evidence instead of hope.

A beta cohort at a reduced price does three jobs at once: it validates the course actually works, it produces testimonials you did not have, and it surfaces the drop-off points before you have spent money on traffic. Launching cold to a list that has never seen your teaching is the slowest path to a first sale.

The method

- Run a small beta at a reduced price with explicit feedback expectations.
- Watch where beta learners stall and fix that lesson first.
- Collect testimonials that name a specific outcome, not "great course".
- Open a waitlist before you open the cart.
- Launch to the waitlist first; they convert far above cold traffic.

Most common mistake. Launching to a cold audience with no beta, no testimonials and no completion data.

Build task. Recruit three beta learners and drip them module 1.

Module 12 — Iteration

What you will learn. Which number to watch, and what to fix first.

Completion rate is the honest metric — refunds lag, reviews are biased toward the people who finished. Find the lesson where learners stop and treat that as the defect. Usually it is one of three things: the lesson assumes a prerequisite you never taught, the practice task is too large, or the lesson is simply too long. Fix one thing, re-measure, repeat.

The method

- Pull completion data per lesson, not per course.
- Find the steepest drop-off.
- Diagnose it as prerequisite, task size, or length.
- Fix that one lesson and re-measure with the next cohort.
- Do not rebuild the whole course on one cohort's data.

Most common mistake. Reacting to the loudest piece of feedback instead of the biggest drop-off.

Build task. Collect completion data from your beta and rewrite the weakest lesson.