

ELITE MASTERMIND CODEBREAKERS

THE CODEBREAKER AI BUILD LAB

EMPLOYEE HANDBOOK + FIELD COURSE

Brief clearly. Build with focus. Verify what works.

Seven modules | 21 lessons | One real capstone

Course promise. Learn to take a real business idea from brief to reviewable website or app prototype with AI, then verify what actually works before calling it live. A focused team may produce a small site in a two-hour sprint. A functioning business takes additional work: operations, customers, payments, support, compliance, and iteration. No course can guarantee overnight revenue.

Audience. EMCB members, KAP LLC staff and collaborators, nonprofit teams, founders, marketers, designers, and project leads. No coding background is assumed; technical implementation and account permissions vary by platform.

Format. Seven modules, 21 short lessons, seven field assignments, one capstone, a 12-question knowledge check, and a 30-day application plan. Suggested pace: two weeks with 45-60 minutes per day, plus one supervised two-hour build sprint. Each lesson follows the EMCB loop: **Learn the principle / Apply it to a real project / Track the work that follows.**

Completion evidence. A brief, source register, page map, prompt log, working prototype, QA log, truthful launch status, and a handoff. Passing standard: at least 80% on the knowledge check **and** 80/100 on the capstone rubric. A completion record represents training, not a professional license, certification from a software vendor, or a guaranteed business outcome.

The operating code

- 1. **Character:** say what is true. A polished preview is not a published site; a click is not a purchase; a form success screen is not delivered email.
- 2. **Discipline:** work from one current brief and a visible task board. Save decisions and version numbers.
- 3. **Knowledge:** verify names, roles, numbers, links, rights, laws, and product capabilities against primary sources or the owner.
- 4. **Execution:** make the next smallest useful change, test it, and move the project forward.
- 5. **Accountability:** record who owns every missing item and what evidence closes it.
- 6. **Community:** ask a teammate to review the experience as a customer, member, or donor, and accept corrections.

The AI assistant is a collaborator. The operator remains responsible for authorization, factual accuracy, customer promises, and launch decisions. A useful prompt is a specification plus evidence and a definition of done, not a magic phrase. “Genius mode” can signal high ambition; it does not replace files, access, verification, or judgment.

The four status words

Status	Meaning	Evidence required
Draft	Copy or code exists, not yet deployed	File or revision link
Preview	Interactive test is available in a private or staging environment	Preview URL and tested functions

Status	Meaning	Evidence required
Published	Production deployment completed	Production URL and deployment result
Verified live	Real user path tested in production	Timestamped test, delivery/transaction evidence where applicable

Never collapse these statuses into “done.” A public page can be live while its checkout, email delivery, booking, or automation remains pending.

MODULE 1 — Translate a vision into a buildable brief

Lesson 1.1 — Start with the person and the action. State who the page serves, what they must understand in 30 seconds, and one primary action. For a nonprofit, this might be “understand the mission, see a project, donate through the approved processor.” For a consulting site: “understand an offer, book a discovery call, and receive a clear confirmation.”

Lesson 1.2 — Separate facts, proposals, and dependencies. Mark each claim as verified, owner-supplied but unverified, proposed, or unavailable. A Promise Pointe organizational chart labeled “proposed” does not prove that a county, contractor, or funder is formally committed. A price from the owner can be published when authorized; a payment flow is functional only after a successful end-to-end test.

Lesson 1.3 — Build a scope ledger. List pages/sections, assets, CTAs, data collection, integrations, owner, and release state. Choose a first release that creates value without pretending unfinished products are ready. Record exclusions explicitly in the internal ledger, not as distracting public copy.

Field assignment 1. Choose one real offer. Deliver a one-page brief with audience, action, five verified facts, two unknowns, content inventory, and a first-release boundary.

Decision rule. If a missing fact changes a public claim or payment behavior, mark that element pending. Continue independent design and copy work.

MODULE 2 — Prompt like a project lead

Lesson 2.1 — Use the CODE prompt.

- **C — Context:** business, audience, current site, existing work, exact files and URLs.
- **O — Outcome:** observable deliverables and desired user actions.
- **D — Details:** constraints, source hierarchy, design, access, preservation, and required integrations.
- **E — Evidence:** acceptance tests, what may be published, and how results will be reported.

Lesson 2.2 — Ask for execution in stages. A strong instruction says: inspect first; write a change map; implement; run relevant checks; publish if authorized; verify; report what remains. It tells the assistant to preserve existing work and resolve routine design choices. The prompt must still leave room to disclose real blockers.

Lesson 2.3 — Revise the failed assumption, not just the wording. When output is wrong, identify the variable: wrong source photo, inaccessible browser, old résumé, ambiguous link, missing API permission, stale build, or untested form. Supply new evidence or choose an authorized route. Repeating “try harder” without changing inputs produces the same uncertainty.

Field assignment 2. Write one CODE prompt for a landing page and a second prompt that corrects a specific observed defect. Log what changed in the input and what changed in the output.

Master execution prompt

You are my product lead, editor, designer, developer, and QA partner for [PROJECT]. The primary user is [AUDIENCE] and their main action is [ACTION]. Inspect the existing site and attached files before editing. Treat [SOURCE] as authoritative for facts; flag conflicts. Preserve [EXISTING CONTENT]. Build [SCOPE] in [PLATFORM], matching [DESIGN SYSTEM]. Use only approved assets and links. Do not invent results, titles, permissions, testimonials, or functional integrations. Make routine design decisions. Deliver a reviewable draft, then implement and publish the authorized changes. Test navigation, mobile layout, accessibility, forms, delivery, and payments as applicable. Report each feature as draft, preview, published, or verified live, with its evidence and any owner-supplied dependency.

Correction prompt

The observed behavior is [EXACT REPRODUCTION] on [URL/DEVICE/TIME]. Expected behavior is [EXPECTED]. Preserve [WORKING PARTS]. Inspect the specific control, trace the failure, make the smallest sound fix, and test the affected path plus one nearby regression path. Show the test result. Do not describe it as fixed until the result supports that conclusion.

MODULE 3 — Research, permissions, and content integrity

Lesson 3.1 — Build a source register. Record each claim, URL/file, date, authority, permission, and publication status. Company websites, official documents, current owner files, and the production site are stronger than search snippets. A public image result can identify a person, but it does not automatically grant reuse rights.

Lesson 3.2 — Handle names, links, and likenesses. Match the exact organization and domain. In our advisory-team work, Jason Tartt, Sr.'s T&T Organics site was torganics.com; tntorganics.com was a different business. An image editor once selected the wrong recent person as a reference. That output was discarded. For a real headshot, use the correct source image, preserve identity, inspect the result, and obtain subject/owner approval appropriate to the use.

Lesson 3.3 — Protect private and regulated information. Never paste passwords, secret tokens, private client health information, bank details, or full donor lists into public course material. Use delegated access and minimum permissions. An IRS letter may prove historical tax recognition while also stating a private-foundation classification and exposing an old address; label the date and classification, and redact only appropriate personal address details in a clearly marked copy.

Field assignment 3. Audit ten claims or assets. Mark verified, proposed, permission pending, or remove. Correct one misleading CTA destination.

The evidence ladder

1. Owner or official source says the fact.
2. The exact approved file, image, or link is in hand.
3. The feature exists in a preview.
4. The production page displays it.
5. A real path completes, including downstream delivery or payment confirmation.

Choose the level required by the public claim. Do not use level 3 evidence for a level 5 claim.

MODULE 4 — The focused two-hour website sprint

Lesson 4.1 — Timebox the first useful release. This sprint is suitable for a small brochure site or landing page when brand assets and copy facts are ready. Complex memberships, checkout, identity verification, and legal review need their own plan.

Minute	Activity	Exit evidence
0-15	Brief, audience, action, facts, release boundary	One-page brief
15-30	Page map and rough mobile-first copy	Section order and CTA labels
30-75	Build the primary page and reusable visual system	Preview with complete content
75-95	Connect real links and supplied assets	Link inventory
95-110	Phone, keyboard, and content QA	Defect log and fixes
110-120	Publish if authorized; verify; hand off	URL and truthful status report

Lesson 4.2 — Keep the design purposeful. Use one clear H1, readable type, contrast, generous spacing, one primary CTA, descriptive links, appropriately sized images, labels on forms, visible focus states, and a usable 375-pixel layout. Skip decorative complexity that delays the user action.

Lesson 4.3 — Keep the handoff small. Record where to edit the hero, services, testimonials, links, policies, and images. Record the current version. A site no one can maintain is an unfinished product.

Field assignment 4. Run the sprint with a partner. Deliver a small site preview and a ten-line QA log. A 120-minute stop is not permission to publish untested claims or purchases.

Site build prompt

Build the first useful release from this approved brief [BRIEF]. Use the existing platform and visual language. Provide mobile-first sections in this order [ORDER], with one primary CTA [ACTION]. Keep copy tied to [SOURCES]. Implement real approved links; label unfinished commerce or automation honestly. Preserve existing content. Test at phone and desktop widths, keyboard navigation, forms and external links. Return the preview or production URL and a change log with actual checks.

MODULE 5 — Integrations that truly work

Lesson 5.1 — Forms and email. A form needs labeled fields, validation, accessible errors, a submission route, spam control, data handling, and a monitored destination. A success screen is not proof of delivery. Test a real submission, verify the destination inbox and timestamp, and avoid sending sensitive data to a general mailbox without a suitable process. For digital signatures, test typing, saving, submission, and the stored record on a phone.

Lesson 5.2 — Commerce and scheduling. A Square link may lead to checkout, but it does not prove price mapping, completed charge, refund terms, tax/shipping configuration, product fulfillment, membership entitlement, or recurring billing. A scheduling link is not an iPhone calendar connection until availability, time zone, booking confirmation, calendar sync, and notifications are tested. If monthly billing is manual, say so.

Lesson 5.3 — Accounts and permissions. Use supported connectors, delegated access, or a user-controlled login window. Do not request passwords in chat. When a browser reports an unsafe credential state, stop that route. Try a fresh authorized session or a supported app connection; carry on with source files and reviewable assets. Never “hack” past authentication or bypass a safety block.

Field assignment 5. Trace one inquiry from button to form to confirmation to inbox. Trace one product from listed price to approved payment URL. Record each tested step and its limits.

Integration test prompt

Map [FEATURE] as a chain of user action, browser response, provider handoff, operator notification, and saved record. Inspect current configuration first. Run a safe test using [TEST DATA]. Report every observed result separately. If a step cannot be tested, label it unverified and specify the exact access or credential needed. Do not create a fake success state.

MODULE 6 — Debugging without giving up

Lesson 6.1 — Reproduce before changing. Capture URL, environment, device, time, expected result, actual result, and any error. Test the smallest path. Preserve a working version before editing. Ask what changed since the last successful test.

Lesson 6.2 — Change the route when evidence changes. If an upload fails, check whether the file exists, its type/size, permissions, and whether a new upload is available. If a plugin is unavailable or declined, use an authorized alternative; do not keep invoking the declined path. If the repository proxy fails, prepare the work locally and report the publication boundary. If the site is cached, confirm the deployed version before blaming the code.

Lesson 6.3 — Close with a regression test. Verify the fix and adjacent behavior. A no-fee PDF that displays fields but does not accept input is still broken. A form that allows typing but cannot submit is still broken. A donation button that opens a provider page should be verified against the approved campaign, not merely checked for an HTTP response.

Field assignment 6. Diagnose one intentionally broken link or form. Deliver a reproduction, root-cause hypothesis, one change, retest, and rollback note.

Recovery ladder

1. **Observe:** reproduce the exact failure.
2. **Inspect:** check files, settings, current version, logs, and permissions.
3. **Isolate:** reduce to one page, control, account, or device.
4. **Change one variable:** input, route, integration, file, or configuration.
5. **Retest:** affected path plus nearby regression.
6. **Escalate:** ask for the missing owner action with a concrete result ready to approve.

Persistence means testing new hypotheses, not bypassing access controls or repeating an unchanged command.

MODULE 7 — Launch, handoff, and improvement

Lesson 7.1 — Run three views of QA. Recruiter/customer scan: do the headline and CTA make sense in 30 seconds? Operator journey: does the submission reach the right person? Technical review: mobile, keyboard, speed, metadata, 404, links, images, and policy text. Report actual checks and limitations.

Lesson 7.2 — Publish deliberately. Confirm title, version, audience, domain, links, and production deployment result. Preserve the existing audience unless the owner explicitly changes it. If a publication tool reports pending, wait for a terminal result. An edited local file is not a live page.

Lesson 7.3 — Build a learning loop. After launch, measure inquiries, qualified leads, checkout starts and completed orders separately. Review user questions and defects weekly. Improve the weakest path, document the experiment, and keep a rollback route.

Field assignment 7. Deliver a launch report with a public URL (if deployed), exact tested paths, unresolved items, owner of each, and the next review date.

Launch report template

Feature	State	Evidence	Next action	Owner
Homepage	Published / verified live	URL, date, phone view	Fix observed defect	Named person
Inquiry	Preview / verified live	Test ID and inbox receipt	Configure delivery	Named person
Payment	Link configured / verified sale	Approved mapping; test receipt	Enable transaction	Named person
Automation	Planned / live	Schedule and observed run	Supply access	Named person

The employee prompt library

Use these templates with real source material. Remove bracketed instructions before sending. Save the final prompt and result in the project log.

1. Audit before editing

Inspect [SITE] and [FILES]. Inventory current pages, CTAs, media, forms, accessibility, policies, and deployed status. Identify what already works and what is unverified. Propose a small change map preserving existing work. Cite the exact source for names, prices, claims, and links.

2. Write executive copy

Draft [PAGE] for [AUDIENCE] and [ACTION]. Use only facts in [SOURCE REGISTER]. Separate direct achievements from supported business scale. Give the first 30 seconds a clear headline, proof, and action. Flag gaps. Avoid generic hype and unsupported numbers.

3. Build an accessible page

Implement [PAGE MAP] in [PLATFORM]. Match [DESIGN TOKENS]. Use semantic headings, descriptive buttons, responsive images, keyboard focus, and reduced motion. Keep an editable path for the owner. Test 375-pixel mobile and desktop layouts; report actual results.

4. Match products to checkout

Make a table of [PRODUCT/SKU], price, variant, photo, and approved payment URL. Open and verify each destination and amount. Do not enable purchase when variants, shipping, tax, fulfillment, or terms are unresolved. Report live and pending states per product.

5. Improve a form

Reproduce the reported form problem at [URL]. Test focus, typing, validation, submission, confirmation, and delivery. Keep existing submissions safe. Fix the smallest cause and retest on phone and keyboard. Include the recipient's proof of delivery where available.

6. Prepare an image edit

The attached image is the identity reference for [PERSON]. Preserve recognizable features and only adjust [BACKGROUND/ATTIRE/CROP]. Match the team's [STYLE]. Do not substitute another person. Inspect the result against the source before it is used publicly.

7. Research a link or claim

Verify [NAME/CLAIM/URL] against the official organization and a second reliable source where useful. Distinguish similar domains and people. Record the source, date, exact supported claim, and any uncertainty. Do not infer affiliation from proximity in search results.

8. Troubleshoot access

We are authorized to edit [SITE], but [EXACT ERROR] blocks [ACTION]. Inspect supported access routes and current tool state. Try one fresh authorized session or supported connector. Do not bypass security controls or request a password in chat. Continue preparing reviewable changes and identify the minimum owner action needed.

9. Verify a launch

Inspect the actual production URL [URL]. Test [CRITICAL JOURNEYS] on phone and desktop, including downstream confirmation. Compare against the approved brief and current assets. Report deployed version, observed pass/fail, untested items, and owner actions. Do not call it live based only on a preview.

10. Preserve context across a handoff

Summarize the project for another employee: objective, source-of-truth files, current production URL/version, approved decisions, completed changes, tested results, blockers, credentials/access route (without secrets), and the next three actions. Separate verified facts from assumptions.

Case files from our work

The advisory link mismatch. Two similar domains appeared for T&T Organics. The official farm site identified the owner, while the alternate domain belonged to another store. The lesson: a plausible URL is not an

approved destination.

The wrong-face image edit. A “recent image” reference selected Ty instead of Tyrone. The result was discarded. The lesson: label the exact source path, inspect identity, and do not publish an unverified likeness.

The tax letter. The historical determination recognized tax exemption and classified the organization as a private foundation. The lesson: quote the document's actual classification and date; do not convert a historical letter into a claim of current public-charity standing.

The form that looked fillable. A visible field can still fail to accept typing, save, or submit. The lesson: test each step in the actual user journey, then verify delivery.

The login barrier. A browser credential-state failure stopped WordPress editing despite owner authorization. The lesson: authorization does not make an unsafe session usable. Prepare the materials, try a supported fresh route, and report the actual publication state.

The Square link. A product button can open the right price while shipping, tax, recurring billing, and fulfillment remain incomplete. The lesson: report checkout mapping and transaction verification separately.

Capstone: launch a small, honest digital experience

Brief. In a team of two or three, create a mobile-first, four-section landing page for a real approved service or community program. Include a hero, proof/offer, how it works, and contact action. Do not use fictional testimonials. Connect one real approved external link and either a tested inquiry form or an explicitly labeled contact route. Produce a source register, change log, and launch report.

Time. Four hours for planning/building/testing plus a 15-minute demo. The first 120 minutes are the focused build sprint; the remaining time is for integration, correction, and evidence.

Rubric	Points
Audience, outcome, and scope are clear	10
Verified copy, claims, rights, and links	20
Mobile design and accessibility	15
Working user path and truthful integration state	20
QA evidence and defect correction	20
Handoff, owner actions, and version record	15

Fail conditions even if the point total is high: fabricated results; a false payment or booking success; another person's likeness presented as the subject; unauthorized account access; sensitive data exposed; claiming publication without a deployment result.

Knowledge check

1. A form displays “sent.” What additional evidence verifies delivery?

2. What is the difference between a preview URL and a production deployment?
3. What should you do when the image editor uses the wrong person's face?
4. Which claim can a dated IRS determination letter support without more research?
5. What must you verify before mapping a product to a Square link?
6. A browser refuses to resume credentials. What is the authorized recovery path?
7. Why does “try again in genius mode” rarely fix an unchanged input?
8. Name two mobile accessibility checks.
9. What evidence supports calling a checkout “verified live”?
10. How should a proposed partner appear in a project chart?
11. What belongs in the handoff to another employee?
12. Does a two-hour site sprint guarantee a successful business? Why?

Answer key. 1: receipt in destination inbox or stored record with timestamp, not just client UI. 2: preview is a test environment; production is a saved, deployed version at the live address. 3: discard, use exact identity source, inspect and approve. 4: the historical recognition, effective date, and stated classification. 5: exact product, price, variant, destination, terms, shipping/tax/fulfillment state. 6: fresh supported session/connector or owner handoff, with no bypass. 7: the failed assumption or access state remains unchanged. 8: e.g. 375-pixel readability and keyboard focus/labels/contrast. 9: approved mapping plus completed test payment/receipt and fulfillment path as applicable. 10: label proposal and pending agreement, no implied commitment. 11: brief, sources, version, decisions, tests, blockers, next owners. 12: no; customer acquisition, operations, support, finances, and iteration take further work.

30-day application plan

- **Week 1 — Learn.** Complete modules 1-3; audit one current project and build its source register.
- **Week 2 — Build.** Complete modules 4-5; run the supervised sprint and trace one real integration.
- **Week 3 — Verify.** Complete modules 6-7; fix a defect and complete capstone QA.
- **Week 4 — Improve.** Demonstrate the project, peer-review another team, and measure one useful user action. Record a revised prompt and one operating improvement.

Team rhythm. Each day: a ten-minute brief, one focused work block, and a five-minute evidence update. Each week: review shipped work, unverified claims, customer questions, and one experiment. Keep the mission ambitious and the status honest.

Printable employee worksheets

A. One-page project brief

Project / owner / date: _____ **Primary audience:** _____
What the audience must understand in 30 seconds: _____ **One primary action:** _____
Approved platform and production URL: _____
Source-of-truth files and last-updated dates: _____ **Five verified facts:** 1. ____
2. ____ 3. ____ 4. ____ 5. ____ **Claims or permissions still pending:** _____

First-release scope: _____ Explicitly later: _____ Owner approval already granted for: _____ Definition of done and test evidence: _____

B. Prompt and decision log

Time	Prompt / changed input	Observed result	Decision / next evidence
—	—	—	—
—	—	—	—
—	—	—	—

C. Defect and retest card

URL and version: _____ Device/browser: _____ Time: _____ Expected: _____ Actual: _____
Reproduction steps: 1. _____ 2. _____ 3. _____ Likely cause / evidence: _____
One change made: _____ Rollback route: _____ Affected-path retest: Pass / Fail / Untested
Nearby regression test: Pass / Fail / Untested Reviewer and timestamp: _____

D. Launch and handoff card

Production URL: _____ Deployed version: _____ Audience/access: _____ Phone and desktop tested: _____ Keyboard tested: _____ Form delivered to inbox or saved record: _____
Payment test: _____ Approved owner content: _____ Unverified items: _____
Next action / named owner / due date: _____ Rollback or support contact: _____