2 COURSE 2 · FACILITATOR PACK

Builder Orientation

From AI user to AI-assisted builder. Decompose first, prompt second. One live build, one student build, one paper drill.

DURATION 2 hr (1 break)
AUDIENCE Aspiring builders
PREREQ Course 1 (AI Fluency)

0:00–0:15	0:15–0:40	0:40–0:50	0:50–1:30	1:30–1:50	1:50–2:00
M1 User → Builder 15 MIN · TALK	**M2 Live Build (Equipment Tracker)**	**Break** 10 MIN	**M3 Student Build + Peer Review**	**M4 Decomposition Drill (paper)**	**M5 Wrap & Assignment** 10 MIN · TALK

PRE-SESSION PREP

Done 24 h before; verified at T-30 min

- **Account check.** Every student needs Power Apps + Power Automate licensed in M365 and a working AI tool (GenAI.mil preferred). Run a roster check 48 h before; pair the unprovisioned with a provisioned partner.
- **Tabs open.** Deck (Week 2), this pack, GenAI.mil, *make.powerapps.com*, a clean SharePoint test site you can create the *EquipmentCheckout* list on during M2.
- **Do *not* pre-build the app.** The whole value of M2 is the room watching you decide live. Have an empty SharePoint site ready; have GenAI.mil and Power Apps open in separate tabs; have nothing else staged.
- **Fallback ready.** Bookmark the "Debugging Exercise Fallback" on the instructor page (orientation.html#module-2) in case Power Apps or GenAI.mil is down for the whole session.
- **Paper exercise.** Print 1 decomposition worksheet per student (or post a Word/Forms link); have whiteboard

LIVE HAND-OFF CUES

Two switches today; narrate every decision

- **0:15 — deck → Power Apps.** Open *make.powerapps.com*. Press P in the deck to keep speaker notes pinned. Audible cue: "Watch — do not type along."
- **During M2.** Narrate *every rejection*: when the AI suggests JavaScript instead of Power Fx, stop and re-prompt out loud. The rejection is the lesson.
- **0:40 — back to deck** for break slide. Leave the half-built tracker visible in a second tab.
- **0:50 — M3 hand-off.** "Now you build. I'll roam in chat. If you get stuck, post the screenshot — do not DM me."
- **1:30 — Power Apps → paper.** Close laptops audibly. M4 is whiteboard / paper only. The whole point is no AI.

WHEN THIS GOES SIDEWAYS

Top failure modes & recovery

- ▶ **Power Platform / GenAI.mil / internet is down for the whole session.**
  → Switch to the four-bug debugging exercise on the instructor page (orientation.html#module-2, "Debugging Exercise Fallback"). Walk the room through diagnosing the broken Filter / Patch / Overdue code on paper. Do *not* cancel.
- ▶ **AI emits JavaScript / C# / Python instead of Power Fx.**
  → Show the error. Re-prompt: "Use Power Fx for canvas apps only — no JavaScript, no forEach, no arrow functions." Use it as a teaching moment.
- ▶ **Half the room is unprovisioned in Power Apps.**
  → Pair-and-share — one driver, one navigator per laptop. Log the unprovisioned roster for the program lead before close.
- ▶ **Student skips decomposition and just prompts.**
  → Stop them. "Show me your fields and actions on paper. We will not unblock you until that exists."
- ▶ **Live build runs long; M3 is being squeezed.**
  → Cut M2 prompts 5 (debug) and 6 (polish) — the room will hit their own errors in M3 and that is more authentic. Protect

POST-SESSION HOMEWORK

Issue at Slide 27, before logoff

- **Build one tool of your own** (Form + Gallery class). Pick a problem from your section. Decompose on paper first. Deploy to a test SharePoint site.
- **Bring the failure case to Week 3.** Whatever broke or confused you. We will use it as a live debugging case during the Frontier-recognition break.
- **Verify M365 access** before Week 3: make.powerapps.com, make.powerautomate.com, a SharePoint site you can write to. Email the program lead 48 h out if any are missing.
- **Read the SOP · Decomposition** page. The paper drill from M4 is the foundation we build on next week.

EXERCISE & ACTIVITY PROMPTS

Read these to the room verbatim

M2 · Decompose-before-build (5 min, whiteboard). "Before I touch anything — what data fields do we need? What does the user need to do? What is the simplest useful version? What data structure backs it?" Capture: item name, serial number, assigned-to, date out, date due, status (out / in / overdue). Actions: check out, check in, see what's out, see overdue.

M2 · The six live-build prompts (chat-paste blocks in week-2-handouts.html). Run in order, narrating every decision:

1. **Prompt 1 · Define the problem (2 min):** ask for a plan, not code — rifle section, ~30 people, ~50 items.
2. **Prompt 2 · Data structure (3 min):** SharePoint list *EquipmentCheckout* + CSV template + post-import steps for the Status Choice column and the IsOverdue calculated column.
3. **Shortcut (audible):** SharePoint → Integrate → Power Apps → Create an app. Power Apps auto-generates the form and gallery.
4. **Prompt 3 · Refine the interface (5 min):** filter the gallery to checked-out items, color overdue rows red, add a check-in button.
5. **Prompt 4 · Iterate (3 min):** add a SearchBox + an overdue warning icon, sort overdue to the top.
6. **Prompt 5 · Debug (4 min):** intentionally introduce one error (wrong Choice syntax or column-type mismatch). Show the debug loop live.
7. **Prompt 6 · Polish (3 min):** Marine red/gold/dark color scheme.

M3 · Student Build (25 min). "Pick one starter problem — Leave request tracker, Training attendance log, Vehicle inspection checklist, or your own (clear scope with me first). Decompose on paper for 2 minutes. Then start prompting. You will hit at least one error — that is the assignment. Post errors and your re-prompts in chat so the room sees the recovery."

M3 · Peer Review (15 min). Pair students. "Six minutes each: demo your build, then your partner walks the four checks — clear problem statement, core function works, evidence of iteration, can explain decisions. Last minute: full-class debrief, one or two standout examples."

M4 · Individual Decomposition (10 min, paper). "Pick *one real*

2 SHARE-IN-CHAT URL SHEET

Builder Orientation — class links

Every URL the Week 2 deck cues you to drop into Microsoft Teams chat. Read aloud or type into the chat — no copy buttons since this is the printed pack.

DECK MATERIALS

The four files instructors typically share at the start of the session.

Week 2 Student Handout

When: Drop in chat at session start. Lists what to bring, the key terms, and the Module 4 worksheet prompts on one landscape page.

https://jeranaias.github.io/ExpertDrivenDevelopment/handouts/week-2-builder-orientation.html

Week 2 Facilitator Pack (HTML)

When: Send to a co-facilitator running Module 3 alongside you.

https://jeranaias.github.io/ExpertDrivenDevelopment/facilitator/week-2-builder-orientation.html

Week 2 Facilitator Pack (PDF)

When: Email-friendly one-pager.

https://jeranaias.github.io/ExpertDrivenDevelopment/facilitator/pdf/week-2-builder-orientation.pdf

Week 2 PowerPoint deck

When: For instructors who present from PowerPoint or want to edit the slides locally.

https://jeranaias.github.io/ExpertDrivenDevelopment/pptx/week-2-builder-orientation.pptx

Week 2 slide deck (HTML)

When: For students who missed the live session.

https://jeranaias.github.io/ExpertDrivenDevelopment/decks/week-2-builder-orientation.html

TEMPLATES & REFERENCE DOCS

Week 2's Module 4 (Decompose) and the homework cue students toward the EDD templates — share these so they can fill them in during or after class.

EDD Templates index

When: Slide 22 points students to a worksheet template "in their student companion." Drop this link so they can grab the markdown templates directly.

https://jeranaias.github.io/ExpertDrivenDevelopment/resources/templates.html

Problem Definition template

When: Hand to students who need a structured sheet to scope their first prototype before building.

https://jeranaias.github.io/ExpertDrivenDevelopment/templates/problem-definition.md

Development Journal template

When: For the homework: capture what worked, what broke, and what AI did well between Week 2 and Week 3.

https://jeranaias.github.io/ExpertDrivenDevelopment/templates/development-journal.md

Tool Registry Entry template

When: Once a student deploys their prototype, this is the form they fill in to register it.

https://jeranaias.github.io/ExpertDrivenDevelopment/templates/tool-registry-entry.md