

Full-Stack Capstone

Eight build cycles, one architecture, one shipped container. React + Vite + TypeScript · Go + HTTP · persistent store · AI integration · auth · Docker.

DURATION 8 hr (2 breaks + lunch)

AUDIENCE Capstone candidates

PREREQ Courses 1–4 + deployed tool + proposal

6

COURSE 6 · FACILITATOR PACK

0:00–0:30 M1 Full-Stack Frontier FRAMING	0:30–1:15 M2 Environment Setup SETUP	1:15–2:00 M3 Backend from Scratch BUILD ·	2:00–2:45 M4 Data Layer BUILD · ARCH 2/4	2:45–3:00 Break 15 MIN	3:00–3:45 M5 Frontend from Scratch BUILD ·	3:45–4:30 M6 Pages & Navigation BUILD	4:30–5:15 M7 AI Chat Integration BUILD	5:15–5:30 Break (often lunch) 15 MIN	5:30–6:15 M8 Auth & Middleware BUILD	6:15–7:00 M9 External Integrations BUILD · COMPRESS	7:00–7:45 M10 Docker & Deploy BUILD · ARCH 4/4	7:45–8:00 Assessment & Wrap CLOSE
---	---	--	--	-------------------------------------	---	--	---	---	---	---	--	--

PRE-SESSION PREP

Done 1 week before; verified at T-60 min

- **Capstone gate.** Every attendee submitted a proposal >= 7 days out. Reject anyone without an approved proposal — 8 hours is no place to discover scope drift.
- **Machines.** Each laptop runs Node 20+, Go 1.22+, Docker Desktop (admin), VS Code or equivalent, GenAI.mil access, GitHub or local repo. Smoke-test 24 h before.
- **Reference repo as your safety net.** Pull `heywood-inventory/` and smoke-test the backend + frontend the night before. Keep `docs/PROMPTS.md` (validated prompts) and `docs/INSTRUCTOR_GUIDE.md` (module → file map) on a side monitor — yours, not the students'.
- **Architecture diagram + breaks.** Architecture printed and posted (the one slide you return to nine times). Coffee 0815, snacks 1045 & 1530, lunch call at 1230 — M7 break often becomes lunch.

LIVE HAND-OFF CUES

Constant deck → IDE switching — the architecture is the anchor

- **Architecture anchor (every module).** End M2 by stopping on the architecture slide — “you’ll see this exact picture eight more times today.” Each build module then ends with a 60-second “arch lights up” beat: deck → diagram → back to IDE.
- **End of M3 (1st curl response).** Twenty seconds of silence. This is the emotional checkpoint of the day — let it land.
- **M4 Path A vs Path B.** Default Path A (JSON store now, SQLite as a stretch — both build-tagged in the reference repo). Path B (Microsoft Graph) only for students whose proposal already requires M365. Announce the room default at M4 kickoff.
- **M9 compression rule.** If you’re behind by 15 min entering M9, cut external integrations to a stub. Everyone ships a container by 1700 — no exceptions.
- **1700 wrap.** Each student demos their running container in 60 sec. You moderate; you do not debug.

EXERCISE & ACTIVITY PROMPTS

Read these to the room verbatim

- M1 (30 min).** “You stop being a Power Apps builder today. Today you direct AI to build whatever the problem needs — backend, frontend, data, auth. Look at the architecture. Four layers. Eight builds. One container at 1700.”
 - M2 (45 min).** “Most boring module of the day; most expensive if we skip it. Verify Node, Go, Docker. AI walks each of you through your own machine — do not let me do it for you.”
 - M3 (45 min).** “Direct AI to write a Go HTTP server that returns a hard-coded JSON array on `GET /items`. You curl it. When you see `200 OK`, raise a hand. Twenty-second pause for everyone to feel that.”
 - M4 (45 min).** “Interface-first. Define what the data layer *does* before you decide how it stores. AI implements the interface against an in-memory store first; persistence in a stretch.”
 - M5–M6 (90 min).** “React + Vite + TypeScript. Proxy to the Go backend. Render the items table. Then add routes for two pages and a nav. Resist beauty — we add features in M6, polish never.”
 - M7 (45 min).** “Wire the OpenAI-compatible chat endpoint to your backend. Stream tokens to the React UI. The first time it streams, applaud once.”
 - M8 (45 min).** “Auth middleware on the backend; login route on the frontend. Single test user. Real password hashing. No production claims yet.”
 - M9 (45 min, compressible).** “One external integration of your choice from the proposal. If the API gives you trouble — stub it. M10 is non-negotiable.”
 - M10 (45 min).** “Dockerfile, build, run. Ship the container. URL or screenshot in chat by 1745.”
- ANCHOR PHRASE** “Architecture first. Light up one box at a time. Everyone ships a container by 1700.”

WHEN THIS GOES SIDeways

Stuck-student triage & top failure modes

- **Stuck-student triage (~3 min).** → 3-Minute Rule → re-run the module’s prompt from `PROMPTS.md` → open the matching file via `INSTRUCTOR_GUIDE.md` on your screen, describe the diff → pair & move on.
- **Docker won’t install (admin denied).** → Defer Docker to M10; pair with a Docker-enabled buddy for the deploy. They still ship.
- **CORS / proxy errors stall the room at M5.** → `Project web/vite.config.ts` from the reference repo and walk the proxy block once; do not re-explain individually.
- **Behind 30+ min at M8, or M9 API throws 401s.** → Stub M9 or cut it — announce it. M10 must ship; capture as Day 2 homework

POST-SESSION PREP

Issued in the 1745 wrap; capstone hand-off

- **Push the container** to your unit’s registry within 72 h. URL goes in the program shared library.
- **Write the Unit Playbook** for hand-off to the next rotation: architecture diagram, run instructions, two known issues, one stretch goal. Use the Week 4 playbook template.
- **Run a 30-min teach-back** with two non-attendees in the next 30 days — show the container, walk the architecture, surface their first three questions to the program lead. Capstone certificate issues once playbook + teach-back evidence land.

DECK MATERIALS

The four files instructors typically share at the start of the eight-hour capstone.

Week 6 Student Handout

When: Drop in chat at session start so students have the day's prompts and homework on one page.

<https://jeranaias.github.io/ExpertDrivenDevelopment/handouts/week-6-fullstack.html>

Week 6 Facilitator Pack (HTML)

When: Send to a co-facilitator running the live debugging segments alongside you.

<https://jeranaias.github.io/ExpertDrivenDevelopment/facilitator/week-6-fullstack.html>

Week 6 Facilitator Pack (PDF)

When: Email-friendly one-pager.

<https://jeranaias.github.io/ExpertDrivenDevelopment/facilitator/pdf/week-6-fullstack.pdf>

Week 6 PowerPoint deck

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

<https://jeranaias.github.io/ExpertDrivenDevelopment/pptx/week-6-fullstack.pptx>

Week 6 slide deck (HTML)

When: For students who missed the live capstone.

<https://jeranaias.github.io/ExpertDrivenDevelopment/decks/week-6-fullstack.html>

Week 6 Instructor Cheat Sheet

When: Single-page cheat sheet with the key prompts, common failure modes, and recovery scripts you'll reach for during the capstone.

<https://jeranaias.github.io/ExpertDrivenDevelopment/decks/week-6-cheatsheet.html>

EDD REFERENCE DOCS (CAPSTONE)

The Week 6 capstone calls back to the SOP, the prompt library, and the templates — share these on demand as students hit the relevant module.

Prompt Library

When: Reach-for-this when students get stuck on a prompt during the live builds.

<https://jeranaias.github.io/ExpertDrivenDevelopment/resources/prompt-library.html>

Agent Setup guide

When: Useful when students hit auth/integration questions during the chat / auth / integrations modules.

<https://jeranaias.github.io/ExpertDrivenDevelopment/resources/agent-setup.html>

EDD Templates index

When: The full template set — problem definition, dev journal, QA checklist, doc package, tool registry — for the homework / hand-off.

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

EDD SOP (PDF)

When: The canonical SOP — the “what does good look like” document for the capstone's QA passes.

https://jeranaias.github.io/ExpertDrivenDevelopment/pdf/SOP_Expert_Driven_Development_v5.pdf

Training Curricula (PDF)

When: For students who want the full program-level reference document going home.

https://jeranaias.github.io/ExpertDrivenDevelopment/pdf/TRAINING_CURRICULA_v5.pdf

CLOSE-OUT

The final “What to do after class” slide cues you to drop the certificate page link and the cohort progress page in chat.

Course completion certificate

When: Slide cues: “Drop the certificate page link in the channel.” Students fill it in to generate their EDD completion certificate.

<https://jeranaias.github.io/ExpertDrivenDevelopment/courses/certificate.html>

Cohort progress tracker

When: For supervisors and the cohort lead — share so the room can see where their section stands.

<https://jeranaias.github.io/ExpertDrivenDevelopment/courses/progress.html>