6 COURSE 6 · STUDENT HANDOUT

# Full-Stack Capstone

Eight build cycles. Light four architecture layers. Ship one deployed container by 1700 — React + Vite + TypeScript + Go + Docker.

**DURATION** 8 hr (2 breaks)
**AUDIENCE** Capstone candidates
**PREREQ** Courses 1–4 + proposal

| 0:00–0:50 | 0:50–1:50 | 1:50–2:50 | 2:50–3:00 | 3:00–4:00 | 4:00–5:00 | 5:00–5:30 | 5:30–6:30 | 6:30–7:15 | 7:15–7:45 | 7:45–8:00 |
|---|---|---|---|---|---|---|---|---|---|---|
| **M1–2 Frontier & Setup** 50 MIN · FRAMING | **M3 Backend (L2 lit)** 60 MIN · BUILD | **M4 Data Layer (L3 lit)** 60 MIN · BUILD | **Break** 10 MIN | **M5 Frontend (L1 lit)** 60 MIN · BUILD | **M6 Pages & Nav** 60 MIN · BUILD | **Lunch** 30 MIN | **M7 AI Chat + Tools** 60 MIN · BUILD | **M8–9 Auth + Integrations** 45 MIN · BUILD | **M10 Deploy (L4 lit)** 30 MIN · BUILD | **M11 Assess & Close** 15 MIN |

## BRING WITH YOU

**Done before you walk in — setup is the dullest moment of the day**

- **Approved capstone proposal.** One paragraph, accepted by the program lead. No proposal, no seat.
- **Five tools installed and signed in:** a code editor (VS Code or similar), Node + Vite, the Go toolchain, Docker Desktop, and your AI tool of choice (GenAI.mil preferred).
- **Admin rights on your laptop** (Docker requires it). If you don't, defer Docker to Module 10 and verify with the instructor.
- **Courses 1–4 complete** with at least one shipped tool, your frontier map, and your workflow playbook. We reference all three today.
- **Headphones, charger, water.** Eight hours, ten modules, one deployed container. Pace yourself.

## WHAT YOU'LL BE ABLE TO DO

**By the end of the day**

- Direct AI to build all four layers of a real application — frontend, backend, data, deploy.
- Recover from any single-layer failure without losing the rest of the build.
- Wire an LLM into your own backend with tool use that mutates real data.
- Package and run your app as a container on any machine.
- Hand the container to a teammate with a one-page README.

## KEY TERMS

**The vocabulary you'll hear today**

- **Full stack**
  All four layers in one app: *frontend (L1) · backend (L2) · data (L3) · deploy (L4).* One person directs all four with AI.
- **Light a layer**
  Make one architecture box solid red on the diagram — verified working, not just scaffolded.
- **Build cycle**
  Framing → build → module-complete check → back to deck. Repeated eight times. Same shape every time.
- **Module-complete check**
  A short list at the end of each module that proves the layer is lit before you advance.
- **Tool use (AI chat)**
  Module 7 pattern: the chat model calls your backend functions to read and write the app's data — not just generate text.
- **Container**
  Module 10 deliverable. Your app, packaged, runnable on any machine with Docker. The proof you shipped.

## HOMEWORK

**Capstone close-out · within 14 days**

- **Push your container** to the program registry with a one-page README: what it does, who it's for, how to run it, known frontier issues.
- **Demo it once** to your section — in person, not in chat. Capture the questions you couldn't answer and follow up.
- **File one frontier-map entry** per layer (L1–L4) covering what AI handled well, what it didn't, and one re-test trigger for next quarter.
- **Pick the next two Marines you'll teach this to.** Calendar the first session before you log off today.
- **Submit the rubric self-assessment** (six categories) for instructor-track consideration.

## EXERCISES IN CLASS

**Eight build cycles — what each lights and what "done" looks like**

**M3 · Backend from scratch (60 min) — lights L2.** A Go HTTP server returning a hardcoded array. *Done:* `curl` against your endpoint returns valid JSON; backend box solid on the diagram.

**M4 · Data layer (60 min) — lights L3.** Replace the hardcoded array with a real store behind a clean interface. *Done:* the same endpoint now reads and writes persistent data; data box solid.

**M5 · Frontend from scratch (60 min) — lights L1.** React + Vite + TypeScript view that calls your backend. First end-to-end view. *Done:* the page in the browser shows live data from your backend; frontend box solid — full stack end-to-end.

**M6 · Pages & navigation (60 min) — fleshes L1.** Multi-page app with routing. *Done:* the prototype is gone; you have a real application skeleton.

**M7 · AI chat with tool use (60 min).** An LLM-backed chat that calls your backend functions to read and write app data. *Done:* a user sentence in chat causes a real change in your store, verified in the UI.

**M8 · Authentication (30 min).** Sign-in flow protecting your endpoints. *Done:* unauthenticated requests rejected; authenticated requests succeed.

**M9 · External integrations (15–30 min — first to compress if running long).** One outbound call to a real service or API. *Done:* one round-trip with a real response; everyone still ships a container.

**M10 · Deploy in a container (30 min) — lights L4.** One Dockerfile, one `docker run`, your full app live in a container. *Done:* 4/4 layers lit on the diagram; you can hand the container to a teammate and they can run it.

**M11 · Assessment (15 min).** Walk through the rubric, name your one biggest frontier surprise, name the one Marine you'll teach this to first.

> **ANCHOR PHRASE** You will not learn to code today. You will learn to direct a build.

## STUCK? COMPARE AGAINST THE ANSWER KEY

Open **`heywood-inventory/`** only *after* the 3-Minute Rule fires twice on the same prompt — never before. Full guide: **`heywood-inventory/docs/STUDENT_GUIDE.md`**.