

If you get lost or fall behind — just ask Claude directly.

Claude always knows where you are. These three phrases get you back on track instantly:

"Claude, where am I?"

"Catch me up."

"Explain that in plain English."

TODAY'S MAP

Command	What you're doing	~Time
<code>/introduction</code>	You made it — the whole series is done Resources + how to stay connected	~3 min
<code>/refresher</code>	The greatest hits from every week, with a quick "how" for each	~4 min
<code>/capstone-instructions</code>	The rules of your build — read these before you start	~5 min
<code>/final-project</code>	Your 90 minutes: plan → build → iterate (twice), then ship & present	~90 min
<code>/conclusion</code>	Your certificate, your portfolio, and what's next	~3 min

HOW TO START

Unzip your **workshop-5-capstone** folder, then launch Claude Code *inside that folder* — that's where today's commands live. Then your instructor will call each command in order. You can work in the terminal or in VS Code, whichever you prefer.

IN THE TERMINAL

Mac: right-click the folder → "New Terminal at Folder." **Windows:** right-click inside the folder → "Open in Terminal." Then type:

```
c laude
```

Or `cd` into the folder first, then run `c laude`. Drag the folder into the terminal after `cd` to auto-fill the path.

IN VS CODE

File → Open Folder → your workshop-5-capstone folder. Open the Claude panel (the spark icon), then type `/introduction`.

VS Code has a built-in terminal (`Ctrl` + backtick) — you never need a second window.

You don't need a `CLAUDE.md` yet — you'll create one for *your* project during `/final-project`, right after your plan is approved. That's a habit worth keeping for every project from now on.

COMMAND-BY-COMMAND REFERENCE

COMMAND 1

You Finished the Series

/introduction

- Every workshop, all the reference material, in one place: buildbeautifully.lovable.app/event-resources
- Follow along — Instagram [@build_beautifully_1](#) and the Build Beautifully [LinkedIn](#) page.
- Know a woman who'd love this? Send her the sign-up form and bring her to the next one.

COMMAND 2

The Refresher

/refresher

- A fast memory-jog of the whole series before you build.
- **WS1**: plan mode, plan-build-iterate. **WS2**: agents, APIs (+ fal.ai). **WS3**: skills, plugins. **WS4**: MCPs, the Gauntlet Loop.
- Every one comes with a one-line "how" — all collected in the [Capstone Cheat Sheet](#) below.

COMMAND 3

The Rules

/capstone-instructions

- **90 minutes** to build your own app, then **present to the room**.
- **Present** your build to the room. Deploy + pass the test → a **certificate**.
- Must include **≥ 2 of**: API, agent, skill, plugin, MCP. Run **plan-build-iterate twice**. Get feedback in round two. Push to GitHub + deploy on Vercel.

COMMAND 4

Your 90 Minutes

/final-project

- Switch to **plan mode** and use Plan → Build → Iterate. **Plan before you build — always.**
- Right after your plan is approved, **create a CLAUDE.md** for your project.
- Two full build-and-iterate passes. Get **feedback during the second one's iterate step**.
- **Ship it** — push to GitHub, deploy on Vercel — then present.

COMMAND 5

Certificate & What's Next

/conclusion

- **Certificate**: go to build-beautifully-certification.vercel.app, submit your live Vercel link + take the short test. Agents grade your build and test — pass and you get a shareable certificate.
- Update your **"My AI Builds"** portfolio (from WS1) — you should have **4+ builds** now.
- Add your portfolio to [LinkedIn](#), and bring someone to the next event.

REMEMBER

The One Rule

- **Plan before you build.** No code until you have a plan you've read and approved.
- The first version is never the last — **iterate**.
- Never paste an API key into the chat. Put it in the file yourself.

CAPSTONE CHEAT SHEET — HOW TO BUILD EACH THING

Your project needs **at least two** of these five. You already know how to do every one — here's the one-liner for each.

Thing	What it is	How to add it
API	One program asking another to do something.	Sign up (e.g. fal.ai for audio/images), grab your key, ask me to use it — you paste the key <i>into the file</i> , never the chat.
Agent	A helper with one job and its own instructions.	Ask me: <i>"make me an agent that [does the thing]."</i>
Skill	A saved set of instructions for something you do often.	Ask me: <i>"create a skill for [task]."</i>
Plugin	An installable bundle (skills, tools, even MCPs together).	Add with <code>/plugin</code> from a marketplace, or just ask me to set it up.
MCP	A one-line install that gives me a new ability (e.g. Playwright = eyes in a browser).	Tell me the ability you want; I'll hand you the one line to paste in your terminal.

Work like a pro (the WS4 habits)

Save your context	Type <code>/context</code> to check how full I am. Around 40%+ : ask me to <i>"summarize what we've built and add decisions to my CLAUDE.md,"</i> then type <code>/clear</code> , and I re-read it. Write it down → clear → carry on. Your files are safe.
Pick your model	Type <code>/model</code> . Opus for complex, heavy thinking · Sonnet for lighter work and to stretch Pro-plan usage (lean on agents too) · Fable for stubborn bugs.
Mind your mode	Plan (think first) · Edit-with-permissions (I ask before each step — your default today) · Auto (I just go). Don't get stuck in auto by accident.
Debug with Fable	Hit a bug you can't shake? Switch to Fable (<code>/model</code>), ask it to look over your code and catch bugs. You approve every change.

Before you present — the checklist

- | | |
|--|--|
| ☐ At least 2 of {API, agent, skill, plugin, MCP} | ☐ Pushed to GitHub + deployed on Vercel |
| ☐ Plan → Build → Iterate, twice | ☐ Presented your build to the room |
| ☐ Feedback from a TA / another woman in round 2's iterate | ☐ Certificate: submitted Vercel link + test |
| ☐ Created a CLAUDE.md for your project | ☐ Updated My AI Builds + added to LinkedIn |

COMMON TROUBLESHOOTING

What's happening	Fix
The slash commands (<code>/</code> <code>introduction</code> , etc.) don't show up	Claude Code isn't in the right folder. Launch it <i>inside</i> your unzipped <code>workshop-5-capstone</code> folder (or open that folder in VS Code), then try again.
Claude feels forgetful or confused mid-build	Type <code>/context</code> . Over ~40%? Ask it to summarize what you've built into your <code>CLAUDE.md</code> , then <code>/clear</code> and it re-reads. Your files are safe — only the conversation resets.
A bug you can't shake	Switch to Fable (<code>/model</code>) and ask it to look over your code and catch bugs. Read what it finds and approve each change yourself.
Running low on usage (Pro plan)	Switch to Sonnet (<code>/model</code>) and lean on agents for the heavy lifting to stretch it further.
Deploy trouble on Vercel	Connect your GitHub repo to Vercel — same flow as your Workshop 2 game. Ask Claude for the exact steps for your project.
Claude asks for an API key	Never paste a key into the chat. Put it in the file yourself. If you already pasted one, delete that message and rotate the key.

GLOSSARY

Word	Plain English
API	One program asking another to do something (e.g. generate audio).
Agent	A helper with one job and its own instructions.
Skill	Saved instructions Claude uses for a task you repeat.
Plugin	A bundle of skills, commands, and MCPs installed together.
MCP	A one-line install that gives Claude a new ability.
Gauntlet Loop	Builders plus separate critics, run until the work is genuinely good.
Plan mode	Claude thinks and proposes but can't touch files. Toggle with Shift+Tab.
CLAUDE.md	Your project's memory file — Claude reads it first every time.

Context window

KEY LINKS

Resources	buildbeautifully.lovable.app/event-resources
Certificate	build-beautifully-certification.vercel.app
Website	buildbeautifully.lovable.app
Instagram	@build_beautifully_1
LinkedIn	Build Beautifully
Invite a friend	tally.so/r/b56GAL

Your certificate is real and shareable — deploy your app, submit the link, pass the test, and put it on your LinkedIn. You earned it.

Word	Plain English
	How much of your conversation Claude can hold. Check with <code>/ context</code> .
Deploy / Vercel	Putting your app live on the internet at a real link.
GitHub	Cloud storage for your code — a folder in the sky that saves every version.