

Build Beautifully

buildbeautifully.lovable.app

Session 4 · Build Like a Pro · Advanced vibe coding

If you get lost, you are not behind.

Type any of these to Claude. She'll pick you back up exactly where you are.

Where am I in the course?

Catch me up.

Explain that in plain English.

Give me a simpler version.

Undo that.

STARTING UP

Everything happens in VS Code

- Open **VS Code**
- **File** → **Open Folder** → the workshop folder
- Click the **orange Claude spark, top right**
- Type `/model` → pick **Sonnet**
- Type `/introduction`

If VS Code opens its own chat panel, close it — that isn't Claude.

Two places to type

The chat. You type English. Claude does the work.

The terminal. You type commands. Your computer does the work, and Claude isn't involved.

The terminal is built into VS Code: **Ctrl + backtick** (above Tab, left of the 1), or **Terminal** → **New Terminal**.

Claude tells you which one, every time.

TODAY'S MAP — YOUR INSTRUCTOR CALLS EACH ONE

4 MIN

Introduction

`/introduction`

Two new concepts and one new tool.

16 MIN

Setup

`/step-1-setup`

Your language, name and mascot. The workspace tour. Your first MCP.

7 MIN

CLAUDE.md

`/step-2-claude-md`

Your colors and feeling, written into the file that holds your instructions.

20 MIN

Plan & Build

`/step-3-plan-and-build`

Plan → build → iterate. Your app comes to life.

34 MIN

The Gauntlet

`/step-4-gauntlet`

Live preview, then a 20-minute capped loop of builders and critics.

9 MIN

Make It Speak

`/step-5-audio`

Real pronunciation audio, through an API call you make yourself.

8 MIN

Ship It

`/step-6-ship`

Save to GitHub. Connect to Vercel at home.

REMEMBER

This is a phone app

Everything gets built and judged at **phone width**. Keep your preview window narrow.

THE TWO NEW WORDS

MCP

A plug-in pack of new abilities for Claude. **A USB port for Claude.**

You install it once, with one line. You never write code to use it.

Today's is **Playwright** — it gives Claude **eyes in a real browser**. She can open your app, screenshot it, read its errors, and click through it.

The Gauntlet Loop

When Claude checks her own work she's grading her own homework, and she's a generous grader.

So we split the job. Claude breaks your app into pieces. For each piece: one agent builds it, and a **completely separate agent judges it** — one that never saw the builder's reasoning.

Every piece loops until its own judge is happy. **There's no set number of rounds.**

COMMANDS YOU'LL TYPE

In the terminal panel	What it does
<pre>claude mcp add --scope user playwright -- npx -y @playwright/mcp@latest -- output-dir ~/playwright-output</pre>	Installs your browser MCP, everywhere on your machine. The last bit keeps screenshots out of your project.
<pre>claude mcp list</pre>	Checks it worked. Look for Connected . If it says failed, wait 5 seconds and run it again — it's still downloading.
<pre>npx live-server</pre>	Opens your app in a browser that refreshes itself every time a file changes.
<pre>npm init -y</pre>	Sets up a project file. Always before installing anything.
<pre>npm install @fal-ai/client dotenv</pre>	The two packages the audio needs.
<pre>node generate-audio.mjs</pre>	Runs the script that makes your audio files.

Stop the live preview before you generate audio. Click into that terminal and press **Ctrl+C**. A dozen new sound files means a dozen page reloads, which breaks playback. Start it again with `npx live-server` afterwards.

VS CODE — THREE THINGS THAT MATTER

1 · The file tree

The list down the left side — every file and folder in your project. When Claude makes something, you watch it appear there.

2 · Diffs

A **diff** is a before-and-after view of the same file. Current version one side, new version the other, changed lines highlighted. **Green** is being added, **red** is being removed.

3 · Rewind

Hover over any message you sent and click rewind. Your files go back to how they were at that point.

Why Sonnet?

The Gauntlet spawns a lot of agents at once, each carrying its own context. **Sonnet** is built for that and keeps you comfortably inside your plan. Check yours any time with `/usage`.

CLAUDE.MD — WHERE YOUR INSTRUCTIONS LIVE

The one sentence to take home: things you only say to Claude can disappear. Things written in **CLAUDE.md** come back — she re-reads it off the disk.

Two rules for writing one

Keep it under a page. If it gets long, Claude follows it less closely.

Write lines she can check. "Buttons have a pressed state" beats "make it look polished" — she can look at the screen and tell you whether she did it.

Making your own at home

Ask for it: "create a *CLAUDE.md* for this project", or type `/init` for a first draft.

1. Plan → 2. Write your CLAUDE.md → 3. Build → 4. Iterate. Correcting the same thing twice? It belongs in the file.

WHERE SECRETS LIVE

Never paste an API key into a chat window. Not to Claude, not to anyone, not "just to check". It goes in a file on your own computer and nowhere else. A key is a password with a credit card attached.

The two files

```
.env      → FAL_KEY=your_key_here
.gitignore → .env
```

`.gitignore` is a list of things that never leave your laptop. Your key lives in `.env`, `.env` is listed in `.gitignore`, so when your app goes online your key stays home.

The gotcha nobody warns you about

Creating `.env` isn't enough — **Node doesn't read it by itself.** Your script needs this as its very first line:

```
import 'dotenv/config'
```

If your API calls fail with an auth error and you're sure the key is right, **this is almost always why.**

TROUBLESHOOTING

What you see	What to do
<code>Failed to connect</code> on the MCP	Wait 5 seconds, run <code>claude mcp list</code> again. It's still downloading.
<code>claude: command not found</code>	The VS Code extension doesn't add <code>claude</code> to your terminal — separate install. Mac: <code>curl -fsSL https://claude.ai/install.sh bash</code>
Can't find Claude in VS Code	Orange spark, top right . If VS Code's own chat opens instead, close it.
Your app flickers and reloads constantly	Screenshots are landing in your project. The MCP needs the <code>--output-dir</code> part — reinstall it with that flag.
The Gauntlet won't stop	It has no off switch, by design. Close the Claude tab , then reopen and type <code>/resume</code> . Nothing is lost.
Audio fails with an auth error	Check <code>import 'dotenv/config'</code> is the first line of the script. Then check <code>.env</code> for stray quotes or spaces.
<code>Cannot find module</code>	You skipped <code>npm init -y</code> , or you're in the wrong folder. Check with <code>pwd</code> .
Claude forgot your brand	That's what <i>CLAUDE.md</i> is for. Point her at it and watch it snap back.

GLOSSARY

MCP	A plug-in pack of new abilities for Claude.	Agent	An AI worker that uses tools and decides what to do next.
Gauntlet Loop	One builds, one judges, for every piece — until it beats the bar.	API	One program asking another program to do something.
IDE	The editor professionals live in. Yours is VS Code.	API key	A password that belongs to a program, not to you.
File tree	The list of your files down the left side.	.env	The file your secrets live in. Never leaves your laptop.
Diff	Before and after of the same file, side by side.	.gitignore	The list of things that never get uploaded.

WHAT YOU ACTUALLY LEARNED

1 · You can give Claude abilities she doesn't ship with

There are hundreds of MCPs and they all install the same way yours did. One line, once.

2 · Written down beats said out loud

Your CLAUDE.md held your brand through an hour of building — and through closing the whole conversation and reopening it.

3 · Secrets live in `.env`, listed in `.gitignore`

And never in a chat window. This is the habit that saves you real money one day.

4 · One builds, one judges, against a real bar

Run the Gauntlet on anything you can look at — a deck, a landing page, a cold email.

Tonight: connect your repo to **Vercel** — same as your Workshop 2 game, about two minutes. Then open it on your phone, which is what it was built for, and send the link to one person.

Want the deep version? The Gauntlet Loop came from Matt Shumer. The original write-up and prompt are at [somethingbig.ai/gauntlet-loop](#), and everything about how it works is in [reference/02-the-gauntlet-loop.md](#) in your workshop folder.