
NO BS AI

CLASS MATERIALS

Week 4: Lab Week

(Open Workshop Session)

Pilot Class // April 2026

Instructor: RJ // Platform: Rippily

The Goblin Mining Cave

"Stop buying software.

You can build it."

-- RJ, Week 4 Lab Week

DO NOT LICK THE TRANSCRIPTS

DEEPER INTO THE MINE WE GO

SECTION 1: MCPS VS. CLAUDE IN CHROME

Two Different Tools. Two Different Jobs.

MCPs and Claude in Chrome are not the same thing — and knowing the difference will save you real money.

MCPs connect apps and services in the background. They're efficient, they don't eat much credit, and you don't have to babysit them. Claude talks directly to the tool through a pipe connection.

Claude in Chrome is different. It opens a labeled tab group and works inside your actual browser — watching, clicking, reading the interface the way a human would. It's powerful when you need it, but it is a token burner. Use it when the task is genuinely visual and you need to see it happen.

"Sometimes it's cool to do through MCP. And sometimes when the project is pretty visual, it's nice to have Claude in Chrome. Know which one you're reaching for."

Setting Up Claude in Chrome

To enable it: **Settings** → **Allow Extension**

A few safety notes:

- **Blocked sites:** Add any sites you'd rather Claude not touch — your bank, anything you want kept private. It's there for peace of mind.
- **Prompt injection warning:** Bad actors sometimes hide instructions in white text on white backgrounds. If a site looks sketchy, stop everything.
- **Financial sites:** Never use Claude in Chrome on banking or financial sites. Period.

When Claude Needs You

Even with Claude in Chrome running, there will be moments when it says: *"I can't do this part — you need to."* It'll tell you exactly what to click or do. That's normal. It's not broken. It's just hitting the edge of what it can see.

SECTION 2: ONE THREAD. ONE JOB.

Why Your Thread Length Costs You Money

Here's something most people don't realize: **every single exchange between you and Claude re-reads the entire conversation from the beginning.** Every time. Even if you're asking something simple.

You can't stop it. That's just how it works right now. So what do you manage instead?

Your threads.

One topic. One session. When the work shifts, start fresh. This is the single most effective thing you can do to keep your costs down.

"One thread, one job. Because every time anything processes — any exchange between you and the AI — it grabs all the context again and eats it all again."

The Interrupt Button

This is your emergency brake, and it is right there on screen. The moment you think "no, that's wrong" — hit it. Don't wait for Claude to finish. Don't let it write four more paragraphs you're going to delete.

When to interrupt:

- Claude's going the wrong direction
- You said "quick question" and it's writing an essay
- You asked for the price of cheese and it's giving you the history of Velveeta
- It's showing signs of tunnel vision — locked onto one solution, can't see anything else

Tunnel Vision and the Goblin Goggles

A lot of RJ's goblin illustrations have goggles. That's intentional.

"Goblins will get tunnel vision. They want to solve your problem so bad — bam. We're in a mode where we can only see the end result and we're not keeping anything else in mind. That's a warning sign."

When you see tunnel vision, don't try to redirect the same thread. Start a new one.

Billing Tips

- **Ran out at a bad moment?** Make a one-time credit purchase. Don't auto-reload unless you truly mean to.
- **Model matching still applies:** Haiku for planning, Sonnet for building, Opus for when you need it right the first time.
- **\$200/month:** Yes, RJ's monthly Claude bill is \$200. She uses it all day and night and it is completely justified. Your number will be different. That's fine.

SECTION 3: THE GLAMOUR ENGINE

A Story About Seeing Something and Building It

RJ gets emails every day about new AI products. One of them was a piece of agency software — product images, writing style, copy all in one pipeline. Built for marketing agencies. \$200+ a month. Not aimed at entrepreneurs. Not designed for her use case.

But she saw what it was *trying* to do.

She asked Claude: *"Could you take a look at this agency software and tell me if I could build something similar for First Draft Ninja — for emails, not products?"*

Two days later, she had the Glamour Engine.

What the Glamour Engine Does

Before it existed, here's what generating a single email looked like:

1. First Draft Ninja writes a draft
2. RJ edits the copy
3. Asks Claude to write a detailed image prompt for position 1
4. Goes to Gemini, generates the image
5. Brings image back to Claude for tweaks
6. Takes the image to Canva — boosts contrast, vibrancy, sharpness for mobile
7. Places the image in the email
8. Repeats for image 2
9. Assembles everything in GHM manually

That process took longer than writing the email itself.

Now, with the Glamour Engine:

1. Paste the finished email in
2. Pick a subject line (or ask for more options)
3. Two images are auto-generated via Gemini through Replicate — about 3-4 cents each
4. Pick captions for each image
5. Pick a sign-off line
6. Coffee recipe is selected automatically
7. Hit approve → polished draft lands in GHM, ready to review

14 workflows. Two days to build. Zero dollars in ongoing software cost.

"This beats the two, three-hour conversation that I have with Claude every day."

The Build-For-Yourself-First Protocol

This is the lesson behind the story — not the Glamour Engine specifically, but the method:

1. See something cool
2. Ask Claude: "Could I build something like this for my situation?"
3. Build it for yourself first
4. Use the hell out of it
5. Standardize it
6. Externalize it when it's working

"Please don't buy any more software. With Claude, Airtable, N8N, and Cloudflare — that's all you need to build anything. A webpage, a chatbot, a full pipeline. Build it. It'll work better for you because it's yours."

Ideas come from anywhere. Other people's software, competitors' products, things you see in videos — all of it is a source of "what if I built my own version of that?"

SECTION 4: THERE'S PROBABLY AN MCP FOR THAT

Finding MCPs

If you use a piece of software regularly, there's a decent chance someone has built an MCP for it — either officially (the company built it) or by the community.

How to find them:

1. Search "[software name] MCP" in your browser or ask Claude directly
2. Look for trophy icons and high GitHub star counts — these indicate the community rates them highly
3. Check whether it requires a paid tier of the software (that's the main catch)

Explored Live This Session

World Anvil (for writers with large lore worlds)

A student wanted to connect Claude to her writing partner's lore world — a massive collection of characters, locations, events, and backstory stored across Google Drive.

The short answer: **skip Google Drive. Go straight to World Anvil.**

Here's why: computers don't read like humans do. A human can scan a chunk of text and get the gist. A computer would rather have a spreadsheet. World Anvil already structures your lore as organized, cross-linked data — characters linked to events, events linked to locations. When you connect Claude via the World Anvil MCP, it doesn't have to dig through folders. It can retrieve exactly what it needs.

"If you are pulling from World Anvil, you do not have to worry about documents and headers and organization. Organize it visually for your people. World Anvil translates that information into pieces of data."

DaVinci Resolve (for video production)

Community-built. The company hasn't made an official one — but the community did, and it's highly rated.

- 342 granular actions + 26 compound actions
- Can write scripts, order shots, automate the production line
- Requires DaVinci Resolve Studio (paid, ~\$295 one-time)
- High star count on GitHub — well maintained

Hailuo AI (for video and audio generation)

Official MCP — maintained by the people who make the software.

Full suite: text-to-video, image-to-video, text-to-speech, voice cloning, image generation, music generation.

Good choice if you're building video workflows and want to automate the generation step.

SECTION 5: N8N IS YOUR PIPELINE

Why N8N Wins

When you need to connect two pieces of software — automate an order, send data somewhere, trigger an action — you need a pipeline tool. Here's the landscape:

Tool	Verdict
Zapier	Gouges you. Skip it.
Make.com	Horrific to build in. No useful MCP. Skip it.
N8N	\$20/month. Any API. Claude builds it for you. ■

The key difference: Zapier and Make.com require apps to build dedicated connectors for their platform. If your app isn't on their list, you can't use it.

N8N just needs an API. If an app has any public API, you can connect it. That's most apps.

Claude Builds Your Workflows

Here's the real magic: with the N8N MCP connected, you describe what you want the pipeline to do and Claude builds it. You don't even need an N8N tab open. You just describe it.

"I described what I wanted to Claude. I have an MCP into N8N. Claude makes it. I didn't have to look at it — it built it and tested it for me."

The Glamour Engine — 14 workflows — was built this way.

The API Rule

For any integration to work, **both sides need an API.**

- If your fulfillment company doesn't have a public API, write them and ask for a private one. 20% chance it works. Worth the email.
- If neither side has an API, you're stuck — no pipeline tool will help.

SECTION 6: SHOOT HOLES IN YOUR THEORY

When Claude Is Too Optimistic

Claude is, by design, helpful and positive. This is mostly great. But sometimes you need a reality check — especially before you commit time to building something.

The fix is simple: **flip the question.**

Instead of "will this work?" ask:

"Give me all the reasons this WON'T work."

Or:

"Shoot holes in this theory, please."

Or:

"One more round of deep analysis — I want to make sure before I commit to this."

Why this works: Going in the opposite direction reveals holes that the happy path conceals. Claude's confidence can become your assumption if you're not careful — and assumptions are where projects fall apart.

The Trainer Mike Story

You're the creator. Not the manager.

RJ had a meeting with a trainer named Mike. About 90 minutes long — gathered colors, fonts, images, preferences, the feeling he wanted people to have when they visited his site.

After the meeting: one hour of analysis and prompting. 30 minutes of tweaks.

Result: **three-page website, fully built and working.**

What would that have taken two years ago? Two and a half weeks.

The F12 key shows all the HTML. RJ didn't write a line of it. She directed it.

"Something like this — I know what it looks like underneath. And if we can touch those things, that throws you back in the seat of the creator instead of the manager of everything."

The question that matters: *What do you want your people to feel?* Start there. The tools will figure out how to get there.
