

NO BS AI

CLASS MATERIALS

Week 6: Lab Day

(Open Workshop Session)

Pilot Class // April/May 2026

Instructor: RJ // Platform: Rippily

The Goblin Mining Cave // Print Edition

SECTION 1: WELCOME — CATCHING UP IN THE CAVE

Who's Here

RJ opened with a check-in — mom had just left that morning. A regular client rescheduled, so RJ was there for the full session.

Students present: Laurel (brief — left early), Delaina, Jamie Buckley, Hermi, Bez.

"I'm here for questions, problems, things you can't solve, things you did solve and you love and you just want to show everybody. I'm here for all of it."

Jamie's Kindlepreneur Launch

Jamie dropped major news: a course she'd spent a year building launched that morning. **Kindlepreneur** — one of the largest writer-assisting platforms on the internet — emailed it to 150,000 writers, with another 150,000 going out at 2 PM.

The topic: migrating to Substack, which Jamie uses daily.

How it happened: Jamie's connection to Kindlepreneur's editor in chief goes back to 2014. He needed someone who worked on Substack, was honest, and obsessed with making things that help people. He called Jamie directly.

"I got asked because their editor in chief has known me since 2014. He says: I know you'll be honest. I know you'll be obsessed with making things that help people. So if I tell you what we need, would you make that for me?"

"I was like: of course I would."

The kicker: they didn't know she had a world building course until after the Substack course wrapped. Now they want that one too.

RJ's note: Don't give the world building course away for free. You've been selling it for years. Don't change that now.

Hockey, Mammoth Zambonis, and Making It Here on Backup Alarm

Delaina was running on a backup alarm — Golden Knights went to double overtime the night before (they won). Laurel was rooting for Utah's team, the Mammoth, which has a Zamboni dressed like a woolly mammoth.

"It does sound adorable. But they're an expansion team."

SECTION 2: CLAUDE DESIGN — PROTOTYPE BUILDING IN TWO HOURS

The Trainer Mike App

RJ has been working with her personal trainer Mike on a **progressive web app (PWA)** — an app that lives entirely on the web. No App Store, no submission hoops, nothing to download.

Background: Mike mentioned wanting an app. RJ had built roughly 13 concept screens in Canva — pre-workout, warm-up, main sets, cool-down, post-workout. Basic wireframes. Then she discovered Claude Design.

Claude Design is a prototyping tool built into Claude — available only on the **web version** (claude.ai). It is NOT in the desktop app.

"People were like, Claude Design, Claude Design. I'm like, where the hell is it? It's on the web version, not the desktop. Don't know why."

Find it: on **claude.ai**, look for the small palette icon in the toolbar.

What Got Built in 2 Hours

RJ fed Claude Design:

- Several meeting transcripts with Mike
- The Canva wireframes

Result: a full **click-through prototype** with:

- Today's workout screen
- Pre-workout, warm-up, main sets, cool-down sections
- "Ready to Explore" (green) and "Work in Progress" tags
- Main set view with couplets (paired exercises)
- Notes fields at three levels: individual exercise, circuit, and full session
- Three visual themes: **Bold**, **Wellness**, and **Athletic** (the athletic theme will hurt your eyes — you've been warned)

RJ also fed Mike's actual exercise database — a snippet of real equipment and exercise data — so the prototype runs on real information, not placeholder text.

"I made him a working prototype that he can click through in two hours."

Why a First Draft Wins Every Time

RJ knows something about humans and design work:

"They stare at a blank page and go: what is going on? It's so hard for a lot of people to come up with something."

People can't say what they want. They CAN say what they don't want. A first draft — even a rough one — unlocks the conversation. They tell you what to cut and keep, and that's how you find out what they actually want.

The same principle works for book covers, widget layouts, app screens, anything with design involved.

The director shift:

We go from being a one-person band — storyteller, sound effects, instruments all tied to you, producing every single aspect — to sitting in the middle row as the **director and curator**.

"Love this. Tweak this. Hate this — throw it out entirely, start again. We are the director and the goblins are producing."

One rule that never changes: Never automate what you love. Always automate what you don't love.

What Comes Next for the App

Mike's dad is a software engineer who has built all the backend. When the design is close, RJ will press a button, send the HTML and assets, and the two halves connect. The prototype gives his dad exactly the spec he needs.

SECTION 3: SKILLS — WHAT THEY ARE AND HOW TO BUILD THEM

What a Skill Is

A skill is a **process you follow every time** — a checklist that doesn't change. You invoke it with a forward slash:

```
/skill-name
```

In Claude Code, typing `/` surfaces your full list of available skills. Some are built-in. Some you build. The user-built ones show *(user)* next to them.

"Skills are a certain process that you follow every time that you want the computer to follow instead of you having to manage the process."

The Wrap-Up Skill — Live Demo

RJ ran `/wrap-up` mid-session to demonstrate.

What RJ's wrap-up does:

1. Scans the entire conversation
2. Writes a session summary
3. Logs lessons learned to a persistent memory file
4. Writes updates to the relevant project changelog
5. Saves tasks and ideas to Airtable
6. Creates a detailed handoff document for the next session
7. Outputs a link to the handoff doc + a copy-paste intro for the next thread

"When you create a skill like this wrap-up, you're creating a self-healing system. You're creating something that makes you better every time you use it."

The system also makes the AI better — every session it learns what worked, what failed, and what to carry forward.

How RJ Builds Skills

1. Go through a process
2. Mid-process, notice: "I never want to repeat these steps manually again"
3. Finish the process completely
4. Say to Claude: **"How about you create a skill for that?"**
5. Claude asks any clarifying questions and writes the skill file

That's it.

"It is not: write out every step. It is: any time you're doing something in Claude and it's annoying and it's the same process every time — finish it and say: Claude, make a skill out of this."

Trigger Phrases as Skill Launchers

You can also program a specific phrase to trigger a skill automatically. RJ uses phrases from her favorite books and shows.

Example: Typing **"We have to go back"** (from LOST) — with several A's — triggers a transcript processing skill. Claude hears that phrase and:

- Processes the transcript
- Files everything where it belongs
- This week: automatically sends the session summary to the client

"I don't have to mess with it. I don't have to remember to do it. I don't have to feel terrible about not doing it because it didn't cross my mind. I don't have to do anything."

SECTION 4: CARRY-ON — BUILDING A SKILL LIVE IN CLASS

The Problem It Solves

Sometimes you're in the middle of an idea and you run out of thread room. You don't need the full wrap-up ceremony — that's for when real work is done and you want the complete cleanup. You just need to pass the baton cleanly and keep going.

Building It Live

RJ asked Claude out loud:

"I want to write a secondary wrap-up skill that only pays attention to: summarizing the thread I just worked on, writing a handoff document for the next thread, and creating an intro to the next thread so I can just keep on with what I was doing. Can you help me with that?"

Three things only:

1. Summarize the conversation
2. Write a handoff document
3. Write an intro for the next thread

Claude wrote the skill. The `/carry-on` command was live within minutes.

"Done. Carry on is live."

Carry-On vs. Wrap-Up

	<code>/carry-on</code>	<code>/wrap-up</code>
Use when	Mid-momentum, just need to pass the baton	Real work is done, full cleanup needed

Does	Summarize + handoff doc + next thread intro	All of the above + memory, changelog, Airtable, tasks
Best for	Anyone in this class	Production work for clients

SECTION 5: AGENTS VS. SKILLS — THE ESSENTIAL DIFFERENCE

The Short Version

Skills = static, known path. Like making a peanut butter and jelly sandwich. There are steps, they go roughly in order, the path doesn't change.

Agents = dynamic, unknown territory. You send them out with a mission and some direction. They handle inputs that vary — different combinations, different results every time.

When to Use Each

Use a skill when: You know exactly what needs to happen, every single time.

Examples: */wrap-up*, */carry-on*, processing a transcript, sending a client email after a call, building a weekly report.

Use an agent when: Things change. Inputs vary. You're combining resources from multiple places and the combination is different each time.

Example: First Draft Ninja picks up your writing style, your audience profile, a story, and a call to action — and the combination is different for every email it builds. That's dynamic. That needs an agent.

"Agents handle dynamic. Agents handle unknown. That's the main difference."

Both Perform Actions On Your Behalf

Skills are more predictable. Agents are more powerful. Agents need clearer boundaries and a little more supervision.

"They're so cute. They just want to do things for you that solve the problem. They need a hall monitor like nobody's business."

Is There a Limit on Skills?

Not that we know of. There used to be a cap around 20 when skills first launched. It's clearly gone now.

SECTION 6: THE CURSOR STORY — A TEACHABLE MOMENT IN AI SAFETY

What the Headlines Said

"Claude Code destroyed an entire database without asking."

What Actually Happened

A company was using Claude Code through **Cursor** — a third-party coding environment that sits between the user and the AI. Cursor had too many permissions granted to it. The secondary layer was the problem, not Claude Code.

They got their data back two days later. Their system was down about a day and a half.

"Their system was down for about a day and a half. And I'm absolutely certain that when people returned to work, there was a different way of working with Cursor."

The Real Lesson

Whoever you're working with — AI or human — the permissions you give them matter.

Working direct in Claude Code means one set of rules. Adding a third-party environment means two sets of rules. That's twice the surface area for something to go wrong.

RJ's approach:

- Works directly in Claude Code (no extra layer)
- Claude is not connected to anything that touches a bank account
- When fixing production bots, creates a staging environment, tests, then tears it down
- Pushes to production only after staging validates

"Deleting it does solve the problem. I know. It also caused another one."

"I just want to write stories and draw pictures. I want to invite people into my world and have a virtual drink. I don't want to manage permissions errors."

SECTION 7: STRIPE + RESEND — TOOLS FOR MONEY AND EMAIL

The Stripe Connector

Claude has an official Stripe MCP connector — built and maintained by Stripe itself.

How to find it: Claude → Settings → Connectors → Browse → search "Stripe"

What it can do:

- Read subscriptions and payment history
- Update or cancel subscriptions
- Create customers and products
- Search the Stripe knowledge base

RJ's use case: A daily automation that scans email + Stripe, surfaces what's coming in and going out, and reports it to a dashboard.

"I have something connected to Stripe that reports when I get paid. And something connected to Gmail that reads invoices so that daily I can surface what my numbers are and let that drive what I do next."

Why this matters for creative entrepreneurs: If money tracking is not your jam, you're probably putting it off until it becomes a problem. Automated daily visibility removes the friction and puts the decision-making data in front of you — without you having to go hunting for it.

Resend

resend.com — transactional email that doesn't require a developer.

What it does: sends one-off emails triggered by events. Not newsletter campaigns — the individual stuff:

- "Class starts in an hour"
- "You're registered for this event"
- "Here's your session summary"

Pricing: Free under 2,500 emails per month. Easy to configure with Claude. Pairs well with n8n automations.

"Email for developers. Don't fret. It's also email for people like us."

SECTION 8: MANAGING API COSTS FOR CLIENT TOOLS

The Core Problem

When you build a tool powered by Claude and let other people use it, their usage = your bill.

RJ's Recommended Model

Set a monthly cost threshold per user.

Example: \$20/month per user.

- **Below the threshold:** You absorb the cost. It's part of your program value.
- **Above the threshold:** The user gets an email: "Hey, you're using this a lot — here's how to add your own API key."

This model says: *I believe in this tool enough to fund reasonable usage. If you're using it so much it goes above \$20, you're getting enough value to invest in your own access.*

"It protects yourself because you don't want to be paying a hundred bucks a month on one person — especially if you're only charging a hundred bucks a month for the program."

Other Techniques

For structured multi-step tools (like a Dream 100 contact builder):

- **Frequency limits:** Users can only run the full sequence once every 90 days. Doors close, doors open.
- **Action gates:** Don't unlock Phase 2 until they've taken real action on Phase 1.
- **Implementation pacing:** Build in waiting periods between phases that force action, not just generation.

"If you want your Dream 100 — have you reached out to the 200? Why develop another 200 if you haven't taken any action on the first batch?"

SECTION 9: THE FUTURE OF COACHING AND IMPLEMENTATION TOOLS

Bez's Dream 100 → 200 Project

Bez is building a contact tracker that started as a bot in Pickaxe. Pickaxe dropped their free tier, so it needs a new home — and she's now thinking about building a full interface instead.

The tool she envisions:

- Generates and organizes 100+ outreach targets
- Tracks contact across channels (email, LinkedIn, WhatsApp, SMS)
- Could limit cadence (40 emails/day to warm up the inbox)
- Includes a structured planning day and implementation sequence
- Lets a coach monitor what clients are actually doing

RJ's take on this:

"The relationship ends at the cash register — no more. People are going to push into relationships where you've not only taught them the stuff, you're going to help them make it happen."

The future of coaching isn't 18 steps to magnificence for \$2,000. It's: here is the tool that actually makes those steps happen.

Implementation Is the Only Metric That Matters

"If they have more two weeks after joining than when they joined — if they have momentum — you'll get a lot more people who scream about your class because they've actually done stuff."

Clients who implement trust you — even if they get different results than they expected. Clients who don't implement drift away and say nothing.

Stack for This Kind of Tool

- **Backend:** Airtable (RJ's choice over Supabase for non-developers — "I know what a spreadsheet looks like. I can handle that.")
- **Frontend:** Interface or PWA built with Claude Code + Cloudflare
- **Cost model:** Absorb up to \$20/user/month, API key handoff above that

SECTION 10: THE MASTER COMMAND CENTER

RJ is building a personal daily dashboard that surfaces:

- What's coming in and going out (Stripe + email)
- What tasks are waiting
- Priority clarity for the day

The goal: replace the creative-brain tendency to drift away from priorities by having everything visible at a glance. Currently being redesigned — first version had too many features and most of them were useless.

"I was going nuts with all of the things I could have on here. And then most of them were useless. Let your imagination go on these things and you will create lots more stuff."

Coming soon: gamification. Earning rewards, tracking momentum, making the admin side of entrepreneurship feel less like admin.

"It's boring. I want to be able to like earn different things and — stuff that's rewarding."

SECTION 11: MEMBERSHIP & WHAT'S NEXT

RJ is building a **NO BS AI membership** around this class.

What that means:

- Class materials remain available forever
- Ongoing access with weekly meetings (Thursdays, 11am)
- **Founders pricing:** If you're already paying RJ, no price increase
- Next cohorts: higher price point

"I really think that so much happens in AI that a weekly meeting is the way I can keep everybody up to date."
