Automate Your Life

A Step-by-Step Guide for Busy Professionals

Use Claude Code and Cowork to eliminate repetitive tasks — no coding experience required.

From spreadsheets to patient digests, expense reports to client follow-ups — learn how to delegate your busywork to AI and reclaim hours every week.

Corey Slick

Slick Engineering & Consulting, Inc.

www.slick-engineering.com

corey@slick-engineering.com

Table of Contents

Introduction

Who This Guide Is For

You are good at what you do. You spent years learning your craft — whether that means diagnosing patients, winning cases, closing deals, or running a business. But somewhere along the way, the administrative side of your work started eating your life. You spend your evenings writing reports that could be templated. You spend your weekends sorting through data that should organize itself. You have a nagging feeling that half the tasks on your to-do list shouldn't require a human brain at all — and you're right.

This guide is for professionals who are excellent at their jobs but have zero interest in becoming programmers. You are a doctor, a lawyer, a realtor, a financial advisor, a small business owner, a consultant — someone whose time is genuinely expensive and who cannot afford to waste it on repetitive busywork. You don't have an IT department. You don't have a personal assistant who can build you custom software. You may not even consider yourself particularly "tech-savvy," and that is completely fine. If you can write a clear email, you can use what this guide teaches. The primary tool you will learn is called Claude Cowork, and it lives inside the Claude Desktop app you may already have on your computer. It works with the files on your machine, creates documents and spreadsheets, and can even run tasks on a schedule — all without you writing a single line of code.

This guide is specifically *not* for software engineers or people who already know how to code. There are plenty of resources for them. This is for the rest of us — the professionals who have been told for years that "automation" and "AI" will change everything, but who have never been handed a practical, step-by-step playbook for making that actually happen in their daily work. That playbook is what you are holding right now.

What You'll Be Able to Do by the End

By the time you finish this guide and work through the examples, you will be able to:

- **Generate polished documents automatically** — patient summary letters, referral responses, client reports, listing presentations — by pointing Claude Cowork at your data files and describing the format you want. No more copying and pasting between spreadsheets and Word documents.

- **Set up scheduled reports that run themselves** — a Monday morning patient volume digest, a weekly revenue summary, a daily inbox brief — using Cowork's built-in scheduling feature. You open your laptop and the work is already done.
- **Organize files without lifting a finger** — rename hundreds of documents according to a consistent convention, sort incoming files into the right folders by date or client or type, and clean up the chaos of your desktop and Downloads folder, all by describing what you want in plain English.
- **Analyze your own business data on demand** — spot trends in patient referrals, compare revenue across quarters, break down caseloads by type, answer questions like "Which referral source sent the most patients last quarter?" in seconds instead of hours.
- **Create repeatable workflows for multi-step processes** — billing summaries, compliance checks, client follow-ups, end-of-month reporting — so that tasks you do weekly or monthly happen with a single prompt instead of an hour of manual work.
- **Build communication drafts using your real data** — follow-up emails to every client who hasn't had an appointment in 90 days, referral thank-you letters populated with actual names and dates, all pulled directly from your spreadsheets and files.
- **Create presentations and formatted documents** — slide decks for partner meetings, formatted PDF reports, summary briefs for staff — by describing the content and letting Cowork handle the layout and structure.
- **Build code-heavy projects like dashboards and web tools** — using Claude Code, the coding-focused tool you will learn about in specific chapters, for tasks that go beyond document generation into custom applications, like a practice finance dashboard or a patient portal update.

The Only Mindset Shift You Need

Here is the single most important idea in this entire guide, and once it clicks, everything else will fall into place:

You are the manager. Claude is your analyst.

Think about what a good manager does. A good manager does not do the analyst's work. A good manager does not need to know the technical details of *how* the analyst builds a spreadsheet model or writes a report. What a good manager does is describe the problem clearly, explain what the end result should look like, and review the work when it comes back.

That is your entire job here. You do not need to learn to code. Not now, not ever. You will never need to understand what "Python" or "JavaScript" or "API" means at a technical level. What you need to know how to do — and what you are already good at — is describe problems in plain English.

When you say to a new hire, "I need a summary of last quarter's referral numbers broken down by source, with the top three highlighted," that person doesn't ask you to write the Excel formulas yourself. They go figure it out. Claude works the same way. You describe the outcome. It figures out the how.

The professionals who get the most out of these tools are not the most technical ones. They are the ones who are best at clearly describing what they want. They are the ones who can say, "Here's the problem I face every Tuesday morning, here's what I wish would happen instead, and here's what the result should look like." If you can do that — and you can, because you do it every time you delegate to a colleague or explain a situation to a client — then you already have the only skill that matters.

This works whether you are talking to Cowork about organizing your files or telling Claude Code to build you a dashboard. The tool is different, but the skill is the same: clear communication about what you need.

Throughout this guide, we will follow Dr. Sarah Mitchell, a physician at Riverside Medical Group, as she discovers how to put this mindset into practice. Sarah is not a technologist. She is a busy doctor who got tired of spending her evenings on paperwork. Her story will show you exactly how a real professional — someone like you — goes from "I've heard of AI" to "I just saved myself six hours this week."

How to Use This Guide

This guide is designed to be read in order the first time through, then used as a reference afterward. Here is how it is structured:

Chapters 1-3: Foundation. You will learn what Claude's tools can do, how to get the Claude Desktop app set up on your computer, and how to have your first successful conversation with Cowork. Even if you are itching to jump ahead, these chapters will save you frustration later. They take about an hour to work through. Most of this guide focuses on Cowork — the desktop tool that handles documents, files, scheduling, and everyday automation — because that is where the biggest time savings are for most professionals.

Chapters 4-7: Core Skills. These are the building blocks. You will learn how to work with your files, make sense of your data, create automations using Cowork's scheduling features, and build workflows that run themselves. Each chapter includes real examples based on Dr. Mitchell's medical practice, but the principles apply to any profession. Take these one at a time — try each example with your own data before moving on.

Chapters 8-10: Putting It All Together. This is where you build real projects. You will create scheduled morning reports, automated document pipelines, and — in the chapters that cover Claude Code — a custom finance dashboard that pulls your practice data into a visual display. The Claude Code sections are clearly marked and are specifically for projects that need actual code, like building web-based tools or connecting to online services. By the end, you will have built something genuinely useful that saves you real time every week.

The Appendix includes a troubleshooting guide, a quick-reference card for common prompts, and a collection of ready-made workflows organized by profession.

A few practical notes:

- **You will need a computer for this.** Claude Cowork runs inside the Claude Desktop app on Mac or Windows. Claude Code is accessible through your web browser at claude.ai. Neither runs on your phone or tablet for the tasks in this guide. You will need to be sitting at your computer for the hands-on sections.
- **Set aside real time.** Do not try to skim this during a lunch break. Block out a Saturday morning or a quiet evening. The first three chapters need about an hour. Each subsequent chapter needs 30-45 minutes.
- **Use your own data.** The examples use Dr. Mitchell's scenarios, but you should follow along with your own files, your own spreadsheets, your own problems. The guide will be ten times more valuable if you finish it having built something real for your practice or business.
- **Expect to make mistakes.** You will type something that doesn't work. Claude will misunderstand you. This is completely normal and not a sign that you're doing it wrong. The guide will teach you how to course-correct quickly.
- **You don't need to memorize anything.** Every prompt and technique in this guide can be looked up later. Focus on understanding the concepts, not on memorizing exact wording.
- **Most chapters focus on Cowork.** That is intentional. Cowork handles the vast majority of what busy professionals need — documents, data, file organization, scheduled tasks. The chapters that cover Claude Code are clearly labeled and are there for when you want to tackle a code-heavy project, like building a custom dashboard or updating a website.

Let's get started.

What Are Claude's Tools?

Claude Can Do More Than Chat

If you have used Claude before — through the website, the phone app, or even the desktop app — you already know it can answer questions, write drafts, explain complicated topics, and hold a genuinely useful conversation. What you may not know is that Claude has grown well beyond that. It can now take actions. Real ones. On your actual computer, with your actual files.

Think of the difference between asking a colleague for advice and asking a colleague to do the work. When you text a smart friend, "How should I organize my referral files?" they might send back a thoughtful paragraph with a good strategy. Helpful, but you still have to do the organizing yourself. Now imagine instead you could say, "Go organize my referral files — sort them by referring physician, rename them with the date first, and put duplicates in a separate folder." And the colleague just does it. That is the leap Claude has made.

This happened because Claude gained the ability to act as an *agent*. An agent, in this context, is simply an AI that can take a series of steps to accomplish a goal, rather than just responding to a single question. When you ask regular Claude to summarize a spreadsheet, it can only work with what you paste into the conversation. When you ask an agentic version of Claude to summarize a spreadsheet, it can open the file itself, read through every row, decide how to structure the summary, write the document, and save it to your desktop. It plans, it executes, it delivers a finished result.

This is not some futuristic concept. It is available right now, and Anthropic has built two specific tools that put this capability in your hands: **Claude Cowork** and **Claude Code**. They serve different purposes, they live in different places, and this chapter will explain both so you know exactly which one to reach for and when.

Cowork: Your Desktop Automation Assistant

Claude Cowork is the tool you will use most in this guide, and it is the one most likely to change your daily routine. Cowork lives inside the Claude Desktop app — the same application you may already have installed on your Mac or Windows computer. If you have been using the desktop app just for conversations, Cowork is what happens when you give Claude permission to stop just talking and start doing.

Here is what Cowork can do:

Work with your local files. Cowork can read documents, spreadsheets, and PDFs that are sitting right there on your computer. It does not need you to upload them one at a time into a chat window. You can point it at a folder and say, "Look at everything in here." It can read a messy CSV export from your practice management system, understand the column structure, and pull out the numbers you care about. It can open a stack of referral letters and extract the key details from each one.

Create documents, spreadsheets, and presentations. This is where Cowork truly shines for professionals. You can say, "Create a summary report of last week's patient visits, broken down by provider, in a clean format I can present at the partner meeting." Cowork will produce the document. Not a suggestion for how you might create it yourself — the actual finished document, saved to your computer, ready to print or email. It can create spreadsheets with formulas already built in. It can put together presentation slides with the data already populated.

Run tasks on a schedule. This is perhaps the most powerful feature for busy professionals, and the one that makes the biggest difference in Dr. Mitchell's story. You can set up Cowork to run specific tasks automatically — every morning, every Monday, every first of the month. Imagine opening your laptop on Monday morning and finding that a patient volume summary for last week is already sitting on your desktop, formatted and ready. You did not have to ask for it. You did not have to open any applications. Cowork ran the task while you were sleeping, or the moment your computer woke up, and the result was waiting for you. We will cover scheduling in detail in a later chapter, but know for now that it exists and it is straightforward to set up.

Search the web and connect to other tools. Cowork can look things up online when you need current information folded into your work. If you are preparing a report and need the latest drug interaction data, or current market comparables for a real estate listing, or recent case law on a specific topic, Cowork can find that information and weave it into the document it is building for you.

The key thing to understand about Cowork is that it works with the world you already live in: the files on your computer, the folders you already use, the documents you already create. It does not require you to learn a new system or migrate your work somewhere else. It meets you exactly where you are.

Claude Code: The Coding Tool

Claude Code is the second tool, and it serves a very different purpose. While Cowork handles everyday document and file tasks on your desktop, Claude Code is designed for building things that require actual programming — websites, dashboards, custom applications, and software projects.

Claude Code lives at claude.ai, in a section called the Code tab. It connects to GitHub, which is the standard platform for storing and managing code projects. When you use Claude Code, it can write code, create entire applications, set up web pages, and build tools that run in a browser.

Here is the honest truth about Claude Code: for most of the tasks in this guide, you will not need it. Cowork handles the overwhelming majority of what busy professionals need — generating reports, organizing files, analyzing data, scheduling tasks. But there are specific projects where Claude Code is the right tool, and this guide includes chapters that walk you through those projects step by step.

The clearest example in this guide is the finance dashboard project. Later, you will build a custom web-based dashboard that displays your practice's financial data — revenue trends, expense breakdowns, provider productivity — in a visual format you can check from any device. That kind of project involves writing code that runs in a web browser, and that is exactly what Claude Code is built for. Another example would be updating a website. If your practice has a patient portal or a public-facing website and you want to add a new page, change the layout, or add a feature, Claude Code can do that by connecting directly to the code repository where your site lives.

You do not need to understand how code works to use Claude Code effectively. You describe what you want the same way you would describe it to a developer you hired. "I want a dashboard that shows monthly revenue in a bar chart, with the ability to filter by provider." Claude Code writes the code, tests it, and delivers the working result. Your job is to describe the outcome and review whether it looks right.

How These Are Different from Regular Claude Chat

It helps to think of Claude as coming in three modes, each one more capable than the last:

Claude Chat is the conversational Claude you may already know. You type a question or a request, and Claude responds with text. It is like having a very knowledgeable pen pal. You can ask it to draft an email, explain a concept, brainstorm ideas, or review something you paste into the conversation. It is tremendously useful, but it lives inside its chat window. It cannot see your files. It cannot save anything to your computer. It cannot take action on your behalf. When the

conversation ends, nothing has changed in your world except that you have new information in your head.

Claude Cowork is Claude with hands. It steps out of the chat window and into your computer. It can read your files without you pasting their contents. It can create new documents and save them where you want. It can process an entire folder of data in one pass. And crucially, it can do these things on a schedule, without you even being present. When you use Cowork, things change in your world — files get created, data gets organized, reports appear on your desktop. It is the difference between a consultant who sends you a memo of recommendations and an assistant who actually implements them.

Claude Code is Claude with a workshop. It does not just take actions on your existing files — it builds entirely new things. Web applications, dashboards, automated systems, custom tools. It works in the world of software development, connecting to code repositories and producing working programs. It is the difference between rearranging the furniture in your office and constructing a new room.

The technical term for what makes Cowork and Code different from regular chat is that they are *agentic*. That just means they can take multiple steps to accomplish a goal. Regular Claude answers your question in one response. Cowork and Code might take ten or twenty steps: reading a file, analyzing its structure, deciding how to organize the data, creating a new document, formatting it, saving it, and then telling you what they did. You give them a destination, and they figure out the route.

For the professionals reading this guide, here is the practical takeaway: start with Cowork for almost everything. It will handle your documents, your data, your file organization, and your scheduled tasks. Turn to Claude Code only when you want to build something that lives in a browser or requires connecting to online code repositories — and when you do, this guide will walk you through it clearly.

A Day in the Life: Dr. Sarah Mitchell's Monday Morning

To make all of this real, let's look at what a typical Monday looks like for Dr. Sarah Mitchell — now that she has set up the tools you will learn about in this guide.

6:15 AM — The Work Is Already Done

Sarah's alarm goes off at 6:15. She makes coffee, opens her laptop, and checks her desktop. Three files are waiting for her, created by Cowork's scheduled tasks while she slept:

The Patient Volume Digest. Every Sunday evening, Cowork runs a scheduled task that reads the patient visit export Sarah downloads from her practice management system every Friday afternoon. It cleans the data — removing blank rows, standardizing date formats, fixing the provider name abbreviations that are never consistent in the export. It produces a formatted summary showing how many patients each provider saw last week, broken down by day, with a comparison to the same week last year. The document is saved as a clean PDF on Sarah's desktop, ready for the Wednesday partner meeting. This used to take her twenty-five minutes of manual Excel work every Monday morning.

The Inbox Summary. Sarah uses a simple rule in her email client to save copies of incoming messages to a local folder. Cowork's scheduled task scans these messages every morning, categorizes them (administrative, requires response, informational, urgent), and produces a one-page brief: "11 new emails since Friday. 4 need your response. 2 are time-sensitive — Dr. Patel's referral question and the insurance pre-auth for the Martinez case. Here are two-sentence summaries of each, with draft responses for the 3 routine ones." This used to be her twenty-minute email triage session.

The Meeting Brief. Sarah has a meeting with the hospital's quality committee at 10 AM today. Cowork's scheduled task pulled together the agenda (from a document the committee chair shared last week, which Sarah saved to her local files), her notes from the previous meeting, and relevant data from recent patient outcomes — all into a single two-page brief. She would have scrambled to prepare this between patients.

Sarah glances through all three documents while drinking her coffee. She approves two of the three draft email responses with minor tweaks and writes a quick personal reply to Dr. Patel — that one requires her professional judgment and a human touch. She makes one note on the meeting brief. Total time: twelve minutes.

6:30 AM — An Ad-Hoc Request

Sarah remembers that the partners asked her to look into whether the new referral source — the urgent care clinic on Oak Street — is actually sending worthwhile cases, or just volume. She opens Cowork in the Claude Desktop app and types:

"Look at the referral tracking spreadsheet on my desktop. Filter for referrals from Oak Street Urgent Care in the last six months. Show me how many there were per month, what percentage converted to ongoing patients, and how their revenue compares to our average referral source."

Ninety seconds later, Cowork has read the spreadsheet, run the analysis, and produced a short summary with a table. The Oak Street referrals are high volume but low conversion — lots of one-time visits, fewer ongoing patients. The revenue per referral is about 60% of average. Sarah now has the data she needs for the partner discussion, and it took her less time than it would have taken to open Excel and remember which column was which.

7:00 AM — A Calm Morning

Sarah showers, dresses, eats breakfast sitting down, and drives to the clinic. She arrives at 7:25, five minutes early, with her meeting brief in her bag and her inbox already under control.

12:30 PM — A Different Kind of Task

During her lunch break, Sarah checks in on a project she started last week: a custom finance dashboard for the practice. This is not a Cowork task — it is a Claude Code project. The dashboard is a web-based tool that pulls data from the practice's monthly financial exports and displays revenue trends, expense breakdowns, and provider productivity in colorful charts that the partners can access from any computer.

Sarah opens claude.ai in her browser and navigates to the Code tab, where her dashboard project is connected to a GitHub repository. She types:

"The partners want to see a year-over-year comparison on the revenue chart. Add a toggle that lets you switch between showing just this year and showing this year versus last year."

Claude Code writes the code, creates what is called a pull request (essentially a proposed change that can be reviewed before going live), and shows her a preview. The toggle works. The year-over-year comparison is clear. Sarah approves the change. She did not write any code. She did not need to understand how charts are built in a browser. She described what she wanted, reviewed the result, and moved on.

What This Morning Tells You

Notice the pattern. The tasks that shaped Sarah's morning fell into two clear categories:

Cowork handled the everyday work. The patient digest, the inbox summary, the meeting brief, the ad-hoc referral analysis — these are all document and data tasks that involve files on her computer. Cowork did most of them automatically through scheduled tasks, and the ad-hoc analysis took a single prompt. This is what Cowork is built for, and it is what the majority of this guide will teach you.

Claude Code handled the building project. The finance dashboard is a custom web application — something that requires actual code running in a browser. That is Claude Code's territory. It is a powerful tool, but it is a specialized one. Sarah uses it for specific projects, not for her daily routine.

For most professionals, Cowork will handle 90% or more of what you need. It is the tool that saves you time every single day. Claude Code is the tool you reach for when you want to build something bigger — and when you do, this guide will be right there with you, walking through every step.

What Changed

Nothing about Sarah's medical skills changed. Nothing about her job responsibilities changed. She did not become a "tech person." She cannot write a single line of code, and she does not need to.

What changed is that she stopped doing work that did not require a physician's brain. Cleaning up spreadsheet data does not require medical school. Copying information from PDFs into a tracking sheet does not require clinical judgment. Sorting emails into categories does not require two decades of patient care experience.

Sarah kept the parts of her morning that need *her* — the clinical review, the professional communication with Dr. Patel, the decision-making about the Oak Street referrals. She handed everything else to tools that do it faster, more consistently, and without complaint.

That is what Claude's tools do. Not magic. Not science fiction. Just a clear division of labor between you and a very capable set of digital assistants.

The rest of this guide will show you, step by step, how to get there.

Cowork vs. Code: Which Tool When?

The Simple Way to Think About It

By now you know that Claude is not just a chatbot you text on your phone. It is a tool that can take real action on your behalf. But you may have noticed two different products mentioned — Claude Cowork and Claude Code — and wondered: do I need both? Are they the same thing? Which one is for me?

Here is the short answer: you will probably use both, but for very different reasons. And once you understand the distinction, you will always know which one to reach for.

Two Tools, One Analogy

Think about two people you might hire for your practice or business.

The first is an office assistant. This person sits at a desk near yours. They organize your files, prepare your reports, compile your meeting notes, sort your emails, and handle the steady stream of administrative tasks that keep your operation running. They work with the documents, spreadsheets, and tools you already use. They show up every morning, follow your routines, and keep things humming without you having to think about it.

The second is a software developer. You call this person when you need something built from scratch. A custom dashboard that shows your key business numbers in one place. A website for your practice. An internal tool where your staff can enter data and have it flow automatically into the right places. The developer does not organize your files or write your reports — they build the systems and tools that did not exist before.

Claude Cowork is your office assistant. Claude Code is your software developer. Both work for you. Both are powered by the same Claude intelligence. But they are designed for fundamentally different kinds of work.

Where They Live

This is one of the easiest ways to tell them apart.

Claude Cowork lives in the Claude Desktop app, which you install on your Mac or Windows computer. It runs in an isolated environment right on your machine, works with your local files and folders, and connects to services you already use — Google Drive, Gmail, DocuSign, and others through built-in connectors. You interact with it in a comfortable, visual interface designed for people who have never written a line of code.

Claude Code (web) lives at claude.ai/code in your browser. It runs on Anthropic's cloud servers and connects to GitHub repositories where software projects are stored. Its interface is built for creating and managing software — writing code, fixing bugs, and building applications. A non-technical person would use it not to write code themselves, but to tell Claude what to build and let it handle the technical side.

When to Use Which: A Quick Reference

The table below covers the most common tasks professionals face. Find the ones that match your work and you will immediately see which tool to reach for.

Task	Use This	Why
Morning patient digest or case summary	Cowork	It reads your local files and produces a document on a schedule
Weekly expense report from spreadsheet data	Cowork	It processes your existing spreadsheets and generates formatted reports
Custom finance dashboard for your practice	Code	This requires building a web application from scratch
Drafting client follow-up emails	Cowork	It works with your contact data and email connectors
Updating or building your practice website	Code	Websites are software — Code is built for creating and modifying them
Turning meeting notes into action items	Cowork	It reads documents and produces structured summaries
Building a custom data visualization tool	Code	Interactive tools with charts and filters need to be built as software

Task	Use This	Why
Organizing files into folders by date or client	Cowork	It works directly with the files and folders on your computer
Creating a staff intake form that auto-fills documents	Code	Custom forms with logic behind them are software projects
Daily email triage and summary	Cowork	It connects to Gmail and processes messages on a schedule
Building an internal scheduling tool for your team	Code	This is a custom application that needs to be built
Generating a polished PDF report from raw data	Cowork	It reads your data files and produces formatted output locally

Where They Overlap

Cowork and Code are not completely separate worlds. They share a common foundation, and understanding the overlap will help you avoid confusion.

Both are powered by Claude. The same intelligence that understands your plain-English instructions in Cowork is the same intelligence building software in Code. You do not need to learn a different way of communicating for each one.

Both can work with files. Cowork reads and writes files on your local computer. Code reads and writes files in software projects. The difference is scope and purpose — Cowork handles your day-to-day documents, while Code handles the files that make up an application.

Both take multi-step actions. Whether you ask Cowork to scan a folder, extract data from twelve PDFs, and compile a summary report, or you ask Code to create a web page, connect it to a database, and add a login screen — both tools break your request into steps and execute them one after another.

The real difference comes down to two things: **what they connect to** and **what they are optimized for**. Cowork connects to your files, your email, your cloud storage, and your everyday productivity tools. Code connects to software repositories and development infrastructure.

Cowork is optimized for knowledge work — documents, data, and recurring tasks. Code is optimized for building software.

When in Doubt

If you are ever unsure which tool to use, ask yourself one question:

Am I working with existing documents and data, or do I need something built from scratch?

If your task involves documents, spreadsheets, emails, file organization, or recurring reports — that is Cowork. You are working with things that already exist and asking Claude to process, analyze, summarize, or automate them.

If your task involves creating a new tool, building a website, designing a dashboard application, or producing custom software that your team will use — that is Code. You are asking Claude to build something that does not exist yet.

Here is how Dr. Sarah Mitchell thinks about it. On a typical Monday, she uses Cowork to run her morning routine — processing patient data, summarizing emails, compiling her utilization report. These are tasks that happen every week, work with files she already has, and produce documents she sends to colleagues. Cowork handles all of this without her thinking about technology.

But last month, Sarah wanted something different. She wanted a dashboard — a single web page she could pull up on her office monitor that showed her practice's key numbers at a glance: patient volume trends, revenue by provider, referral sources, and upcoming deadlines. That did not exist anywhere. No spreadsheet or document could do what she was imagining. She needed something built.

So she went to claude.ai/code, described what she wanted in plain English, and Claude Code built it for her. She did not write a single line of code. She described the dashboard the same way she would describe it to a contractor: "I want a clean page with four sections. The top left should show patient volume by week as a line chart. The top right should show revenue broken down by provider..." Claude Code built the application, and now she has a dashboard she opens every morning alongside her Cowork-generated reports.

Two different tools. Two different jobs. One professional who knows when to use each.

What This Guide Covers

Now that you understand the distinction, here is how the rest of this guide is organized.

The majority of this guide focuses on Cowork. Chapters 4 through 9 will teach you everything you need to know about using Cowork to automate your daily work — from writing effective prompts to setting up scheduled tasks that run without you. For most professionals, Cowork is where you will spend ninety percent of your time, and where you will see the biggest immediate return on the hours you invest in learning.

Claude Code appears in specific use cases. When a particular project calls for building something from scratch — like the finance dashboard in the use cases chapter — the guide will walk you through using Claude Code for that specific task. You will not need to become a developer. You will simply learn how to describe what you want built, the same way you would describe it to a contractor or consultant.

Think of it this way: Cowork is your daily driver. Code is the specialist you call in for specific projects. Both are valuable. Both are accessible to you without any technical background. And by the end of this guide, you will know exactly how to get the most out of each one.

Getting Started

This chapter gets you up and running with the two tools you will use throughout this guide: **Cowork** (for everyday automation) and **Claude Code** (for code-oriented projects). There is nothing to install from a command line. Everything starts with downloading an app or opening a website.

By the end of this chapter, you will have Cowork running on your computer with a folder connected, and you will know where to find Claude Code in your browser if you ever need it.

What You Will Need Before You Start

- A Mac or Windows computer with a working internet connection.
- A Claude account with a **Pro**, **Max**, **Team**, or **Enterprise** subscription. If you do not have one yet, go to claude.ai and sign up before continuing.
- About 15 minutes of uninterrupted time.

That is the entire list. No special software prerequisites, no technical background required.

Part 1: Setting Up Cowork

Cowork is the tool you will use for the vast majority of tasks in this guide. It lives inside the Claude Desktop app and works by reading and writing files on your computer — organizing documents, generating reports, drafting emails, and running scheduled tasks in the background. Think of it as a tireless assistant sitting inside your computer, ready to work whenever you give it instructions.

Downloading the Claude Desktop App

Step 1. Open your web browser (Safari, Chrome, Edge, or whichever you prefer) and go to the following address:

```
https://claude.ai/download
```

Step 2. The page will detect whether you are on a Mac or Windows computer and show you the correct download button. Click **Download for macOS** or **Download for Windows**.

Step 3. Once the download finishes, open the installer:

- **On a Mac:** Open your Downloads folder and double-click the `.dmg` file. When a window appears, drag the Claude icon into your Applications folder. Then open your Applications folder and double-click **Claude** to launch the app.
- **On Windows:** Open your Downloads folder and double-click the `.exe` file. Follow the setup wizard — click **Next** on each screen and then **Install**. Windows may ask for permission; click **Yes**. When the installer finishes, click **Finish** to launch the app.

Step 4. The Claude Desktop app will open for the first time. You will see a sign-in screen.

Signing In

Step 5. Click **Sign In** and enter the email address and password for your Claude account. If your organization uses single sign-on (SSO), follow the prompts to sign in through your company's login page.

Step 6. After signing in, you will land on the main Claude Desktop screen. This is where you can chat with Claude directly, but more importantly, it is where Cowork lives.

Opening Cowork

Step 7. Look at the left sidebar of the Claude Desktop app. You will see an option labeled **Cowork**. Click it.

Step 8. Cowork will open and greet you with a brief introduction. Take a moment to read it — it explains what Cowork can do and how it interacts with files on your computer.

Connecting a Local Folder

Cowork needs access to a folder on your computer so it has a place to read files from and save its work to. This is the most important setup step.

Step 9. In the Cowork interface, look for the option to **connect a folder** (sometimes labeled "Add folder" or "Connect local folder"). Click it.

Step 10. A standard file-picker window will appear — the same kind you see when attaching a file to an email. Navigate to a folder you want Cowork to work with. Good choices for your first folder:

- Your **Documents** folder
- A specific project folder, like a folder of client files or financial records
- A new empty folder you create just for testing (for example, create a folder on your Desktop called "Cowork Test")

Select the folder and click **Open** (Mac) or **Select Folder** (Windows).

Step 11. Cowork will confirm the folder is connected. From now on, Cowork can read files inside that folder and create new files there when you ask it to.

> **Tip:** You can connect more than one folder later. Start with one for now and add others as you get comfortable.

Connecting Additional Services (Optional)

Step 12. Cowork can also connect to cloud services like **Google Drive**, **Gmail**, and others. These connections are entirely optional and not required to follow this guide. If you want to set one up now:

- Look for **Integrations** or **Connected Services** in the Cowork settings.
- Click the service you want to connect (for example, Google Drive).
- Follow the on-screen prompts to sign in and grant permission.

You can always come back and add these later. For now, a connected local folder is all you need.

Your First Task

Let's confirm everything is working by giving Cowork something simple to do.

Step 13. In the Cowork input area (the text box where you type messages), type the following:

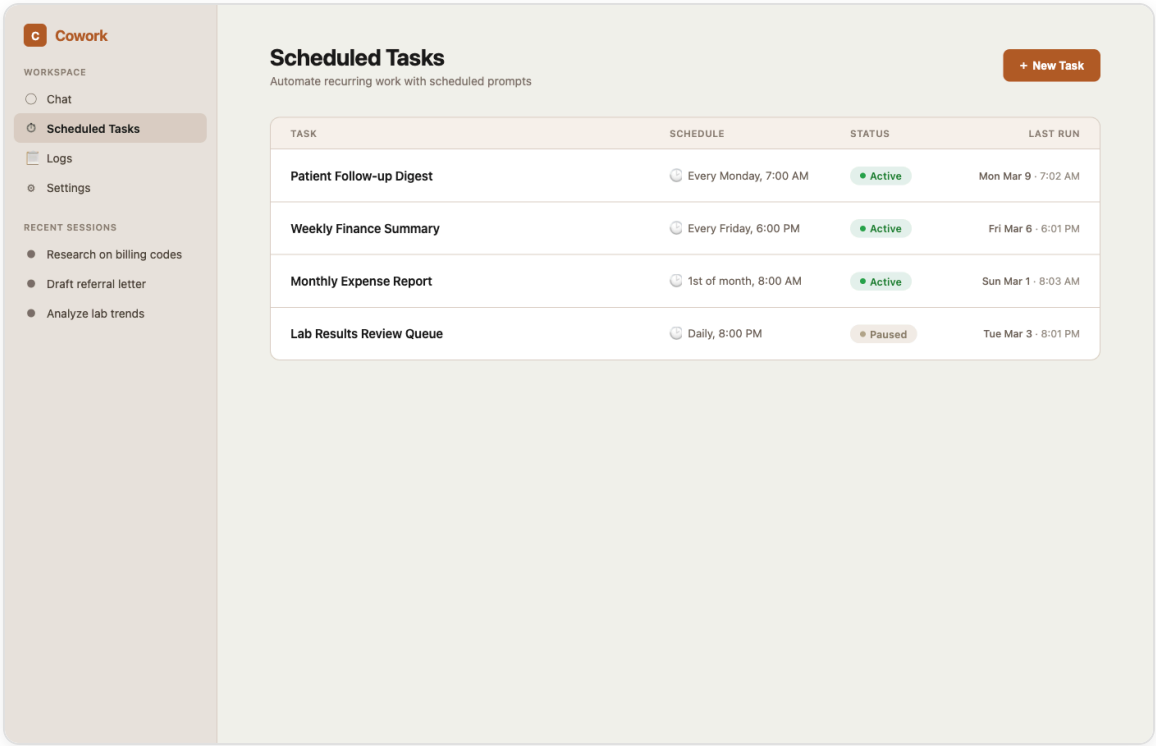
```
Look at the files in my connected folder and tell me what's in there. Then create a new f
```

Step 14. Press **Enter** (or click the send button). Cowork will read through the folder you connected, describe what it finds, and create the new text file.

Step 15. Open the folder on your computer using Finder (Mac) or File Explorer (Windows). You should see the new file called `hello.txt` sitting right there. Open it to confirm the contents.

If you see the file, congratulations — Cowork is set up and working. You just delegated your first task.

> **Want to try something more useful?** Ask Cowork to organize the files in your connected folder by type. Type something like: *"Sort the files in my folder into subfolders by file type — put all PDFs in a folder called PDFs, all images in a folder called Images, and everything else in a folder called Other."* Cowork will create the subfolders and move your files for you.



Mock Cowork Tasks

The Cowork interface showing a connected folder and a completed task. The left sidebar shows the Cowork section, and the main area displays the conversation where Cowork has carried out an instruction and confirmed the result.

Part 2: Accessing Claude Code (for Code Projects)

Claude Code is the tool you will use when a project involves actual programming — building a dashboard, creating a web page, or modifying code in a software repository. Most chapters in this

guide use Cowork, but a few specific projects (like building a personal dashboard in Chapter 8) will walk you through Claude Code.

You do not need to install anything to use Claude Code. It runs in your web browser.

Opening Claude Code

Step 1. Open your web browser and go to:

```
https://claude.ai
```

Step 2. Sign in with your Claude account if you are not already signed in.

Step 3. Look at the left sidebar. You will see a tab or section labeled **Code**. Click it.

Step 4. You are now in the Claude Code interface. It looks similar to regular Claude chat, but it is designed specifically for working with code and software projects.

Connecting Your GitHub Account (Optional)

If you have a GitHub account — a place where software projects are stored online — you can connect it so Claude Code can read and modify your repositories directly.

Step 5. In the Claude Code interface, look for an option to **connect GitHub** (you may find this in settings or as a prompt when you first open a project).

Step 6. Click the connect button and follow the prompts to sign in to GitHub and grant Claude Code access to your repositories.

Step 7. Once connected, you can select a repository from your account to work in. Claude Code will be able to read the files in that repository and make changes when you ask it to.

> **Important:** You do **not** need a GitHub account to follow most of this guide. GitHub is only relevant if you are working on a software project that lives there. If you do not know what GitHub is, you can safely skip this section and come back to it if a later chapter asks you to use Claude Code with a repository.

Your First Task in Claude Code

Step 8. Whether or not you connected GitHub, you can start a new session in Claude Code. Look for a **New Session** or **Start New** button and click it.

Step 9. In the input area, type a simple request:

```
Create a basic HTML page that says "Hello, World!" with a blue background and white text,
```

Step 10. Press **Enter**. Claude Code will generate the code and show it to you. You can preview it, download it, or — if you connected a GitHub repository — commit it directly to your project.

This confirms Claude Code is working. You will not need it again until the guide specifically calls for it in later chapters.

Part 3: Which One Am I Using Right Now?

As you work through this guide, you will always be using one of three things: Cowork, Claude Code, or regular Claude chat. Here is how to tell them apart at a glance.

You Are Using Cowork If...

- You are in the **Claude Desktop app** (the application you downloaded and installed on your computer).
- You see **Cowork** highlighted in the left sidebar.
- Cowork has access to folders on your computer and can read or create files directly.

This is the tool you will use for most of this guide. File organization, report generation, email drafting, data analysis, and scheduled tasks all happen in Cowork.

You Are Using Claude Code If...

- You are in your **web browser** at **claude.ai**.
- You clicked the **Code** tab in the left sidebar.
- The interface is focused on code files, repositories, or software projects.

This tool appears in a few specific chapters where we build something that involves programming, like a personal dashboard or a simple web tool.

You Are Using Regular Claude Chat If...

- You are in your **web browser** at **claude.ai**.
- You are in the standard chat view — no Code tab selected, no Cowork sidebar.
- You are having a conversation with Claude but it does not have access to your computer's files.

Regular chat is great for brainstorming and asking questions, but it cannot read your files or create documents on your computer the way Cowork can. When this guide asks you to do hands-on work, it will always direct you to Cowork or Claude Code.

A Quick Reference

Question	Answer
Where am I?	Desktop app = Cowork. Browser at claude.ai/code = Claude Code. Browser at claude.ai in chat = Regular Claude.
Can it access my files?	Cowork = Yes (connected folders). Claude Code = Yes (connected repositories). Regular chat = No.
Do I need it for this guide?	Cowork = Yes, for most chapters. Claude Code = Only for specific technical projects. Regular chat = Optional, for brainstorming.

Troubleshooting: Common Setup Issues

The download page does not show a button for my operating system

Make sure you are visiting `claude.ai/download` in a browser on your Mac or Windows computer, not on your phone or tablet. If the page still does not detect your system, look for a manual download link that lets you choose your operating system.

The app asks me to sign in but my email is not recognized

Confirm that you are using the same email address you used when you created your Claude account. If you signed up through Google or Apple, use the **Sign in with Google** or **Sign in with Apple** option instead of typing your email manually.

I cannot find the Cowork option in the sidebar

Make sure your Claude Desktop app is updated to the latest version. If Cowork does not appear, close the app completely and reopen it. On a Mac, quit the app by pressing **Command + Q** and reopening it from Applications. On Windows, right-click the app icon in the taskbar, click **Close window**, and reopen it from the Start menu.

Cowork cannot see files in my connected folder

Double-check that you selected the correct folder during setup. You can disconnect and reconnect a folder in Cowork's settings. Also confirm that the folder actually contains files — if it is empty, Cowork will report that there is nothing to read, which is expected behavior.

I do not see the Code tab in the browser

Make sure you are signed in at `claude.ai` with a Pro, Max, Team, or Enterprise account. The Code tab may not be available on free plans. If you are on a supported plan and still do not see it, try refreshing the page or clearing your browser cache.

What's Next?

You now have Cowork set up on your computer with a folder connected, and you know where to find Claude Code in your browser. In the next chapter, we will explore what Cowork can do in more detail and show you how to set up tasks that run automatically on a schedule — so your work gets done even when you are not at your desk.

What Is Cowork?

The Short Version

Imagine hiring an assistant who works while you step away, never forgets a deadline, and handles the same tedious tasks flawlessly every single time. That is essentially what Cowork does — and it lives right inside the Claude Desktop app on your computer.

Cowork is the primary tool you will use throughout this guide. It is the feature inside the Claude Desktop application that lets you point Claude at your local files and folders, connect it to services you already use (like Google Drive and Gmail), and schedule tasks to run automatically on a repeating basis. You describe what you want done in plain English, tell Cowork when to do it, and walk away. The task runs in the background while you focus on the work that actually needs your brain.

Think of it like setting a recurring alarm on your phone, except instead of just making noise, the alarm actually does the work for you.

Where Cowork Lives

Cowork is not a website you visit in your browser. It is not a command you type into a terminal. It is a feature built directly into the **Claude Desktop app** — the application you download and install on your Mac or PC, the same way you would install Microsoft Word or Zoom.

When you open the Claude Desktop app, Cowork is right there. You do not need to install anything extra, sign up for a separate service, or learn a different interface. If you have the Claude Desktop app on your computer, you already have Cowork.

This matters because Cowork needs to be on your computer to do its best work. Unlike a web-based tool that can only see what you upload to it, Cowork can see and work with the files and folders sitting right on your machine — your Documents folder, your spreadsheets, your client records, your exported data. You point it at a directory, and it can read what is there, create new files, and organize everything without you lifting a finger.

For readers who also write code or build software, there is a companion tool called **Claude Code** that runs in the terminal and is designed specifically for programming tasks. But for the knowledge work this guide focuses on — reports, file organization, email drafts, data summaries, scheduling

— Cowork inside the Claude Desktop app is your primary tool. We will mention Claude Code only in the rare cases where a task involves actual software development.

Working with Your Local Files and Folders

The single most important thing to understand about Cowork is that it works directly with the files on your computer. This is what makes it so practical for everyday professional work.

Here is how it works in practice. You tell Cowork something like: "Look at the spreadsheet in my Documents/Practice/visits.csv file, pull the last seven days of patient visits, and save a summary report to Documents/Practice/Weekly Reports/." Cowork reads the file right off your hard drive, does the analysis, and saves the result exactly where you told it to — no uploading, no downloading, no copy-pasting between apps.

This local file access means Cowork can handle tasks like:

- **Reading your data files.** CSV exports, spreadsheets, text files, PDFs — Cowork can open and analyze files sitting in any folder on your computer.
- **Creating new documents.** Need a formatted report, a draft email, or a summary document? Cowork creates the file and saves it wherever you specify.
- **Organizing your folders.** Cowork can sort files into subfolders, rename files using consistent conventions, and clean up the digital clutter that accumulates over months of busy work.
- **Processing multiple files at once.** Point Cowork at a folder full of invoices, contracts, or patient records, and it can read through all of them and produce a consolidated summary.

You stay in control of what Cowork can access. You choose which folders to point it at, and it works only within the boundaries you set. It will not wander into folders you have not directed it to.

The Sandbox: A Safe Space to Work

One concern people naturally have is: "What if it accidentally deletes something important?" This is a fair question, and Cowork has a built-in answer.

Cowork works in a safe, isolated space — sometimes called a sandbox — so it cannot accidentally break anything on your computer. Think of it like a protective bubble. Cowork can read your files to understand them, but when it creates or modifies things, it does so in a controlled way that keeps your system safe. You review what it produces before anything permanent happens.

This sandbox approach means you can experiment freely. Try a new prompt, point Cowork at a test folder, see what it produces. If the result is not what you wanted, nothing has been damaged. Adjust your instructions and try again. The sandbox is your safety net while you learn.

Connectors: Linking Cowork to the Tools You Already Use

Cowork does not just work with files on your hard drive. It can also connect to the cloud services and apps that are already part of your daily workflow. These connections are called **connectors**, and they dramatically expand what Cowork can do for you.

Here are some of the connectors available:

Google Drive. Cowork can read files stored in your Google Drive, pull data from Google Sheets, and save output back to Drive folders. If your team shares documents through Google Workspace, Cowork can tap directly into that shared knowledge without you downloading anything first.

Gmail. Cowork can read your incoming emails and draft responses. Imagine waking up to find that Cowork has already sorted your overnight emails into categories — urgent, routine, FYI — and drafted replies for the routine ones. You review, tweak if needed, and send. What used to be an hour of morning email triage becomes ten minutes of review.

DocuSign. For professionals who deal with contracts and agreements, the DocuSign connector lets Cowork track document status, identify which agreements are awaiting signatures, and flag items that need your attention.

And more. The list of available connectors continues to grow. Each one follows the same principle: Cowork reaches into a tool you already use, pulls or pushes information, and saves you the manual work of switching between applications.

Setting up a connector is straightforward. In the Claude Desktop app, you authorize the connection (similar to how you grant a new app permission to access your Google account), and from that point on, Cowork can include that service in its tasks. You can revoke access at any time.

Scheduled Tasks: The Heart of Automation

The feature that makes Cowork truly powerful is scheduled tasks. Instead of opening the app and telling Cowork what to do every time, you set up a task once and it runs automatically on whatever schedule you choose — daily, weekly, monthly, or any custom interval.

Scheduled tasks are built right into Cowork's interface. You create them, manage them, and monitor them all from the same place you do everything else in the app. No cron jobs, no command-line wizardry, no technical setup. Just a description of what you want and a schedule for when you want it.

Here is the critical detail: **scheduled tasks only run when your computer is awake and the Claude Desktop app is open.** This is not a cloud service that runs on a remote server somewhere. Cowork lives on your machine, so your machine needs to be on and the app needs to be running for scheduled tasks to fire. If your computer is asleep or the app is closed when a task is scheduled to run, it will not execute until the next scheduled time when conditions are met.

For most professionals, this is not an issue. If you keep your computer on during work hours and leave the Claude Desktop app running in the background (it uses minimal resources), your tasks will fire reliably. Many people set their computers to stay awake during business hours specifically for this reason.

When you create a scheduled task, you provide three pieces of information:

1. **A name** — something short that reminds you what the task does, like "Morning Patient Digest" or "Weekly Expense Report."
2. **A schedule** — when and how often it should run. This can be as simple as "every weekday at 7 AM" or "the first Monday of every month at 9 AM."
3. **A prompt** — the instructions Cowork follows when the task runs. This is the most important part, and we will spend all of Chapter 5 teaching you how to write prompts that get great results.

Real Examples: What People Actually Schedule

The best way to understand Cowork is to see what it looks like in practice. Here are real-world examples organized by profession. As you read through these, pay attention to which ones sound like tasks you currently do by hand.

For Doctors and Healthcare Professionals

- **Daily briefings.** Every morning at 6 AM, Cowork reads through the day's patient notes and produces a one-page summary of key items that need your attention — abnormal lab results, medication conflicts, follow-up reminders. It is waiting on your desktop when you sit down with your coffee.

- **Referral letter drafts.** After each clinic day, Cowork takes your brief notes and turns them into properly formatted referral letters, ready for your review and signature the next morning.
- **Weekly caseload reports.** Every Friday afternoon, Cowork compiles your patient volume, procedure counts, and outstanding tasks into a clean summary you can hand to your office manager.

For Lawyers and Legal Professionals

- **Case status updates.** Every Monday morning, Cowork scans your case files and generates a status summary for each active matter — upcoming deadlines, recent filings, items awaiting client response.
- **Billing time compilation.** At the end of each day, Cowork takes your rough time entries and reformats them into proper billing descriptions with accurate time calculations.
- **Contract clause summaries.** When you drop a new contract into a designated folder, Cowork reads it and produces a plain-English summary of key terms, obligations, and red flags.

For Realtors and Real Estate Professionals

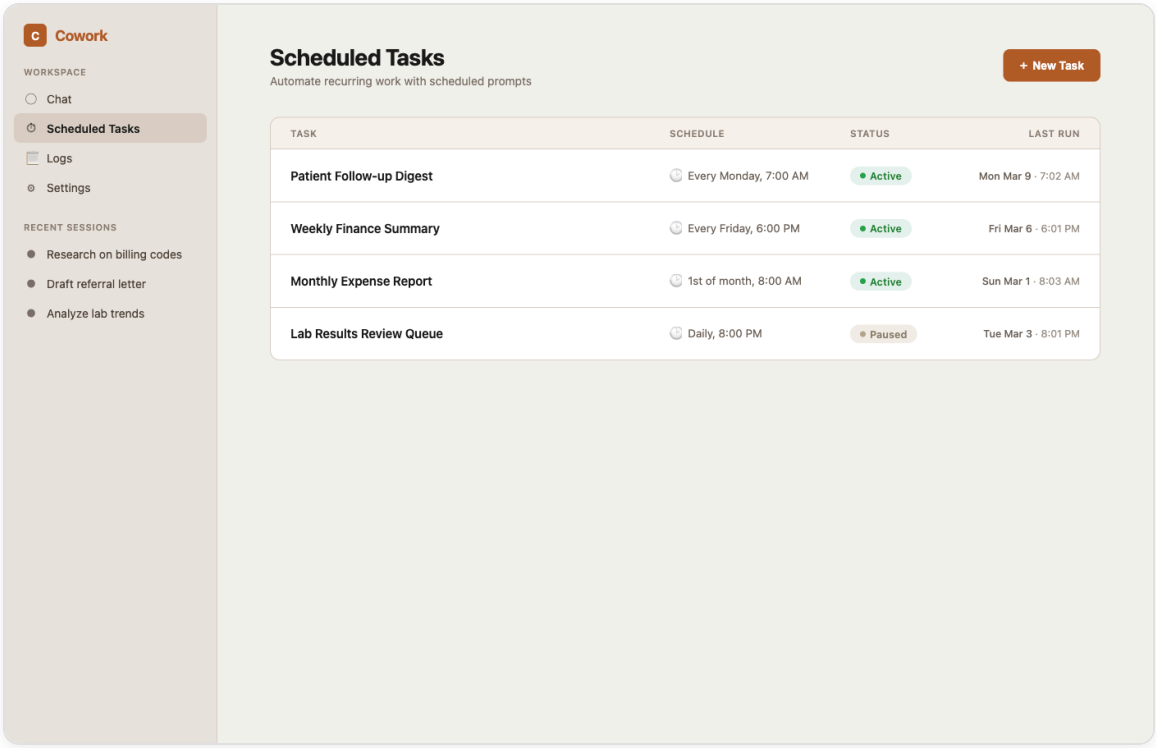
- **Market snapshot reports.** Every morning, Cowork pulls together recent comparable sales data and price trends for your active listing areas, giving you fresh talking points before your first showing.
- **Client follow-up drafts.** After an open house, Cowork takes your sign-in sheet data and drafts personalized follow-up emails for each attendee, ready for you to review and send.
- **Listing description drafts.** When you add property details to a folder, Cowork generates polished MLS descriptions and social media captions for each new listing.

For Small Business Owners

- **Daily expense categorization.** Every evening, Cowork reads your transaction exports and categorizes each expense, flagging anything unusual or over budget.
- **Weekly cash flow summaries.** Every Monday at 8 AM, Cowork compiles your income and expenses from the past week into a simple report that tells you exactly where you stand.
- **Monthly invoice generation.** On the first of every month, Cowork takes your client records and service logs, then generates draft invoices ready for your review.

Tasks That Work for Everyone

- **Email draft preparation.** Cowork reads incoming messages via the Gmail connector and drafts responses for you to review each morning.
- **Meeting prep summaries.** Before your first meeting each day, Cowork compiles relevant background information, previous notes, and agenda items into a single briefing document.
- **End-of-week reviews.** Every Friday, Cowork summarizes what you accomplished during the week and lists anything that carried over, giving you a clean starting point for Monday.



Mock Cowork Tasks

The Cowork task list showing several scheduled tasks, their timing, and their current status. Each task displays its name, schedule (such as "Every weekday at 7:00 AM"), and whether it ran successfully on its last execution.

Why This Matters for Your Practice

The average professional spends 10 to 20 hours per week on administrative tasks that follow a predictable pattern. A doctor's skill is in diagnosis, not formatting referral letters. A lawyer's value is in legal judgment, not compiling billing entries. A realtor's strength is in relationships, not writing the same listing description for the hundredth time.

Cowork lets you reclaim those hours. Start with one or two tasks and add more as you get comfortable. Over time, you build a collection of automated routines that handle the predictable parts of your work, freeing you to focus on the parts that actually require you.

The setup is simple: install the Claude Desktop app, open Cowork, point it at your files, connect it to your services, and start describing what you need. In the next chapter, we will teach you how to write the prompts that make these tasks work well.

How to Write Great Prompts

Why Prompting Matters

Cowork does exactly what you ask it to do. The problem is that most people do not ask clearly enough. A vague prompt produces vague results. A specific prompt produces output you can actually use — the kind you barely need to edit before sending, filing, or acting on.

You do not need any technical background. If you can describe a task clearly enough for a new employee to follow, you can write a great prompt.

The Anatomy of a Great Prompt

Every effective Cowork prompt has four components. You do not always need all four, but the more you include, the better your results will be.

1. Context — Tell Claude What It Needs to Know

Context is the background information Claude needs to do the job well. Without it, Claude is guessing. With it, Claude makes informed decisions.

Context answers questions like: - Who is this for? - What is the situation? - What has already happened? - What does Claude have access to?

Example: "I am a family law attorney. I have a client consultation summary in the file client-notes-march.txt. The client is seeking a custody modification."

2. Task — Tell Claude What to Do

The task is the specific action you want Claude to take. Be direct. Use action verbs. Say exactly what the output should be.

Example: "Write a draft follow-up letter to the client summarizing the key points from our consultation and outlining next steps."

3. Format — Tell Claude What the Output Should Look Like

Format tells Claude how to structure and present the result. Should it be a bulleted list? A formal letter? A table? A one-paragraph summary? If you do not specify, Claude will guess, and it might guess wrong.

Example: "Format the letter with a professional header, use numbered paragraphs for each key point, and keep the total length under one page."

4. Constraints — Tell Claude What to Avoid or Limit

Constraints are the guardrails. They prevent Claude from going off track, being too verbose, including things you do not want, or making assumptions it should not make.

Example: "Do not include any legal advice or opinions on likely outcomes. Keep the tone warm but professional. Do not use legal jargon that a non-lawyer would not understand."

When you put all four together, you get a prompt that leaves very little room for misinterpretation. Claude knows the situation, knows what to do, knows what the output should look like, and knows what to avoid.

Bad Prompt vs. Good Prompt: Three Side-by-Side Examples

The difference between a bad prompt and a good prompt is often just 30 seconds of additional thought. Here are three real-world scenarios showing what that difference looks like.

Scenario 1: A Doctor Preparing Patient Summaries

Bad Prompt:

> Summarize my patients.

What is wrong with this? Claude does not know which patients, what kind of summary, how long it should be, or what information matters. You will get a generic response that is not useful.

Good Prompt:

> I am a primary care physician. Read the file patients-march-12.csv, which contains today's patient appointment data with columns for patient name, chief complaint, vitals, and medications.

> For each patient, write a 3-sentence summary that includes: (1) the chief complaint, (2) any vital signs outside normal range, and (3) current medications relevant to the complaint.
>
> Format the output as a numbered list, one patient per entry. Do not include patients whose visits are marked as "routine check-up only." Keep each summary under 50 words.

This prompt has clear context (who you are, what the file contains), a specific task (3-sentence summaries), a defined format (numbered list), and constraints (exclude routine visits, word limit).

Scenario 2: A Realtor Writing Listing Descriptions

Bad Prompt:

> Write a listing description for 123 Oak Street.

Claude has no details about the property, no sense of your style, and no idea where this description will be used.

Good Prompt:

> I am a residential realtor in Austin, Texas. I need an MLS listing description for 123 Oak Street.
>
> Here are the property details: 3 bedrooms, 2 bathrooms, 1,850 square feet, built in 2018, recently renovated kitchen with quartz countertops, large backyard with mature trees, walking distance to Zilker Park, listed at $485,000.
>
> Write a listing description between 150 and 200 words. Lead with the strongest selling point (the location near Zilker Park). Use an enthusiastic but not exaggerated tone. Do not use the phrases "dream home," "must see," or "won't last long." End with a call to action to schedule a showing.

Now Claude knows the property, the audience, the desired tone, the length, and what cliches to avoid. The result will be dramatically better.

Scenario 3: A Small Business Owner Categorizing Expenses

Bad Prompt:

> Organize my expenses.

Organize them how? Into what categories? From what source? This will produce something, but probably not what you need.

Good Prompt:

> I run a small landscaping business. Read the file march-transactions.csv, which has columns for date, vendor, amount, and description.
>
> Categorize each transaction into one of these categories: Equipment, Fuel, Labor, Supplies, Insurance, Marketing, or Other. If a transaction does not clearly fit any category, mark it as "Other — needs review."
>
> Output a new CSV file called march-categorized.csv with the original columns plus a new "Category" column. After the CSV, include a summary table showing total spending per category, sorted from highest to lowest.

This prompt gives Claude the source file, the exact categories to use, instructions for edge cases, and a precise description of the desired output format.

The "Describe It to a New Employee" Mental Model

If you are unsure whether your prompt is detailed enough, use this test: imagine you hired someone on their first day. They are smart and capable, but they know nothing about your business, your preferences, or your clients. Would your prompt give them enough information to do the task correctly on the first try?

If not, your prompt needs more detail:

A new employee would not know your preferences. Add them. "I prefer bullet points over paragraphs." "Round all dollar amounts to the nearest whole number."

A new employee would not know your clients. Add context. "This audience is non-technical, so avoid industry terminology."

A new employee would not know your systems. Be specific. "The data is in Q1-revenue.csv in the reports folder." "Save the output as summary-march.txt."

A new employee would ask clarifying questions. Anticipate them and answer in your prompt.

This works because Claude is like that capable new employee — intelligent and able to follow complex instructions, but unaware of your specific situation unless you tell it.

How to Give Cowork Your Data

Cowork can work with many types of files. The way you provide data depends on which tool you are using.

In Cowork (desktop app): Connect a local folder on your computer. Cowork can then read any files in that folder — CSVs, PDFs, text files, and more. You can also connect Google Drive to give Cowork access to your cloud documents.

In Claude Code (for code-heavy tasks): Connect a GitHub repository. Claude Code can read and work with the files in your repo directly.

Once your data is connected, reference files by name in your prompts. Here is how to get the best results with common file types.

CSV and Spreadsheet Exports

CSV files are the most reliable way to give Cowork structured data. Most software — QuickBooks, Excel, Google Sheets, your EMR, your CRM — can export data as CSV. Save the file in your connected folder and reference the file name in your prompt. Always mention what the columns contain.

Tip: If your spreadsheet has multiple tabs, export each tab as a separate CSV file.

PDF Files

Cowork can read PDF files, which is useful for contracts, statements, and reports. Save the PDF in your connected folder and reference it by name. For long PDFs, tell Claude which pages or sections to focus on.

Tip: Digital PDFs work much better than scanned images. Use the digital version whenever possible.

Email Exports

Most email applications let you export messages as text files or save them to a folder. Export the relevant emails into your connected folder. Be specific: "read all emails in the inbox-export folder and list any that require my response."

Tip: Only export the emails relevant to the task. There is no need to share your entire inbox.

Plain Text and Notes

Save notes and documents in your connected folder and reference them by name. The simpler the format, the better — plain text works better than complex formatted documents.

How to Iterate When the First Result Is Not Quite Right

Your first attempt will rarely be perfect, and that is completely normal. The key is knowing how to iterate efficiently.

Identify what is wrong. Before changing anything, pinpoint the specific problem. Is the output too long? Wrong tone? Missing information? Wrong format? Including things you do not want? Name the issue before fixing it.

Adjust one thing at a time. Resist the urge to rewrite the entire prompt. Change one element, run the task again, and evaluate. This way you know exactly what fixed the problem.

Use "more like this, less like that" language. Reference what Claude already produced:

- "The summary was good, but make the tone more formal."
- "Keep the same structure, but add a section at the end for action items."
- "The categorization is correct, but round the totals to the nearest dollar."

Save your refined prompt. Once a prompt produces great results, save it. When you set up the same task as a scheduled Cowork job, use the refined version, not your original draft.

For simple tasks, expect one to two rounds of refinement. For complex tasks, expect two to four. If you are still struggling after five rounds, break the task into smaller pieces.

Prompt Templates: 10 Ready-to-Use Starting Points

Below are ten fill-in-the-blank templates you can copy and customize. Replace the bracketed sections with your own details. These are starting points — you will likely refine them after seeing the first results.

#	Use Case	Prompt Template
1	**Daily Summary**	"Read the file [FILE NAME] which contains [DESCRIPTION OF DATA]. Produce a summary of the top [NUMBER] most important items. Format as a numbered list. Each item should be [NUMBER] sentences. Exclude anything related to [EXCLUSION CRITERIA]."
2	**Email Draft**	"I am a [YOUR ROLE] and I need to respond to the email in [FILE NAME]. The email is from [SENDER CONTEXT]. Draft a reply that [WHAT THE REPLY SHOULD ACCOMPLISH]. Keep the tone [TONE]. Keep the length under [WORD COUNT] words."
3	**Expense Categorization**	"Read the file [FILE NAME] which contains transactions with columns for [COLUMN NAMES]. Categorize each transaction into one of these categories: [LIST YOUR CATEGORIES]. Output a new CSV file called [OUTPUT FILE NAME] with the original columns plus a Category column. Add a summary table at the end showing totals per category."
4	**Client Follow-Up Letter**	"I am a [YOUR PROFESSION] writing to [CLIENT NAME] after [EVENT/MEETING]. Summarize the key points we discussed: [LIST KEY POINTS]. Outline the next steps: [LIST NEXT STEPS]. Use a [TONE] tone. Format as a professional letter. Keep it under [LENGTH]."
5	**Meeting Prep Brief**	"I have a meeting with [WHO] about [TOPIC] at [TIME]. Read the files [FILE NAMES] for background. Produce a one-page briefing document that includes: key facts, open questions, and recommended talking points. Format with headers for each section."
6	**Weekly Report**	"Read all files in the [FOLDER NAME] folder. These contain [DESCRIPTION OF CONTENTS] from the past week. Produce a weekly summary report that includes: [LIST SECTIONS YOU WANT]. Format with headers. Keep the total report under [LENGTH]. Highlight any items that need immediate attention."
7	**Data Cleanup**	"Read the file [FILE NAME] which contains [DATA DESCRIPTION]. Clean the data by: [LIST CLEANUP STEPS, e.g., removing duplicates, standardizing date formats, fixing capitalization]. Save the cleaned data as [OUTPUT FILE NAME]. List any rows that could not be cleaned automatically."
8	**Document Comparison**	"Read the two files [FILE 1] and [FILE 2]. These are [DESCRIPTION, e.g., two versions of a contract]. Compare them and list all differences. For each

#	Use Case	Prompt Template
		difference, show the text from both versions side by side. Highlight any differences that change the meaning or obligations."
9	**Invoice Generation**	"Read the file [FILE NAME] which contains [SERVICE/BILLING DATA] with columns for [COLUMN NAMES]. For each [CLIENT/CUSTOMER], generate an invoice that includes: [LIST INVOICE ELEMENTS]. Format each invoice as [FORMAT]. Save all invoices in the [OUTPUT FOLDER] folder. Name each file [NAMING CONVENTION]."
10	**Content Repurposing**	"Read the file [FILE NAME] which contains [CONTENT TYPE, e.g., a blog post, a presentation, meeting notes]. Rewrite this content as a [NEW FORMAT, e.g., social media post, email newsletter, one-page handout]. Target audience is [AUDIENCE]. Keep the tone [TONE]. Length should be [LENGTH]."

Putting It All Together: A Complete Example

Say you are a small business owner who wants to automate your weekly expense review. Here is the process from start to finish.

Write the initial prompt:

> I run a small consulting firm with 5 employees. Every week, I export our credit card transactions from Chase as a CSV and save it in the weekly-expenses folder as chase-export-YYYY-MM-DD.csv. The columns are: Transaction Date, Post Date, Description, Category, Type, Amount, and Memo.
>
> Read the most recent CSV in that folder. Categorize each transaction into: Travel, Meals, Software, Office Supplies, Professional Services, or Other. Flag any transaction over $500 as "Large — needs review." Produce a summary table with totals per category, list the top 3 largest transactions, and compare total spending to a $5,000 weekly budget. Save the report as weekly-expense-review-[DATE].txt. Keep the tone brief and factual.

Run it and review. The first run categorizes "Uber" as "Other" when you want it under "Travel." So you add: "Treat all rideshare services (Uber, Lyft) as Travel. Treat all restaurant and food delivery charges as Meals."

Save and schedule. Once the output looks right, set this up as a Cowork task running every Friday at 4 PM. From that point forward, your expense review is waiting for you every Friday afternoon — without you lifting a finger.

Key Takeaways

- **Include all four components** — context, task, format, and constraints — for the best results.
- **Use the new employee test.** If someone with no background knowledge could follow your prompt, it is good enough.
- **Start simple and refine.** Iterate in small steps rather than rewriting from scratch.
- **Be specific about files and formats.** Tell Claude exactly where to find data and how to structure output.
- **Save your refined prompts.** You will reuse them when setting up scheduled Cowork tasks.
- **Break complex tasks into pieces.** If a prompt tries to do too many things, split it into smaller prompts.

In the next chapter, we will walk through setting up your first Cowork task step by step.

Profession-Specific Prompts

The prompts earlier in this guide work for almost anyone. But the real power of AI automation shows up when you tailor prompts to the exact work you do every day. A prompt that knows you need HIPAA-safe language, or that your clients expect a specific contract format, or that your monthly close follows a particular checklist — that is transformative.

This chapter gives you twenty-five ready-to-use prompts across five professional categories. Copy the prompt, replace the parts in [BRACKETS] with your own details, and paste it into Cowork to get started.

These prompts work in **Cowork** for document generation, data analysis, and report creation — the tasks most knowledge workers need every day. They can also be adapted for **Claude Code** if you want to build custom tools or automated pipelines around them.

Every item inside [SQUARE BRACKETS] is a placeholder. Replace it with your actual information. The more specific you are, the better your results. For example, "[YOUR SPECIALTY]" works fine if you write "family medicine" — but even better if you write "family medicine with a focus on geriatric patients in a rural clinic setting."

Healthcare

These five prompts target the administrative tasks that pull healthcare professionals away from patient care.

1. Patient Research Digest

This prompt creates a focused research summary you can scan quickly between appointments.

> You are a medical research assistant helping a [YOUR SPECIALTY] physician stay current. I need a research digest on the following topic:
>
> **Topic:** [CONDITION OR TREATMENT — e.g., "recent advances in GLP-1 receptor agonists for type 2 diabetes management"]
>
> Please provide: 1. A plain-language summary of the current standard of care (2-3 sentences) 2. Three to five recent developments or study findings worth knowing about, with enough

> detail that I can decide whether to read the full study 3. Any changes to clinical guidelines in the past 12 months 4. Practical takeaways — what, if anything, should I consider changing in my practice?
>
> Keep the total length under 500 words. Use clinical terminology where appropriate but explain any newer terms. Do not include patient-specific advice — this is for my professional development only.

2. Lab Results Queue

This prompt helps you triage a batch of lab results by urgency so nothing critical gets missed.

> I am a [YOUR SPECIALTY] provider reviewing a batch of lab results. I will paste the results below. For each patient's results, please:
>
> 1. Flag any values that are outside the normal reference range
> 2. Categorize urgency as HIGH (needs same-day follow-up), MODERATE (follow up within the week), or ROUTINE (address at next scheduled visit)
> 3. Suggest a brief, plain-language talking point I can use when calling or messaging the patient about their results
> 4. Note if any result combinations suggest a pattern worth investigating further
>
> **Patient results:** [PASTE LAB RESULTS HERE — remove or replace any patient identifiers if needed for privacy]
>
> Present the output as a table with columns: Patient Initials, Test, Value, Reference Range, Urgency, Suggested Talking Point.

3. Prior Authorization Tracking

This prompt helps you build and maintain a prior authorization tracking document.

> I need to create a prior authorization tracking sheet for my [YOUR SPECIALTY] practice. Here are the current pending authorizations:
>
> [LIST EACH AUTHORIZATION WITH: Patient initials, insurance carrier, procedure or medication requiring auth, date submitted, any reference or case numbers you have]

> For each entry, please: 1. Organize them by insurance carrier 2. Calculate how many business days each has been pending 3. Flag any that have exceeded [NUMBER — e.g., "10"] business days as "FOLLOW UP NEEDED" 4. Draft a brief, professional follow-up script I can use when calling the insurance company, including the key information they will ask for 5. Create a summary count: how many are pending, how many need follow-up, and how many were submitted this week
>
> Format as a clean table I can paste into a spreadsheet or print for my office manager.

4. CME Tracking

This prompt helps you take stock of your CME progress and plan what you still need.

> I am a [YOUR SPECIALTY] provider licensed in [YOUR STATE]. My current CME cycle runs from [CYCLE START DATE] to [CYCLE END DATE]. Here is what I have completed so far:
>
> [LIST COMPLETED CME ACTIVITIES: topic, number of credits, date completed, accrediting body]
>
> Please: 1. Total up my completed credits by category (e.g., general, specialty-specific, ethics, opioid prescribing, etc.) 2. Based on typical [YOUR STATE] requirements for [YOUR SPECIALTY], tell me what categories I may still need to fulfill and how many credits remain in each 3. Suggest specific topic areas that would be both useful for my practice and would satisfy remaining requirements 4. Create a timeline showing how many credits I need to earn per month between now and my cycle end date to stay on track
>
> Note: I understand you may not have the exact current requirements for my state. Please provide your best understanding and remind me to verify with my licensing board.

5. Schedule Gap Analysis

This prompt helps you spot patterns in scheduling gaps so you can address them proactively.

> I am going to share my appointment schedule data for the past [TIME PERIOD — e.g., "4 weeks"]. Please analyze it for patterns and opportunities.
>
> [PASTE SCHEDULE DATA — can be a simple list of dates, times, appointment types, and whether slots were filled, cancelled, or no-showed]

> Please provide: 1. My overall fill rate (percentage of available slots that had patients) 2. No-show rate and any patterns (certain days, times, or appointment types with higher no-show rates) 3. Cancellation patterns — when are cancellations happening and how much notice is typically given? 4. My busiest and slowest days/times 5. Three specific, actionable recommendations to improve my fill rate based on the patterns you see
>
> Present the data as a brief report with a summary section at the top that I can share with my office manager.

Legal

Legal professionals spend significant time on research, document review, and deadline management. These prompts streamline the administrative side so you can focus on substantive legal work.

1. Case Research Summary

This prompt gives you an organized research starting point that you can verify and expand upon.

> You are a legal research assistant. I need a research summary on the following legal issue:
>
> **Jurisdiction:** [STATE OR FEDERAL JURISDICTION] **Area of law:** [e.g., "employment discrimination," "landlord-tenant," "contract dispute"] **Specific question:** [YOUR LEGAL QUESTION — e.g., "What are the elements required to establish a claim of constructive dismissal in Texas?"]
>
> Please provide: 1. A clear statement of the general legal rule or standard that applies 2. Key elements or factors that courts typically consider 3. Any notable trends or recent developments in how this issue is being decided 4. Potential counterarguments or defenses I should be aware of 5. Suggested search terms I can use to find relevant cases in Westlaw or LexisNexis
>
> Important: This is a starting point, not a final legal opinion. Flag areas where the law is unsettled. Do not fabricate case citations — if you are not certain a case exists, describe the principle instead.

2. Contract Clause Extractor

This prompt turns a full contract into a focused summary of the clauses that matter most.

> I am going to paste a contract below. Please extract and summarize the following types of clauses. For each one you find, quote the relevant language and then provide a plain-language summary of what it means.
>
> **Clauses to look for:** [LIST THE CLAUSE TYPES — e.g., "indemnification, limitation of liability, termination rights, non-compete, assignment, governing law, dispute resolution, confidentiality, force majeure"]
>
> For each clause found: 1. Quote the exact contract language 2. Provide a plain-language explanation (suitable for a client who is not an attorney) 3. Flag anything unusual, one-sided, or potentially problematic 4. Note if any standard clauses from my list are MISSING from this contract
>
> **Contract text:** [PASTE THE FULL CONTRACT TEXT HERE]
>
> At the end, provide a brief overall assessment: Is this contract generally balanced, or does it favor one party? What are the top three items I should negotiate?

3. Billing Hours Digest

This prompt organizes and reviews your time entries for completeness and accuracy.

> I am going to paste my raw time entries for [TIME PERIOD — e.g., "this week" or "the month of October"]. Please organize them into a clean billing summary.
>
> **My billing rate:** [YOUR HOURLY RATE] **Client/Matter format:** [HOW YOU IDENTIFY CLIENTS — e.g., "Client Name - Matter Number"]
>
> **Raw time entries:** [PASTE YOUR TIME ENTRIES — include date, client/matter, description of work, and time spent]
>
> Please: 1. Group entries by client and matter 2. Total hours and dollar amounts per client/matter 3. Rewrite any vague descriptions to be more specific and professional (e.g., change "worked on brief" to "Drafted argument section of motion for summary judgment") 4. Flag any entries that seem unusually short or long for the described task 5. Provide a grand total of billable hours and revenue for the period 6. Note any days where my total hours seem low — these might indicate missed entries

> Format as a clean table I can review before submitting to our billing system.

4. Client Follow-Up Tracker

This prompt helps you build a follow-up system so no client falls through the cracks.

> I am going to give you information about my current active clients and their status. Please create a follow-up tracking report.
>
> **Active clients and current status:** [LIST EACH CLIENT WITH: name or initials, matter type, last contact date, what you are waiting on or what the next step is, any deadlines]
>
> Please: 1. Sort clients by urgency — who needs attention first? 2. Flag any client I have not contacted in more than [NUMBER — e.g., "14"] days 3. Identify any upcoming deadlines within the next 30 days 4. Draft a brief, professional follow-up email template for clients I am waiting to hear back from — customize it to each client's situation 5. Create a simple daily action list: who should I contact today, tomorrow, and later this week?
>
> Format as a report I can print or save as a reference document.

5. Deadline Calendar Builder

This prompt builds a comprehensive deadline calendar from your case information.

> I need to build a deadline calendar for my active cases. Here is the information for each case:
>
> [FOR EACH CASE, PROVIDE: case name or identifier, jurisdiction, case type, filing date, current status, any known deadlines, and any court orders setting specific dates]
>
> Please: 1. List all known deadlines in chronological order 2. Based on the case type and jurisdiction, suggest additional deadlines I should be aware of (e.g., discovery cut-offs, motion deadlines, statute of limitations dates) 3. For each deadline, suggest a "preparation start date" — the date I should begin working on the task to have it ready on time 4. Flag any deadlines that fall within the next 14 days as URGENT 5. Note any potential conflicts — deadlines in different cases that fall on the same day or very close together
>
> Important: These are suggestions based on general rules. Remind me in your output to verify all deadlines against my jurisdiction's specific rules and any applicable court orders.

> Format as a clean calendar-style table with columns: Date, Case, Deadline Type, Description, Prep Start Date, Urgency Level.

Finance and Accounting

Financial professionals deal with large volumes of data and strict regulatory requirements. These prompts help you process, organize, and communicate financial information more efficiently.

1. Client Portfolio Summary

This prompt speeds up client meeting preparation by organizing portfolio data and talking points.

> I am a [YOUR ROLE — e.g., "financial advisor," "wealth manager"] preparing for a client review meeting. Here is the current portfolio information:
>
> **Client:** [CLIENT NAME OR INITIALS] **Portfolio details:** [PASTE HOLDINGS, ALLOCATION, AND PERFORMANCE DATA] **Client profile:** [RISK TOLERANCE, TIME HORIZON, INVESTMENT GOALS] **Last meeting date:** [DATE]
>
> Please create a meeting prep summary that includes: 1. A high-level portfolio overview (total value, overall return since last meeting, year-to-date return) 2. Asset allocation breakdown with percentages — is it still aligned with the client's risk profile? 3. Top three performing and bottom three performing holdings 4. Any rebalancing recommendations based on drift from target allocation 5. Three talking points or discussion topics for the meeting based on current market conditions and this client's specific situation 6. A plain-language summary paragraph I could include in a follow-up email to the client after our meeting
>
> Write client-facing portions in non-technical language. Keep analytical portions detailed for my own preparation.

2. Tax Document Checklist

This prompt creates a customized tax document checklist based on a client's specific situation.

> I am preparing to file taxes for a client with the following profile:

Filing status: [e.g., "Married Filing Jointly"] **Income sources:** [LIST — e.g., "W-2 employment, rental income from two properties, small business (Schedule C), investment accounts at two brokerages"] **Major life events this year:** [e.g., "sold a home, had a baby, started a side business"] **State(s):** [STATE(S) OF RESIDENCE AND/OR INCOME]

Please create a comprehensive document checklist that includes: 1. Every tax form and document they need to gather, organized by category 2. The expected source of each document (who sends it and when it typically arrives) 3. A status column I can use to check off received items 4. Any documents that are commonly forgotten or overlooked for their specific situation 5. A list of questions I should ask the client before starting the return

Format as a client-facing checklist I can send directly, plus a separate internal version with the questions and reminders.

3. Monthly P&L Digest

This prompt translates a P&L statement into plain language a business owner can understand and act on.

I am going to paste a monthly profit and loss statement for a [TYPE OF BUSINESS — e.g., "dental practice," "restaurant," "e-commerce company"]. Please analyze it and create a digestible summary.

P&L Data: [PASTE THE P&L STATEMENT]

Comparison period: [e.g., "Compare to same month last year" or "Compare to last month" — include prior period data if available]

Please provide: 1. A three-sentence executive summary: How did the business do this month? 2. Revenue analysis: total revenue, change from comparison period, and any notable trends 3. Expense analysis: top five expense categories, any that increased or decreased significantly, and any that seem out of line for this type of business 4. Profit margins: gross margin, operating margin, and net margin — are they healthy for this industry? 5. Three specific questions or action items the business owner should consider based on these numbers

Write the summary in language a business owner (not an accountant) would understand. Avoid jargon, or define it when you must use it.

4. Invoice Aging Report

This prompt helps you organize outstanding invoices and prioritize collections.

> I need an invoice aging analysis. Here are my outstanding invoices:
>
> [LIST EACH INVOICE WITH: invoice number, client name, invoice date, amount, due date, any partial payments received, any notes about communication]
>
> Please: 1. Categorize each invoice into aging buckets: Current (not yet due), 1-30 days past due, 31-60 days past due, 61-90 days past due, 90+ days past due 2. Total the amount outstanding in each bucket 3. Calculate the percentage of total receivables in each bucket 4. Flag the top five highest-risk invoices (based on amount and age) that need immediate attention 5. Draft a professional but firm collection email for each aging category (current reminder, first past-due notice, second notice, and final notice) 6. Provide an overall assessment: Is my collections health good, concerning, or critical?
>
> Format the aging data as a table, followed by the email templates.

5. Regulatory Update Tracker

This prompt helps you stay informed about regulatory developments that affect your practice or clients.

> I am a [YOUR ROLE — e.g., "CPA," "financial advisor," "CFO"] focused on [YOUR AREAS — e.g., "small business taxation," "retirement planning," "nonprofit accounting"]. I need a regulatory update summary.
>
> **Areas of interest:** [LIST SPECIFIC REGULATORY AREAS — e.g., "IRS guidance on pass-through deductions, SEC filings requirements for small funds, state sales tax nexus rules"]
>
> **Time period:** [e.g., "the past 30 days" or "Q1 2025"]
>
> Please provide: 1. A summary of any significant regulatory changes, proposed rules, or guidance issued in my areas of interest 2. For each item: what changed, who it affects, when it takes effect, and what action (if any) I need to take 3. Any upcoming comment periods, effective dates, or compliance deadlines I should know about 4. A brief assessment of how each change might affect my clients or my practice
>
> Note: Your information has a knowledge cutoff. Please indicate the timeframe of your knowledge and remind me to verify with primary sources (Federal Register, IRS.gov, or my

> professional association).

Real Estate

Real estate professionals live and die by responsiveness, market knowledge, and follow-up. These prompts help you stay on top of your pipeline and client relationships.

1. Listing Research Digest

This prompt creates a research brief to prepare you for a listing appointment or buyer consultation.

> I am a real estate [AGENT/BROKER] in [YOUR MARKET AREA]. I need a research digest for the following property or listing opportunity:
>
> **Property address or area:** [ADDRESS OR NEIGHBORHOOD] **Property type:** [e.g., "single-family residential," "multi-family," "commercial retail"] **Approximate details:** [SQUARE FOOTAGE, BEDROOMS/BATHS, LOT SIZE, YEAR BUILT — whatever you know] **Purpose:** [e.g., "preparing a listing presentation," "advising a buyer," "evaluating an investment opportunity"]
>
> Please provide: 1. A summary of the neighborhood or area — what is it known for, who typically lives or works there, and what are the key selling points? 2. Factors that could positively or negatively affect value (schools, transit, development plans, environmental considerations) 3. Suggested talking points I can use with the seller or buyer 4. Questions I should ask or research further before the appointment 5. If this is a listing appointment: five specific marketing angles I could highlight for this property
>
> Keep the tone professional and suitable for client-facing materials.

2. Client Follow-Up Sequence

This prompt creates a multi-touch follow-up plan with actual messages you can send.

> I need a follow-up communication sequence for the following real estate contact:
>
> **Contact name:** [NAME] **Contact type:** [e.g., "buyer lead from open house," "past client," "expired listing," "FSBO," "sphere of influence"] **What I know about them:** [ANY DETAILS —

budget, timeline, preferred area, family situation, how we met] **Last contact:** [DATE AND WHAT WAS DISCUSSED]

Please create a follow-up sequence that includes: 1. An immediate follow-up message (email or text) I can send today 2. A follow-up for one week from now 3. A follow-up for two weeks from now 4. A follow-up for one month from now 5. A long-term nurture message for three months from now

For each message: - Keep it personal, warm, and not pushy - Reference something specific about their situation - Include a clear but soft call to action - Vary the format (some emails, some texts, suggest a phone call for at least one)

Also suggest one "value add" I could send at each stage (a market report, a relevant article, a new listing alert, etc.).

3. Market Comp Report

This prompt organizes comp data into a clear, client-ready market analysis report.

I need to create a comparative market analysis summary for a client. Here is the subject property and comparable sales data:

Subject property: [ADDRESS, BEDS/BATHS, SQUARE FOOTAGE, LOT SIZE, YEAR BUILT, CONDITION, ANY UPGRADES OR ISSUES]

Comparable sales: [FOR EACH COMP: address, sale price, sale date, beds/baths, square footage, lot size, year built, condition, days on market, any relevant notes]

Please: 1. Create a side-by-side comparison table of the subject property versus each comp 2. Note the key differences between each comp and the subject property (adjustments a client should understand) 3. Calculate price per square foot for each comp 4. Suggest a price range for the subject property based on the comps, with a brief explanation of how you arrived at it 5. Write a client-friendly narrative summary (3-4 paragraphs) explaining the market analysis in plain language 6. Include any caveats — what factors might push the price higher or lower than the suggested range?

Format as a report suitable for a listing presentation or buyer consultation.

4. Transaction Deadline Tracker

This prompt creates a comprehensive deadline tracking sheet across all your active transactions.

> I have the following active real estate transactions. Please create a comprehensive deadline tracking sheet.
>
> [FOR EACH TRANSACTION: property address, client name or initials, transaction type (buy/sell/lease), contract date, closing date, and any specific deadlines already established (inspection period, financing contingency, appraisal deadline, etc.)]
>
> Please: 1. List all known deadlines in chronological order across all transactions 2. Based on standard [YOUR STATE] real estate transaction timelines, suggest any common deadlines I may not have listed (e.g., title search completion, HOA document review period, final walkthrough) 3. Flag any deadlines in the next 7 days as URGENT 4. Flag any deadlines where two transactions have conflicts on the same day 5. For each deadline, include a brief note about what needs to happen and whose responsibility it is (mine, the other agent's, the lender's, the title company's, etc.)
>
> Format as a table with columns: Date, Property, Deadline, Action Required, Responsible Party, Urgency.
>
> Note: I will verify all deadlines against my actual contracts. This is a planning tool, not a substitute for reading the contract.

5. Lead Scoring Summary

This prompt helps you evaluate and prioritize your lead pipeline by likelihood to convert.

> I need help prioritizing my current real estate leads. Here is my pipeline:
>
> [FOR EACH LEAD: name or initials, source (referral, website, open house, cold call, etc.), first contact date, property interest (buying, selling, or both), budget/price range, timeline, level of engagement so far (responsive, somewhat responsive, gone quiet), any notes about motivation or urgency]
>
> Please: 1. Score each lead on a scale of 1-10 based on likelihood to transact in the next 90 days, using the information provided 2. Explain your reasoning for each score in one sentence 3. Group leads into three categories: HOT (pursue aggressively this week), WARM (maintain regular contact), and COLD (long-term nurture only) 4. For each HOT lead, suggest a specific next action I should take 5. For each COLD lead, suggest whether they

> are worth continuing to nurture or should be deprioritized 6. Give me a "pipeline health" summary: Am I generating enough leads? Is my pipeline balanced across the funnel?
>
> Format as a weekly review report for planning my prospecting and follow-up activities.

General Business

These prompts work across industries and save you time on tasks you probably do every week.

1. Weekly KPI Digest

This prompt creates an executive-level KPI summary from your raw weekly numbers.

> I run a [TYPE OF BUSINESS] and need a weekly KPI summary. Here are this week's numbers:
>
> [PASTE YOUR KEY METRICS — e.g., revenue, number of new customers/clients, website traffic, conversion rate, customer support tickets, inventory levels, social media followers, email open rates, or whatever you track]
>
> **Comparison data:** [LAST WEEK'S NUMBERS AND/OR SAME WEEK LAST YEAR, if available]
>
> Please: 1. Create a clean KPI dashboard showing each metric, its current value, the change from last period (both absolute and percentage), and a simple indicator (UP, DOWN, or FLAT) 2. Highlight the top two "wins" — metrics that improved meaningfully 3. Flag the top two "watch items" — metrics that declined or are trending in a concerning direction 4. Provide a brief (3-4 sentence) narrative summary I can share with my team or partners 5. Suggest one specific action I should consider based on this week's data
>
> Keep it concise. I want to review this in under five minutes.

2. Employee Hours Summary

This prompt organizes timesheet data into an actionable summary ready for payroll review.

> I need to review employee hours for [TIME PERIOD — e.g., "the pay period ending March 7"]. Here are the raw timesheet entries:

> [PASTE TIMESHEET DATA — employee names, dates, hours worked, overtime if applicable, any PTO or sick time, project or department codes if used]
>
> Please: 1. Total regular hours and overtime hours for each employee 2. Flag any employee approaching or exceeding [NUMBER — e.g., "40"] hours (potential overtime) 3. Flag any employee with unusually low hours who may have missed entries 4. Break down hours by project or department if that data is available 5. Identify any patterns — are certain days understaffed? Is overtime concentrated with specific people? 6. Provide a total labor cost estimate based on [HOURLY RATE OR RATE STRUCTURE — e.g., "all employees at $25/hour regular, $37.50/hour overtime"]
>
> Format as a summary table followed by any flags or observations.

3. Vendor Invoice Tracker

This prompt organizes vendor invoices by priority and helps you plan cash flow.

> I need to organize and track my vendor invoices. Here are my current outstanding invoices and recent payments:
>
> [LIST EACH INVOICE: vendor name, invoice number, invoice date, due date, amount, description of goods/services, any payment terms (Net 30, Net 60, early payment discount, etc.), payment status]
>
> Please: 1. Sort all invoices by due date 2. Flag anything due within the next 7 days as URGENT 3. Identify any invoices where an early payment discount is available and calculate the savings versus paying on the due date 4. Total upcoming payments by week for the next four weeks so I can plan cash flow 5. Flag any vendors where I have multiple outstanding invoices — this might indicate a billing issue or a need to consolidate 6. Create a brief payment schedule recommendation: what should I pay today, what can wait, and where can I save money?
>
> Format as a prioritized payment table followed by a four-week cash flow forecast for vendor payments.

4. Meeting Notes to Action Items

This prompt turns messy meeting notes into organized minutes with clear action items.

> I am going to paste my notes from a meeting. Please turn them into organized meeting minutes with clear action items.
>
> **Meeting:** [MEETING NAME OR TYPE — e.g., "Weekly Team Standup," "Q1 Planning Session," "Client Kickoff"] **Date:** [DATE] **Attendees:** [LIST OF PEOPLE IN THE MEETING]
>
> **My raw notes:** [PASTE YOUR NOTES — these can be messy, shorthand, bullet points, whatever you captured during the meeting]
>
> Please: 1. Write a brief summary of the meeting (3-5 sentences covering the main topics and outcomes) 2. Organize the discussion points by topic with brief summaries of what was said or decided 3. Extract every action item mentioned or implied, and for each one list: - What needs to be done (specific and clear) - Who is responsible (based on context from the notes; write "UNASSIGNED" if unclear) - Deadline (if mentioned; write "TBD" if not) 4. Note any decisions that were made 5. Note any items that were tabled or need follow-up discussion
>
> Format so I can send it directly to attendees as a follow-up email.

5. Competitor Monitoring Digest

This prompt organizes competitive intelligence into a structured analysis with actionable recommendations.

> I run a [TYPE OF BUSINESS] in [YOUR MARKET/LOCATION]. I want to organize what I know about my competitors and identify gaps in my knowledge.
>
> **My main competitors:** [LIST EACH COMPETITOR WITH: name, what they offer, their approximate size or market position, and anything you have noticed recently (new products, pricing changes, marketing campaigns, hires, news mentions, customer complaints you have heard about)]
>
> Please: 1. Create a competitor comparison matrix showing each competitor's strengths and weaknesses relative to my business 2. Identify areas where competitors appear to be gaining ground or trying something new 3. Highlight any potential threats — things competitors are doing that could take business from me 4. Identify opportunities — gaps in the market that no competitor (including me) is filling well 5. Suggest three specific things I should monitor over the next month (and where to look for that information — websites, social media, review

> sites, industry publications, etc.) 6. Recommend one defensive action (protect my current position) and one offensive action (gain ground)
>
> Keep the analysis practical and specific to my market — I need actionable intelligence, not generic business strategy.

10 Real-World Use Cases

Every automation in this chapter follows a simple principle: describe the problem clearly, let the right tool build the solution, and set it to run on its own. Nine of these ten use cases are built entirely inside Claude Cowork — the desktop app you already have open. You point Cowork at your local files, describe what you need, and schedule it. No terminal. No command line. No technical knowledge required. The tenth use case — the personal finance dashboard — uses Claude Code, the web-based coding tool at claude.ai/code, because it builds an entire web application from scratch. That one is the showpiece, and we will give it the attention it deserves.

These are not hypothetical examples. Each one is based on a real professional workflow, with the exact prompt you would type, a description of what gets built, and an honest estimate of how much time it saves.

Read through all ten, but pay special attention to the ones closest to your own work. Then pick one — just one — and build it today.

Use Case 1: Patient Background Research Digest

Persona: Dr. Sarah Mitchell, Internist — Riverside Medical Group

Tool: Claude Cowork

Dr. Sarah Mitchell sees between 18 and 24 patients a day. Every morning before her first appointment, she spends 30 to 45 minutes pulling up each patient's chart in the EHR, scanning through recent lab results, checking medication lists, reviewing the reason for today's visit, and mentally assembling a picture of where each patient stands. It is not difficult work. It is the kind of work that makes her feel like a very expensive data entry clerk.

The real problem is not the time — it is the cognitive cost. By the time she has reviewed her seventh or eighth chart, her attention is fragmented. She has been context-switching between a diabetic patient whose A1C has been creeping up, a post-surgical follow-up who may need imaging, and a new patient transferring from another practice whose records arrived as a 40-page PDF. She hasn't seen a single patient yet, and she is already mentally tired. On days when her morning is rushed, she skips this prep entirely and walks into appointments cold, spending the first

two minutes of each visit re-orienting herself while the patient watches her scroll through their chart on the screen. She knows it doesn't inspire confidence.

Why Cowork: Sarah's data lives in local files on her computer — CSV exports from the scheduling system and text files with patient notes. She does not need a web application or custom software. She needs Cowork to read those files and produce a document. That is exactly what Cowork does.

The Prompt

> I have a CSV export from our scheduling system that lists today's patients. The columns are: patient_id, patient_name, appointment_time, visit_reason, and last_visit_date.
>
> I also have a folder called `patient_notes/` that contains one text file per patient, named by patient_id (e.g., `P-4821.txt`). Each file contains their recent lab results, current medications, allergies, and notes from their last three visits.
>
> Build me a morning briefing document that I can read before clinic starts. For each patient on today's schedule, in appointment order, show: - Patient name and appointment time - Reason for today's visit - Days since last visit - Current medication list (just the names and doses, as a compact list) - Any flagged lab values from the most recent results (anything outside normal range, highlighted clearly) - A 2-3 sentence summary of the trend from their last three visit notes — are they improving, stable, or getting worse? - Any allergies, displayed prominently so I cannot miss them
>
> Make it clean and scannable. I will be reading this on my iPad between patients. Use large text for patient names and appointment times. Group morning and afternoon patients with a visual divider.

What Claude Builds

Cowork reads the scheduling CSV, matches each patient to their notes file, and produces a clean briefing document. Each patient appears as a card-style section with their name in large, bold text alongside the appointment time. The visit reason is displayed in a colored label — blue for follow-ups, green for wellness visits, orange for urgent concerns. Medications run in two compact columns. Lab values outside normal ranges are highlighted with the abnormal value displayed next to the reference range. The three-visit trend summary appears as a short narrative paragraph, and allergies are displayed in a prominent red banner at the top of each patient's section so they are impossible to overlook.

The entire briefing is generated from files Sarah already has. No database, no server, no special software — just files she already produces as part of her daily workflow, reorganized into something useful.

How It Gets Scheduled

Sarah sets this up as a scheduled task in Cowork that runs every weekday at 5:45 AM, fifteen minutes before she typically starts her morning prep. She configures it once in Cowork's scheduling panel — select the days, set the time, and point it at the folder. By the time she arrives at the office and opens her iPad, the briefing is already waiting in her synced folder.

Time Saved

3-4 hours per week. Sarah's morning prep drops from 30-45 minutes to about 5 minutes of reviewing a document that has already done the chart-scanning work for her. She also reports that the quality of her first minutes with each patient improved noticeably — she walks in already knowing the story.

Use Case 2: Personal Finance Dashboard

The "I Can't Believe AI Built This" Example

Tool: Claude Code (claude.ai/code)

This use case deserves special attention because it represents the moment when most people realize what is actually possible. We are not talking about a document that summarizes your expenses. We are talking about a fully functional, interactive financial dashboard — the kind of thing you would expect to pay a software company a monthly subscription for — built entirely from a single conversation with Claude Code.

Think of Claude Code as hiring a developer to build you a custom tool. You go to claude.ai/code in your web browser, connect your GitHub account (Claude Code walks you through this — it takes two minutes), and then describe what you want built. Claude Code writes the actual software, creates the project, and deploys it so you can open it in your browser. You do not need to understand what it is building. You just need to describe what you want.

The professional in this case is someone who has been meaning to "get a handle on their finances" for years. They have bank statements downloading into CSV files. They have a vague

sense that they are spending too much on subscriptions. They tried Quicken, found it overwhelming. They tried Mint before it shut down. They tried building a spreadsheet, but it became a maintenance chore within two weeks. What they actually want is a single screen they can look at once a week that tells them where their money is going, what is recurring, and whether they are trending in the right direction.

This is not a toy example. The dashboard Claude Code builds is a real, working web application that runs with your own data. Nobody else sees it. No bank login credentials are shared with any service. You export CSVs from your bank's website — something every bank supports — and the dashboard reads those files. That's it.

The Prompt

This prompt is typed into Claude Code at claude.ai/code, not into Cowork. You are talking to a coding tool, so you can be ambitious about what you ask for.

> I have CSV files exported from my bank accounts. They are in a folder called `bank_exports/` and each file has columns: date, description, amount, and balance. Negative amounts are debits, positive are credits.
>
> Build me a personal finance dashboard as a local web app I can open in my browser. I want:
>
> 1. **A top row of KPI cards** showing: total spending this month, total income this month, net cash flow, and spending compared to last month (up or down, by how much, in both dollars and percent)
> 2. **A spending-over-time chart** — bar chart showing daily or weekly spending for the current month, with a line overlay showing the cumulative trend
> 3. **A spending breakdown by category** — you will need to auto-categorize transactions based on the description. Use common sense categories: Groceries, Dining, Transportation, Healthcare, Subscriptions, Utilities, Shopping, Entertainment, and Other. Show this as a donut chart with dollar amounts and percentages
> 4. **A recurring charges detector** — find transactions that appear monthly with similar amounts and descriptions. Show these in a table with the merchant name, typical amount, and how many months it has recurred. Flag any that changed in amount recently
> 5. **An API and subscription cost tracker** — pull out anything that looks like a software subscription, cloud service, or API charge (things like AWS, OpenAI, Anthropic, Netlify, GitHub, Heroku, domain registrars, SaaS tools). Show these separately with monthly cost and annual projection

6. **A transaction list** at the bottom — searchable, sortable, with the auto-assigned category shown as a colored tag. Let me click a category tag to filter the whole dashboard to just that category

Make it look professional. Use a clean, modern design — dark sidebar, white content area, subtle shadows on the cards. The charts should use a cohesive color palette. Everything should be responsive so I can check it on my phone too.

What Claude Code Builds

This is where things get impressive. Claude Code produces a complete, working dashboard application — not a document, not a spreadsheet, but actual software. Here is what you get.

Finance Dashboard

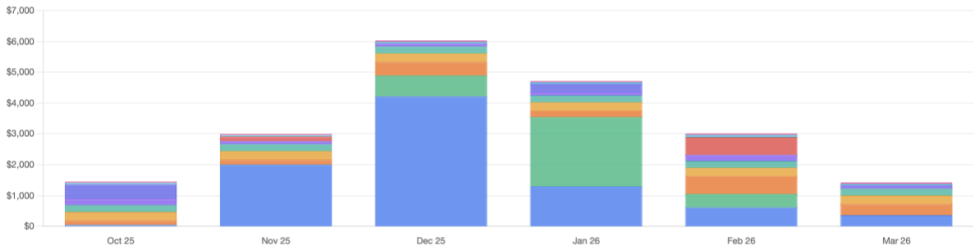
115 transactions

Last fetch: 2026-03-13

FROM 10/17/2025 TO 03/12/2026 | 7D | 30D | 90D | All | Accounts All | Search transactions...

THIS MONTH SPEND	VS LAST MONTH	INCOME	SAVINGS RATE
$1,421.13	-52.8%	$18,500.00	92.3%
$3,010.68 last month	↓ Spending less	$18,500.00 last month	↑ Healthy rate

Monthly Trend



Spending by Category

Transactions 115 | Recurring 13 | AI/API Costs 3

DATE	MERCHANT	AMOUNT	CATEGORY	ACCOUNT
Mar 12, 2026	Riverside Medical Group RIVERSIDE MEDICAL GROUP PAYROLL	+$18,500.00	Income	
Mar 11, 2026	Whole Foods Market WHOLE FOODS MARKET	$142.50	Food & Drink	
Mar 10, 2026	Nobu NOBU RESTAURANT	$68.40	Food & Drink	
Mar 10, 2026	Wells Fargo Mortgage MORTGAGE PAYMENT - WELLS FARGO	$3,200.00	Loan Payments	
Mar 9, 2026	Starbucks STARBUCKS	$18.50	Food & Drink	
Mar 8, 2026	BMW Financial Services BMW FINANCIAL SERVICES	$650.00	Loan Payments	
Mar 8, 2026	Netflix NETFLIX	$24.99	Entertainment	
Mar 8, 2026	Spotify SPOTIFY	$19.99	Entertainment	
Mar 7, 2026	Shell SHELL OIL	$47.20	Transportation	
Mar 7, 2026	Anthropic ANTHROPIC	$89.00	Services	
Mar 6, 2026	Equinox EQUINOX FITNESS	$220.00	Personal Care	
Mar 5, 2026	Nordstrom NORDSTROM	$312.80	Shopping	
Mar 5, 2026	OpenAI OPENAI	$15.00	Services	
Mar 4, 2026	Florida Power & Light FLORIDA POWER AND LIGHT	$185.00	Bills & Utilities	
Mar 4, 2026	Comcast Xfinity COMCAST XFINITY	$94.00	Bills & Utilities	
Mar 3, 2026	Trader Joe's TRADER JOES	$128.75	Food & Drink	
Mar 2, 2026	Amazon AMAZON PRIME	$55.00	Shopping	

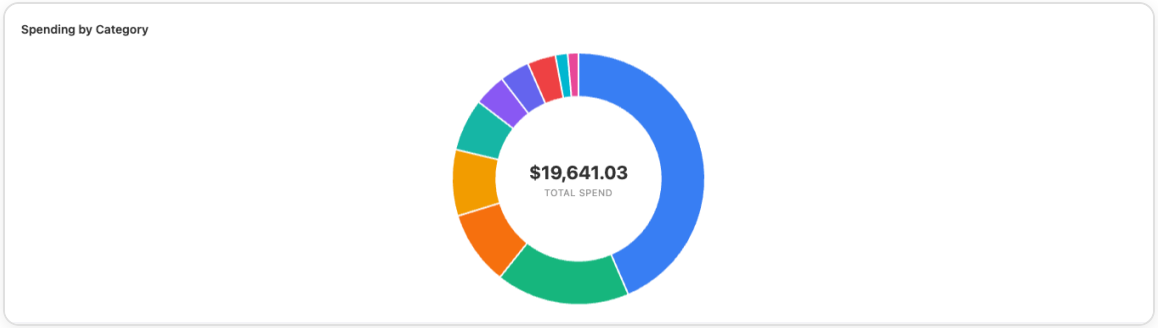
Mar 1, 2026	Chase Savings Transfer TRANSFER TO SAVINGS	$500.00	Transfer Out
Feb 28, 2026	Riverside Medical Group RIVERSIDE MEDICAL GROUP PAYROLL	+$18,500.00	Income
Feb 27, 2026	Wells Fargo Mortgage MORTGAGE PAYMENT - WELLS FARGO	$3,200.00	Loan Payments
Feb 26, 2026	BMW Financial Services BMW FINANCIAL SERVICES	$650.00	Loan Payments
Feb 25, 2026	Delta Airlines DELTA AIRLINES	$247.30	Travel
Feb 24, 2026	Marriott MARRIOTT HOTELS	$189.00	Travel
Feb 23, 2026	Whole Foods Market WHOLE FOODS MARKET	$78.50	Food & Drink
Feb 22, 2026	Netflix NETFLIX	$24.99	Entertainment
Feb 22, 2026	Spotify SPOTIFY	$19.99	Entertainment
Feb 21, 2026	Anthropic ANTHROPIC	$89.00	Services
Feb 20, 2026	OpenAI OPENAI	$15.00	Services
Feb 19, 2026	Costco COSTCO WHOLESALE	$163.40	Food & Drink
Feb 18, 2026	Equinox EQUINOX FITNESS	$220.00	Personal Care
Feb 17, 2026	Shell SHELL OIL	$52.00	Transportation
Feb 16, 2026	Florida Power & Light FLORIDA POWER AND LIGHT	$185.00	Bills & Utilities
Feb 16, 2026	Comcast Xfinity COMCAST XFINITY	$94.00	Bills & Utilities
Feb 15, 2026	Dr. Jennifer Walsh Dermatology DR JENNIFER WALSH DERMATOLOGY	$425.00	Medical
Feb 14, 2026	The Capital Grille THE CAPITAL GRILLE	$312.00	Food & Drink
Feb 13, 2026	Starbucks STARBUCKS	$18.50	Food & Drink
Feb 12, 2026	Saks Fifth Avenue SAKS FIFTH AVENUE	$560.00	Shopping
Feb 11, 2026	Amazon AMAZON PRIME	$55.00	Shopping
Feb 10, 2026	CVS Pharmacy CVS PHARMACY	$145.00	Medical
Feb 9, 2026	Uber UBER	$22.00	Transportation
Feb 8, 2026	Adobe ADOBE CREATIVE CLOUD	$95.00	Services
Feb 7, 2026	Chase Savings Transfer TRANSFER TO SAVINGS	$500.00	Transfer Out
Jan 31, 2026	Riverside Medical Group RIVERSIDE MEDICAL GROUP PAYROLL	+$18,500.00	Income
Jan 30, 2026	Wells Fargo Mortgage MORTGAGE PAYMENT - WELLS FARGO	$3,200.00	Loan Payments
Jan 29, 2026	BMW Financial Services BMW FINANCIAL SERVICES	$650.00	Loan Payments
Jan 28, 2026	Apple Store APPLE STORE	$1,240.00	Shopping
Jan 27, 2026	Anthropic ANTHROPIC	$89.00	Services
Jan 26, 2026	Netflix NETFLIX	$24.99	Entertainment
Jan 26, 2026	Spotify SPOTIFY	$19.99	Entertainment
Jan 25, 2026	Florida Power & Light FLORIDA POWER AND LIGHT	$185.00	Bills & Utilities

Showing 1–50 of 115 transactions

‹ 1 2 3 ›

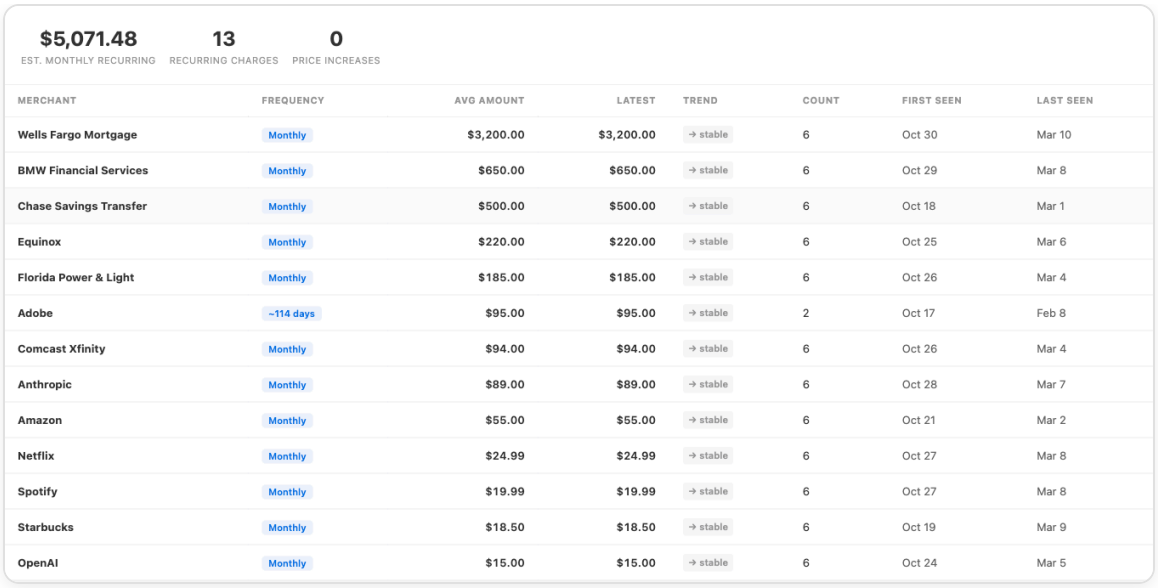
Dashboard Overview

The full dashboard presents a polished, professional interface that looks like it belongs to a commercial fintech product. The top row displays four KPI cards: total spending, total income, net cash flow, and a month-over-month comparison with a green or red arrow indicating the trend direction. Below the KPI cards, a bar chart visualizes spending patterns across the current month, with each bar representing a day's outflows and a smooth trend line overlaid to show whether spending is accelerating or leveling off. The transaction list at the bottom shows every transaction with its date, description, assigned category (displayed as a color-coded pill), and amount. A search bar at the top of the list lets you type a merchant name to instantly filter. Clicking any category pill filters the entire dashboard — every chart, every KPI — down to just that category.



Dashboard Donut

The spending breakdown donut chart is immediately useful. Each slice represents a category — Groceries in green, Dining in orange, Transportation in blue, Subscriptions in purple — with the dollar amount and percentage displayed in a clean legend beside the chart. Hovering over any slice highlights it and shows the exact figure. For most people, this chart alone is worth the ten minutes it takes to set up, because it answers the question "where is my money actually going?" in a way that bank statements never do. The person who built this discovered they were spending $847 per month on dining — a number that genuinely surprised them because it was distributed across dozens of small transactions that never felt significant individually.

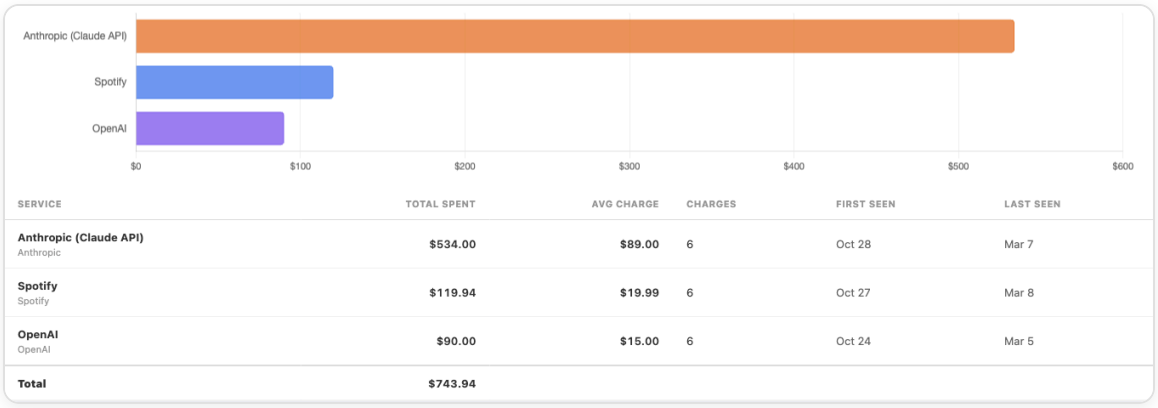


$5,071.48 EST. MONTHLY RECURRING | **13** RECURRING CHARGES | **0** PRICE INCREASES

MERCHANT	FREQUENCY	AVG AMOUNT	LATEST	TREND	COUNT	FIRST SEEN	LAST SEEN
Wells Fargo Mortgage	Monthly	$3,200.00	$3,200.00	→ stable	6	Oct 30	Mar 10
BMW Financial Services	Monthly	$650.00	$650.00	→ stable	6	Oct 29	Mar 8
Chase Savings Transfer	Monthly	$500.00	$500.00	→ stable	6	Oct 18	Mar 1
Equinox	Monthly	$220.00	$220.00	→ stable	6	Oct 25	Mar 6
Florida Power & Light	Monthly	$185.00	$185.00	→ stable	6	Oct 26	Mar 4
Adobe	~114 days	$95.00	$95.00	→ stable	2	Oct 17	Feb 8
Comcast Xfinity	Monthly	$94.00	$94.00	→ stable	6	Oct 26	Mar 4
Anthropic	Monthly	$89.00	$89.00	→ stable	6	Oct 28	Mar 7
Amazon	Monthly	$55.00	$55.00	→ stable	6	Oct 21	Mar 2
Netflix	Monthly	$24.99	$24.99	→ stable	6	Oct 27	Mar 8
Spotify	Monthly	$19.99	$19.99	→ stable	6	Oct 27	Mar 8
Starbucks	Monthly	$18.50	$18.50	→ stable	6	Oct 19	Mar 9
OpenAI	Monthly	$15.00	$15.00	→ stable	6	Oct 24	Mar 5

Dashboard Recurring

The recurring charges detection table is the feature that pays for itself. The dashboard scans transaction descriptions for patterns — same merchant name appearing monthly, amounts within a few dollars of each other — and assembles them into a clean table. Each row shows the merchant name, the typical monthly amount, when it first appeared, and how many consecutive months it has been charged. A yellow warning icon appears next to any charge whose amount changed recently, so you can spot price increases immediately. The person who built this dashboard found three subscriptions they had forgotten about entirely — a meditation app, a cloud storage upgrade, and a professional association membership they no longer used — totaling $67 per month, or $804 per year in money that was simply evaporating.



SERVICE	TOTAL SPENT	AVG CHARGE	CHARGES	FIRST SEEN	LAST SEEN
Anthropic (Claude API) Anthropic	$534.00	$89.00	6	Oct 28	Mar 7
Spotify Spotify	$119.94	$19.99	6	Oct 27	Mar 8
OpenAI OpenAI	$90.00	$15.00	6	Oct 24	Mar 5
Total	**$743.94**				

Dashboard Api Costs

The API and subscription cost tracker separates technology spending into its own focused view. For anyone who uses cloud services, AI tools, SaaS products, or developer platforms — and increasingly, that includes non-technical professionals who are using tools exactly like the ones in this guide — this panel shows each service, its monthly cost, a sparkline showing cost trends over recent months, and an annualized projection. It answers the question every professional eventually asks: "How much am I actually spending on all these digital tools?" The annualized column is particularly sobering. A $20/month charge does not feel like much until you see it displayed as $240/year alongside fifteen other similar charges.

The entire dashboard runs locally. It reads CSV files from a folder on your computer. No data is uploaded anywhere. No account is needed. No monthly fee. You described what you wanted, and Claude Code built it — like hiring a developer for a ten-minute conversation instead of a ten-thousand-dollar contract.

How It Works After the Build

Unlike the Cowork use cases in this chapter, the finance dashboard is a one-time build. You go to claude.ai/code once, have the conversation, and Claude Code creates the project and deploys it. After that, the dashboard is yours. To update it with new data, you download your latest bank CSV (or set up your bank's auto-export if it supports it), drop it into the `bank_exports/` folder, and refresh the page. The dashboard reads the new data automatically.

If you want to add features later — maybe a savings goal tracker or a net worth chart — you go back to Claude Code and describe what you want added. It picks up where you left off.

Time Saved

2-3 hours per week compared to manually reviewing bank statements and maintaining spreadsheets. But the real value is not time — it is visibility. Most people do not track their finances because the friction is too high. This removes the friction entirely. The person who built this dashboard reported that their discretionary spending dropped by 15% in the first month, not because they set a budget, but simply because they could finally see where the money was going.

Use Case 3: Weekly Expense Categorization and Summary Email

Persona: Marcus Reeves, Independent Financial Advisor — Reeves Wealth Management

Tool: Claude Cowork

Marcus runs a small advisory firm with three employees. Every week, his office manager Lisa exports credit card and bank transactions into a CSV and manually categorizes each one — client entertainment, office supplies, software subscriptions, travel, marketing. She then builds a summary in Excel and emails it to Marcus. The process takes Lisa about 90 minutes every Friday afternoon, and the results are inconsistent because she sometimes categorizes the same merchant differently from week to week. When tax season arrives, Marcus spends an entire weekend re-categorizing transactions because the year's data is full of inconsistencies.

What Marcus actually needs is a system that applies the same categorization rules every single time, learns from corrections, and delivers a clean summary without anyone manually touching a spreadsheet.

Why Cowork: All of this data lives in local files — CSV exports and a rules file on Marcus's computer. Cowork reads the files, applies the rules, and generates the output. No coding tool needed.

The Prompt

> I have a CSV file of this week's business transactions in `weekly_transactions/`. Columns are: date, description, amount, account_name.
>
> I also have a file called `category_rules.json` that maps merchant keywords to expense categories. For example, "Staples" maps to "Office Supplies" and "Delta" maps to "Travel."
>
> Categorize every transaction using those rules. For any transaction that does not match a rule, make your best guess based on the merchant name and add the new rule to `category_rules.json` so it is automatic next time.
>
> Then generate a summary document that includes: - Total spending by category, sorted highest to lowest, with a simple horizontal bar chart - Any individual transaction over $500, highlighted for my review - A comparison to the same week last year if that data exists in `weekly_transactions/archive/` - A list of any new categorization rules you created, so I can review them
>
> Save the categorized transactions back to a new CSV with the category column added.

What Claude Builds

Cowork reads the weekly CSV, applies the categorization rules from the JSON file, handles unrecognized merchants by making an intelligent guess and recording the new rule, and generates both a categorized CSV file and a formatted summary document. The summary includes a horizontal bar chart showing spending by category, a highlighted section for large transactions, and a year-over-year comparison table when historical data is available. New categorization rules are listed at the bottom so Marcus can approve or adjust them. The `category_rules.json` file grows smarter every week — after a month, almost every transaction is categorized automatically.

How It Gets Scheduled

Marcus opens Cowork's scheduling panel and creates a task that runs every Friday at 3:00 PM. Lisa drops the exported CSV into the folder anytime during the week, and the automation picks it up, processes it, and saves the summary. Marcus reviews it before he leaves for the weekend. No scripts to run, no commands to type — Lisa saves the file, Cowork does the rest.

Time Saved

1.5-2 hours per week. Lisa's Friday afternoon categorization task is eliminated. More importantly, the year-end data is already clean, saving Marcus an entire weekend during tax season.

Use Case 4: Email Drafting and Follow-Up Workflow

Persona: Vanessa Torres, Residential Real Estate Agent — Compass Realty

Tool: Claude Cowork

Vanessa closes about 30 transactions a year, which means she is simultaneously managing between 8 and 12 active clients at any given time, each at a different stage of the buying or selling process. Her biggest operational headache is follow-up communication. After a showing, she needs to send the buyer a summary of the properties they visited with her notes on each one. After a listing appointment, she needs to send the seller a comparative market analysis recap. After an inspection, she needs to email the buyer with a plain-language summary of the findings and recommended next steps. After closing, she needs to send a thank-you and request a review.

None of these emails are difficult to write. But there are dozens of them every week, they need to feel personal and specific to each client, and forgetting to send even one can damage a

relationship. Vanessa estimates she spends six to eight hours per week writing emails that follow predictable patterns.

Why Cowork: Vanessa's notes live in folders on her computer. She does not need a web application — she needs Cowork to scan those folders, figure out which clients need an email, and draft them. Cowork works directly with her local files.

The Prompt

> I have a folder called `client_files/` with a subfolder for each active client. Each subfolder contains my notes as text files — showing_notes.txt, inspection_report.txt, listing_details.txt, etc.
>
> I also have a file called `email_templates.json` that defines my email types: post_showing, post_listing_appointment, post_inspection, post_closing, and check_in. Each template has a tone guide (warm but professional), required sections, and example phrases I like to use.
>
> Do the following: 1. Scan each client folder for new or updated note files since the last run 2. Determine which email type is appropriate based on what files are present 3. Draft the email using the template structure and my tone guide, incorporating specific details from the notes (property addresses, prices, features I highlighted, inspection findings) 4. Save each draft as a file in a `drafts/` folder, named with the client name and email type 5. Create a summary file listing all drafts generated, so I can review and send them
>
> The drafts should sound like me, not like a robot. Use the example phrases from my templates and vary the language so two clients who toured the same property do not get identical emails.

What Claude Builds

Cowork acts as Vanessa's first-draft writing assistant. It scans client folders, detects the stage of each transaction based on which documents are present, and generates personalized email drafts. Each draft incorporates specific details — the address on Maple Street with the updated kitchen that the Johnsons loved, or the inspection finding about the HVAC system at the Oak Lane property. Drafts are saved as files she can review, copy into her email client, personalize with a final sentence or two, and send. A summary document lists every draft generated so nothing falls through the cracks.

How It Gets Scheduled

Vanessa creates two scheduled tasks in Cowork: one runs every morning at 7:00 AM and one runs every evening at 6:00 PM. When she saves her notes after showings or inspections, the next scheduled run picks them up and has drafts ready. She reviews and sends them during her morning coffee or after dinner — a ten-minute task instead of an hour-long writing session. Setting up the schedule takes about 30 seconds in Cowork's scheduling panel.

Time Saved

5-6 hours per week. Vanessa's email writing drops from 6-8 hours to about 1-2 hours of reviewing and personalizing drafts that are already 90% complete.

Use Case 5: Contract and Document Summarization

Persona: David Chen, Corporate Attorney — Chen & Associates

Tool: Claude Cowork

David's firm handles contract review for mid-sized businesses. A typical engagement involves reviewing commercial leases, vendor agreements, employment contracts, and partnership documents — often 40 to 80 pages each. David can read and analyze a contract thoroughly, but the first pass — identifying the key terms, flagging unusual clauses, and producing a summary memo for the client — takes two to three hours per document. His paralegal handles some of this, but the output still requires David's review, and the quality varies depending on who prepared it and how rushed they were.

What David needs is a consistent first-pass analysis that identifies the important provisions, flags anything unusual, and produces a client-ready summary memo that he can review and sign off on in 20 minutes instead of building from scratch.

Why Cowork: David's contracts are files on his computer. He needs Cowork to read each document and produce a structured summary — a document-in, document-out workflow that is exactly what Cowork excels at.

The Prompt

> I have a contract saved in `contracts/incoming/`. It is a vendor services agreement.

> Produce a contract summary memo that includes: - **Parties and effective date** - **Term and renewal provisions** — is it auto-renewing? What is the notice period for termination? - **Payment terms** — amounts, schedule, late payment penalties - **Liability and indemnification** — caps on liability, who indemnifies whom, carve-outs - **Non-compete and exclusivity provisions** — any restrictions on either party - **Termination clauses** — what triggers termination, what are the consequences - **Unusual or non-standard provisions** — anything that deviates from a typical vendor agreement of this type, flagged clearly with a brief explanation of why it is unusual
>
> Use a professional memo format with my firm's header. The audience is a business executive, not a lawyer — explain legal terms in plain language where possible. At the end, include a section called "Points for Negotiation" listing 3-5 provisions I might recommend the client push back on, with brief reasoning.
>
> Save the summary to `contracts/summaries/` with the same filename.

What Claude Builds

Cowork reads the contract text, identifies and extracts key provisions, and generates a professionally formatted summary memo. Unusual clauses are flagged with explanatory annotations. The "Points for Negotiation" section provides actionable recommendations in plain language. The memo format is consistent every time, regardless of which paralegal or associate feeds in the document — a level of consistency that is nearly impossible to achieve when the summaries are written manually by different people.

How It Gets Scheduled

David's paralegal drops incoming contracts into the `contracts/incoming/` folder. A Cowork scheduled task runs every 30 minutes during business hours — David sets this up once in the scheduling panel by selecting the time range and interval. When a new contract appears, the summary is generated and placed in `contracts/summaries/` within minutes. David reviews it at his convenience. No commands to run, no notifications to configure — the summary simply appears in the folder.

Time Saved

6-8 hours per week. David reviews three to four contracts per week. Each summary that previously took 2-3 hours to build now takes 15-20 minutes to review and finalize. His paralegal's

time is freed up for higher-value work like client communication and filing preparation.

Use Case 6: Meeting Notes to Action Items Pipeline

Persona: Rachel Okonkwo, Owner — Okonkwo Strategic Consulting

Tool: Claude Cowork

Rachel runs a seven-person consulting firm. Her team has four to six client meetings per day, and after each one, someone — usually Rachel — needs to extract the action items, assign them to team members, and update the project tracker. The meetings are recorded and auto-transcribed by Zoom, so the raw material exists. But a 45-minute meeting produces a 6,000-word transcript that nobody wants to read. The action items are buried in conversational back-and-forth, hedged language ("we should probably look into..."), and tangential discussions. Rachel spends an hour each evening extracting action items from the day's transcripts, and she still misses things.

Why Cowork: The transcripts are text files saved to Rachel's computer. She needs Cowork to read through conversational text and produce structured outputs — summaries, action item tables, and email drafts. This is a reading-and-writing task, not a software-building task.

The Prompt

> I have meeting transcripts in `meetings/transcripts/`. Each file is named with the date and client name (e.g., `2025-01-15_meridian_partners.txt`). The transcripts include speaker labels.
>
> For each new transcript, produce: 1. A **meeting summary** (3-5 bullet points covering the key topics discussed and decisions made) 2. An **action items list** as a table with columns: action item, owner (inferred from the conversation — who said "I'll do this" or was asked to do it), deadline (if mentioned, otherwise "TBD"), and priority (High/Medium/Low based on the urgency expressed in the conversation) 3. A **follow-up email draft** to the client summarizing what was discussed and confirming the next steps
>
> Save the summary and action items as a structured JSON file in `meetings/processed/` so I can import them into our project management tool. Also save the email draft in `meetings/emails/`.
>
> Skip any transcripts that already have a matching file in `meetings/processed/`.

What Claude Builds

Cowork processes each transcript, identifying commitments and assignments even when they are expressed informally ("I'll circle back on that pricing question" becomes an action item assigned to whoever said it). The output includes a concise summary, a structured action items list with owners and priorities, and a client-ready follow-up email. The JSON format makes it easy to copy action items into project management tools like Asana, Monday.com, or Notion.

How It Gets Scheduled

Rachel creates a Cowork task that runs every evening at 7:00 PM, processing any transcripts that arrived during the day. Zoom's auto-transcription saves files to her computer, and by the time Rachel sits down the next morning, summaries, action items, and draft follow-up emails are waiting for review. She sets up a second scheduled task that runs Friday afternoons to generate a weekly rollup of all action items across all clients, highlighting anything marked High priority or past its deadline. Both tasks are configured in Cowork's scheduling panel in under a minute.

Time Saved

4-5 hours per week. Rachel's evening transcript review is eliminated. Action items are captured more consistently because the automation does not get tired or distracted at the end of a long day.

Use Case 7: Social Media Content Drafting Queue

Persona: Priya Sharma, Owner — Willow & Sage Botanicals

Tool: Claude Cowork

Priya runs a small skincare company that sells through her website and two local boutiques. She knows she needs to post on Instagram and Facebook regularly — her customers expect it, and her sales correlate directly with her social media activity. But creating content is a constant stress. She needs five to seven posts per week across two platforms, each with a caption, relevant hashtags, and a suggestion for what image to pair with it. She spends her Sunday evenings writing the entire week's content, and it takes three to four hours. By Wednesday, she is already behind.

What Priya needs is a system that drafts a full week of content in minutes, aligned with her brand voice, her product calendar, and the seasonal topics her audience cares about.

Why Cowork: Priya's brand guide and product calendar are files on her computer. She needs Cowork to read them and generate a week of content drafts. No web application, no code, no API integrations — just reading files and writing new ones.

The Prompt

> I have a file called `brand_guide.txt` that describes my brand voice (warm, knowledgeable, nature-inspired — we talk about ingredients, self-care rituals, and seasonal skincare). I also have `product_calendar.csv` which lists upcoming product launches, sales, and seasonal collections with dates.
>
> Generate a week of social media content — 7 posts total. For each post, include: - Platform (alternate between Instagram and Facebook) - Caption (150-250 words for Instagram, 80-120 words for Facebook) - 8-10 relevant hashtags for Instagram posts - Image suggestion (describe what photo or graphic would pair well) - Best posting time based on general engagement data - A content category tag: educational, promotional, behind-the-scenes, community, or seasonal
>
> Mix the content categories throughout the week — no two promotional posts in a row. If there is a product launch or sale within the next two weeks on the product calendar, include one post that teases it naturally.
>
> Save the output as a JSON file and also as a nicely formatted calendar document I can print out and pin above my desk.

What Claude Builds

Cowork reads Priya's brand guide and product calendar, then produces a week's worth of platform-specific social media posts. The output includes a JSON file she can use with scheduling tools and a printable calendar showing each day's post with its caption, hashtags, image suggestion, and posting time. The content rotates through categories automatically and weaves in upcoming promotions without making every post feel like an advertisement. Each week's content is fresh because Cowork generates it based on the current product calendar and season, not from a static template.

How It Gets Scheduled

Priya opens Cowork's scheduling panel and creates a task that runs every Saturday at 8:00 AM. When she sits down with her coffee Saturday morning, the next week's content queue is already waiting. She reviews and adjusts the drafts, then schedules them through her social media tool. The entire process takes 30-40 minutes instead of an entire evening.

Time Saved

2.5-3.5 hours per week. Priya's Sunday evening content sessions are replaced by a quick Saturday morning review. More importantly, she actually posts consistently now — she hasn't missed a week since setting up the automation.

Use Case 8: Inventory and Appointment Tracking

Persona: Dr. James Whitfield, Owner — Whitfield Family Dentistry

Tool: Claude Cowork (for reports) — with Claude Code as an option if you want to build a custom tracking tool

Dr. Whitfield runs a dental practice with two hygienists and a front desk coordinator. He has two persistent operational headaches. First, supply inventory: his practice uses dozens of consumable items — gloves, composite resin, impression material, anesthetic cartridges — and they consistently run out of things at inconvenient moments because nobody is systematically tracking usage rates against remaining stock. Second, appointment utilization: he has a vague sense that certain time slots go unfilled more often than others, but he has never had the data laid out clearly enough to adjust his scheduling template.

Both problems have the same root cause: data exists (in purchase orders, supply closet counts, and the scheduling system), but nobody has time to analyze it.

Why Cowork (primarily): The weekly operations report is a perfect Cowork task — read CSV files, analyze the data, and produce a document. Dr. Whitfield's front desk coordinator updates the inventory spreadsheet during her weekly supply closet count, and the scheduling system exports to CSV. Cowork reads those files and generates the report. No coding required.

When to consider Claude Code: If Dr. Whitfield eventually decides he wants an interactive tracking tool — something where his team can click buttons to update inventory counts in real time, or a visual scheduling board — that is a software project. He would use Claude Code at

claude.ai/code to build it, the same way the finance dashboard was built in Use Case 2. But the weekly report itself needs nothing more than Cowork.

The Prompt

> I have two data sources:
>
> 1. `inventory/` — a CSV file updated weekly with columns: item_name, category (e.g., PPE, restorative, anesthetic), current_quantity, unit_cost, and reorder_threshold. I also have `inventory/purchase_history.csv` with past orders: date, item_name, quantity_ordered, total_cost.
> 2. `scheduling/` — CSV exports from our scheduling system with columns: date, time_slot, patient_name (blank if unfilled), procedure_type, duration_minutes, provider.
>
> Build me a weekly operations report that covers:
>
> **Inventory Section:** - Items currently below reorder threshold (flagged prominently) - Items projected to hit reorder threshold within 2 weeks based on average usage rate - Monthly spending on supplies by category, trending over the last 6 months - An automatically generated reorder list with suggested quantities based on typical usage
>
> **Scheduling Section:** - Utilization rate by provider (percentage of available slots filled) - Utilization by day of week and time of day — where are the gaps? - Cancellation and no-show patterns if the data includes status fields - Average revenue per time slot by procedure type
>
> Save the report in `reports/` with the date in the filename.

What Claude Builds

Cowork produces a comprehensive weekly operations report that combines supply chain management with scheduling analytics. The inventory section functions as an early warning system — items approaching reorder thresholds are flagged before they run out, and a suggested purchase order is pre-generated with quantities based on historical usage. The scheduling section reveals patterns: maybe Tuesday afternoons consistently run at 60% capacity while Thursday mornings are overbooked, suggesting a template adjustment. The entire report is a document Dr. Whitfield can review on his tablet between patients.

How It Gets Scheduled

A Cowork task runs every Monday at 6:00 AM. The front desk coordinator updates the inventory CSV during her Friday supply closet count, and the scheduling export is saved to the folder throughout the week. Dr. Whitfield reviews the report during his Monday morning coffee and makes any ordering decisions before the week gets busy. One scheduling configuration in Cowork, and the report generates itself every week indefinitely.

Time Saved

2-3 hours per week in direct analysis time. But the larger savings are indirect: the practice stopped running out of supplies (eliminating emergency rush orders that cost 30-40% more) and identified scheduling patterns that led to a 12% improvement in chair utilization.

Use Case 9: Weekly Business Performance Summary

Persona: Tony Calabrese, Owner — Calabrese's Italian Kitchen

Tool: Claude Cowork

Tony owns a popular neighborhood restaurant with 60 seats and a small catering operation. He has a POS system that tracks every transaction, but the reports it generates are either too granular (transaction-level data he doesn't need) or too generic (monthly summaries that arrive too late to act on). What Tony wants is a weekly report that answers the questions he actually cares about: Are we up or down compared to the same week last year? Which menu items are selling and which are sitting? Is the catering side growing? How is labor cost tracking as a percentage of revenue? He knows these numbers exist somewhere in his POS exports, but extracting them requires an hour with a spreadsheet that he never has time for on a busy restaurant schedule.

Why Cowork: Tony's POS system exports CSVs. His timesheets are in a spreadsheet. Cowork reads both and generates a report. This is the same file-in, document-out pattern that Cowork handles perfectly — no web application, no coding, no technical setup.

The Prompt

> I have POS data exports in `pos_data/` — CSV files with columns: date, time, item_name, category (appetizer, entree, dessert, beverage, catering), quantity, unit_price, total_price, payment_method, server_name. I also have `labor/` with weekly timesheets: employee_name, role, hours_worked, hourly_rate, total_pay.

> Build me a weekly restaurant performance report that includes: - **Revenue summary**: total revenue, dine-in vs. catering split, comparison to same week last year (if data exists) - **Top 10 sellers by quantity and by revenue** — these are often different items and I want to see both - **Bottom 5 sellers** — items I should consider rotating off the menu - **Category breakdown**: what percentage of revenue comes from appetizers, entrees, desserts, and beverages - **Day-of-week analysis**: which days are strongest, which are weakest - **Labor cost ratio**: total labor divided by total revenue, compared to my target of 30%, with a clear indicator (green if under, yellow if 30-33%, red if over) - **Average ticket size** and trend over the last 4 weeks - **Catering growth**: monthly catering revenue trend, last 6 months
>
> Keep it simple and visual. I will look at this for five minutes on Monday morning while I plan the week. Save it in `reports/` with the date.

What Claude Builds

Cowork produces a one-page restaurant performance report that distills POS and labor data into the metrics Tony actually manages by. The revenue comparison answers "how are we doing?" immediately. The top and bottom sellers inform menu decisions. The labor cost ratio — the single number every restaurant owner watches most closely — is displayed prominently with a color-coded indicator. The day-of-week breakdown helps Tony think about staffing and promotions (if Tuesdays are consistently slow, maybe that's the night for a prix fixe special). Everything is designed to be consumed in five minutes by someone who does not want to interact with a spreadsheet.

How It Gets Scheduled

Tony opens Cowork, creates a scheduled task for every Monday at 5:00 AM, and points it at his data folders. His POS system auto-exports daily CSV files, and the timesheets are entered by his manager on Sunday evening. By Monday morning, the report is generated and sitting in his synced folder. He reads it on his phone during his morning espresso before the kitchen staff arrives. Setup took him about two minutes.

Time Saved

1.5-2 hours per week in direct analysis time. Tony previously did this analysis sporadically in a spreadsheet, meaning he often managed by gut feeling rather than data. The automation means he makes data-informed decisions every single week without any additional effort.

Use Case 10: Lead Research and CRM Data Entry Automation

Persona: Keisha Williams, Enterprise Sales — Meridian Software Solutions

Tool: Claude Cowork

Keisha sells enterprise software to mid-market companies. Her pipeline has 40 to 60 active leads at any given time, and for each one, she needs to know the basics before making contact: company size, industry, recent news, key decision-makers, their tech stack if publicly known, and any mutual connections. She currently does this research manually by sifting through her own notes, the company's website printouts she's saved, and news clips she's bookmarked. Each lead takes 15 to 20 minutes to research, and she adds 8 to 10 new leads per week. That is two to three hours just on research — before she has sent a single email or made a single call.

The second half of the problem is CRM data entry. After each call or email exchange, Keisha is supposed to log the interaction in her CRM with a summary, next steps, and an updated deal stage. She falls behind on this weekly, which means her pipeline data is always stale and her manager's forecasts are based on incomplete information.

Why Cowork: Keisha's lead data, contact databases, and notes are files on her computer. She needs Cowork to read through them and produce structured research briefings and CRM-ready records. This is an information synthesis task, not a software development task.

The Prompt

> I have a CSV of new leads in `leads/new_leads.csv` with columns: company_name, contact_name, contact_title, contact_email, source (where the lead came from), and notes.
>
> I also have a folder `leads/research_cache/` where previous research on companies is stored as JSON files. And I have `leads/news_clips/` with saved articles and press releases about various companies, and `leads/contacts_db.csv` with my full contact database.
>
> For each new lead, produce a one-page briefing that includes: - **Company overview**: what they do, approximate size, industry, headquarters location — based on any information available in my files - **Recent news**: any press releases, funding rounds, executive changes, or product launches from my news clips folder - **Key decision-makers**: the contact provided plus any other executives in relevant roles from my contacts database - **Potential pain points**: based on their industry and size, what challenges might they face that our software

> addresses? - **Suggested opening angle**: a 2-3 sentence suggestion for how I might open the first email or call, referencing something specific about their company
>
> Save each briefing in `leads/briefings/` and also generate a summary CSV of all new leads with a "research_status" column I can import into my CRM.
>
> Separately, I have a folder `crm_updates/` where I drop text files with my call notes after each conversation. The files are named with the date and company name. Process these into structured CRM update records: a JSON file for each one with fields for company_name, contact_name, date, interaction_type (call/email/meeting), summary (2-3 sentences), next_steps, and suggested_deal_stage.

What Claude Builds

Cowork produces two things. First, lead research briefings assembled from Keisha's own data sources — her contact database, saved news clips, and cached research from previous leads in the same industry. Each briefing is a clean, one-page document she can scan in two minutes before picking up the phone. The suggested opening angle is based on specifics from her files, not generic boilerplate — if a news clip mentions the company recently announced an expansion, the opener references that.

Second, Cowork processes Keisha's informal call notes (which she can dictate into her phone and save as text files) into structured CRM records. The JSON output matches the fields her CRM expects, so her admin can import them directly instead of re-typing everything. Keisha's handwritten note "Talked to Mike at Pinnacle, he's interested but needs board approval, follow up in 2 weeks" becomes a clean record with all fields populated.

How It Gets Scheduled

Keisha creates two scheduled tasks in Cowork. The first runs every morning at 6:30 AM to process any new leads added to the CSV the previous day, generating briefings before she starts her outreach calls. The second runs every evening at 8:00 PM to process any call notes she dropped into the folder during the day, turning them into structured CRM updates that her admin imports the next morning. Both tasks are set up in Cowork's scheduling panel in under a minute each.

Time Saved

4-5 hours per week. Lead research drops from 15-20 minutes per lead to 2 minutes of reviewing a pre-built briefing. CRM updates go from being a dreaded Friday afternoon catch-up task to

something that happens automatically every night. Keisha's pipeline data stays current, which has the secondary benefit of making her forecasting conversations with her manager dramatically shorter and less contentious.

What These Use Cases Have in Common

Look at these ten examples together and you will notice a pattern. Every single one follows the same three-step structure:

1. **Data already exists somewhere** — in CSV exports, text files, transcripts, notes, or documents that the professional already produces as part of their work.
2. **The transformation is predictable** — the same type of analysis, formatting, or summarization needs to happen every time, following consistent rules.
3. **The output is immediately useful** — a briefing, a dashboard, a report, a set of email drafts — something the professional can act on without further processing.

Nine of these automations were built entirely inside Claude Cowork — the desktop app for knowledge workers. You open Cowork, point it at your files, describe what you want in plain English, and schedule it to run on its own. No terminal. No command line. No npm install. No technical setup of any kind. Cowork works with your local files and produces documents, summaries, and structured data.

The tenth — the personal finance dashboard — used Claude Code at claude.ai/code because it required building actual software: a web application with interactive charts, search functionality, and a responsive design. Think of Claude Code as hiring a developer. You describe the tool you want, and Claude Code builds it. You use it when you need something that goes beyond documents and into custom applications.

Most of your automations will be Cowork tasks. That is by design. The vast majority of professional workflows are about reading information, transforming it, and producing a useful output — and that is exactly what Cowork does.

The professionals in these examples did not become technologists. They became better managers of their own time — better at describing what they needed and reviewing what they got back. That is the skill this guide teaches, and these ten examples are proof that it works across industries, across workflows, and across levels of technical comfort.

Pick the use case closest to your own work. Open Cowork. Start typing. Your first automation is one conversation away.

Scheduling with Cowork

Why Scheduling Matters

Up to this point, every automation you have built requires you to sit down, open the Claude Desktop app, and tell Cowork what to do. That is already a massive time saver. But the real magic happens when your automations run without you — when you wake up on Monday morning and your weekly summary is already waiting, or when your end-of-day report generates itself at 5 PM every Friday while you are driving home.

That is what scheduling does. It turns your one-off automations into reliable, recurring processes that work in the background, the same way your office's heating system keeps the building warm without anyone flipping a switch every morning.

Cowork's scheduling is built right into the Claude Desktop app. Think of it as a calendar app, but instead of scheduling meetings, you are scheduling tasks for Cowork to perform. You describe what to do, when to do it, and how often — and then you walk away.

Dr. Mitchell uses scheduling to run her patient follow-up report every Monday at 7 AM, her billing reconciliation every Friday at 4 PM, and her medication inventory check on the first of every month. Before she set these up, each of those tasks required 30 to 45 minutes of her attention. Now they happen automatically, and she reviews the results over her morning coffee.

The Important Caveat: Your Computer Needs to Be On

Before we go further, there is one critical detail to understand: **scheduled tasks only run when your computer is awake and the Claude Desktop app is open.** Cowork is not a cloud service running on a distant server. It lives on your machine, which means your machine needs to be on and the app needs to be running for tasks to fire on schedule.

If your computer is asleep or shut down when a task is scheduled to run, that run will be missed. The task will not retroactively execute — it will simply wait for the next scheduled time when your computer is awake and the app is open.

For most professionals, this is easy to manage. Keep your computer awake during business hours and leave the Claude Desktop app running in the background (it uses very few resources). Many people configure their computer's energy settings to prevent sleep during work hours specifically

so their scheduled tasks run reliably. If you schedule a task for 6:30 AM, just make sure your computer is on and the app is running by that time.

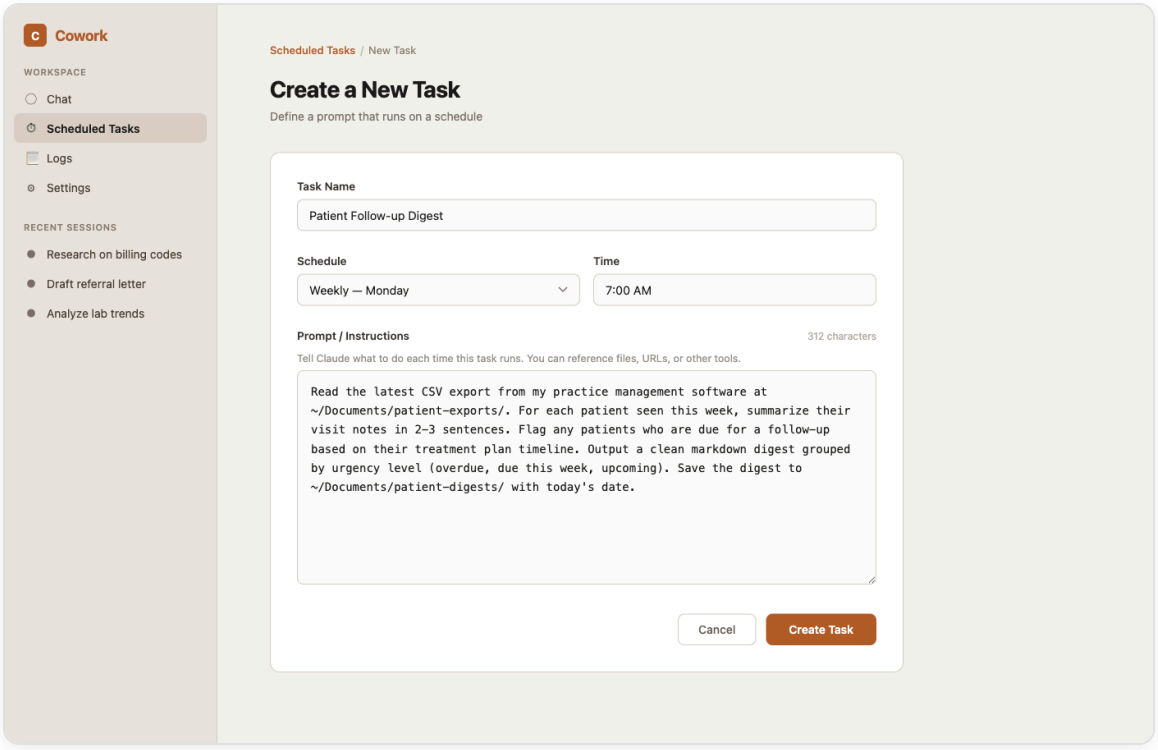
Step-by-Step: Setting Up Your First Scheduled Task

Let us walk through creating your first scheduled task. We will set up a simple weekly report that runs every Monday morning and prepares a summary you can review when you start your day.

Step 1: Open the Claude Desktop App and Create a New Task

Open the Claude Desktop app on your computer. Navigate to the Cowork section, where you will see your active tasks, your task history, and a button to create a new task. If this is your first time, the active tasks section will be empty — that is expected.

Click the "New Task" button to open the task creation form.



Mock Cowork New Task

You will see several fields. Here is what each one means:

Task Name: A short, descriptive name that helps you identify the task later. "Monday Morning Patient Summary" is better than "Weekly Report." Future-you should be able to glance at the name

and know exactly what this task does.

Description (optional): A note about what the task does and why. Helpful when you come back three months later and cannot remember the details.

Prompt: The most important field. This is where you write the instructions Cowork will follow when the task runs. Paste in the same prompt you have already been using successfully. For example, Dr. Mitchell's prompt looks like this:

> Look at the patient visit log in /Documents/Practice/visits.csv. Pull all visits from the past 7 days. Create a summary that shows: total visits, visits grouped by type (new patient, follow-up, urgent), the three busiest days, and any patients who were flagged for follow-up. Save the report as a PDF in /Documents/Practice/Weekly Reports/ with the filename "Weekly Summary - [today's date].pdf"

Notice that this prompt is specific — it names exact file paths, describes the output, and specifies where to save the result. Be as precise as you would be when giving instructions to a new employee on their first day.

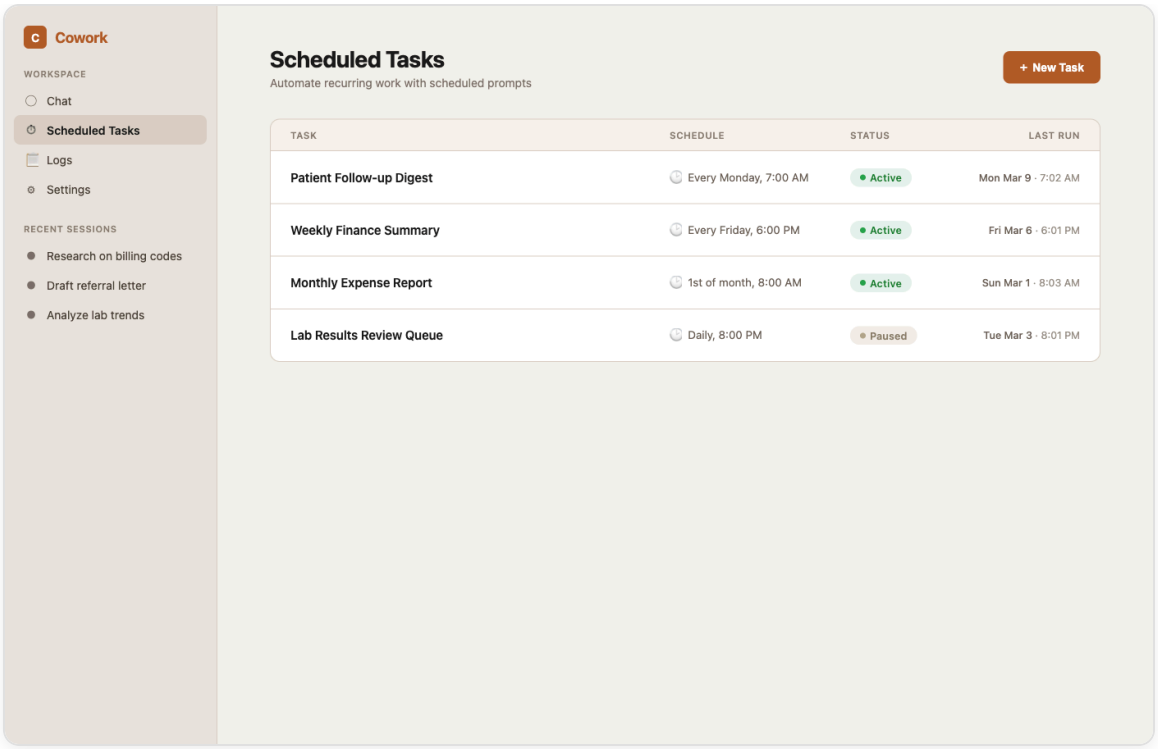
Schedule: Tell Cowork when and how often to run the task. Options include Daily, Weekly, Monthly, and Custom. For our Monday morning report, select "Weekly," choose "Monday," and set the time to 6:30 AM so the report is ready when you arrive. Remember, your computer and the Claude Desktop app need to be running at 6:30 AM for this to work.

Step 2: Test Before You Schedule

Before you activate the schedule, click the "Run Now" or "Test" button. This executes the task immediately so you can verify the output. Check that it pulled the right data, formatted it correctly, and saved it to the right location. If something is off, edit the prompt and test again. Spend ten minutes refining now rather than discovering three weeks later that your reports have been wrong.

Step 3: Activate the Schedule

Once the test looks good, toggle the task to "Active." Your task is now live and will run automatically at the time you specified.



Mock Cowork Tasks

Check the output after the first real scheduled run to confirm everything works outside of test mode.

How to Connect an Existing Automation to a Schedule

If you have already built something useful in a Cowork conversation, you do not need to rebuild it. You just need to connect what you have to a schedule. Here is the process:

1. Find the prompt that works. Go back to the Cowork conversation where you built the automation and copy the prompt that produced the result you were happy with. Use it exactly as-is.

2. Check for file path assumptions. Make sure your prompt uses full, specific file paths. Instead of "look at my spreadsheet," write "look at /Documents/Business/Q1-Revenue.xlsx." Scheduled tasks cannot ask you for clarification, so every detail needs to be in the prompt.

3. Handle date references. Make sure the prompt uses relative language that will be correct on any run date. "The past 7 days" works every time. "The week of March 3rd" only works once. This is one of the most common mistakes people make when moving from manual to scheduled automations.

4. Specify the output clearly. Your prompt needs to explicitly say where to put the result. A scheduled task runs without you watching, so it needs to know exactly where to save its output — a specific folder, a specific filename pattern, a specific format.

5. Paste the prompt into a new scheduled task, set your schedule, test it, and activate.

Best Practices: When to Schedule Daily, Weekly, or Monthly

Choosing the right frequency matters. Run a task too often and you create noise. Run it too rarely and you miss the point of automation.

Schedule daily when the underlying data changes every day and you need to act on the results the same day. Examples: morning dashboard refresh, daily appointment summary, overnight email triage summary via the Gmail connector.

Schedule weekly when the data is meaningful in weekly chunks and you have an existing weekly rhythm. Examples: weekly client activity report, case status updates for your Monday meeting, weekly inventory check.

Schedule monthly when the task involves large datasets or feeds a monthly process like billing or compliance. Examples: monthly billing reconciliation, regulatory compliance checks, client retention analysis.

Rule of thumb: If you are not sure, start weekly. After a month, ask yourself: "Did I wish I had this more often?" Switch to daily. "Did some reports sit unread?" Switch to monthly. Let your own behavior guide the frequency.

How to Pause, Edit, and Delete Tasks

All task management happens inside the Claude Desktop app. There is no need to open a browser, navigate to a separate website, or type any commands.

Pausing a Task

Find the task in your Cowork task list and look for the "Pause" or "Disable" toggle. This stops the schedule but keeps all settings intact. When you are ready to resume, flip it back on. Pausing is always better than deleting when the break is temporary, because deleting means recreating everything from scratch.

Going on vacation? Pause your tasks before you leave and reactivate them when you return. This prevents a backlog of reports nobody is reading and saves processing resources.

Editing a Task

Click any task to open its settings. You can change the name, prompt, schedule, or output destination. After making changes, always run a test before reactivating. Common reasons to edit: file paths changed, you want more detail in the output, you want to shift the schedule to a different time, or a data source format changed.

Deleting a Task

If a task is no longer useful, delete it through the menu on the task card. Before you delete, copy the prompt and save it somewhere — in a text file, a notes app, anywhere you can find it later. Rebuilding a prompt from memory is frustrating and usually produces a worse result than the original.

Troubleshooting: What to Do When a Scheduled Task Fails

Scheduled tasks will occasionally fail. This is normal. Here are the most common problems and how to fix them, all from within the Claude Desktop app.

The task ran but produced wrong or empty output. The data it was looking for has probably changed — a file was moved, renamed, or reformatted. Open the task, click the "Run Now" button to test it manually, check the file or folder the prompt references, update the prompt if needed, and reactivate.

The task did not run at all. First, check the obvious: was your computer awake and was the Claude Desktop app open at the scheduled time? If your laptop went to sleep at 6 AM and the task was scheduled for 6:30 AM, it simply did not fire. Adjust your computer's energy settings or shift the task to a time when your machine is reliably awake. If the computer was on and the app was running, check whether the task was accidentally paused. Look at the run history for attempted runs that failed versus no attempts at all.

The task produces an error message. Read the error message carefully — they are often more helpful than people expect. "File not found" means you should check the file path. "Column not found" means your spreadsheet structure changed. Fix the issue, test with "Run Now," and reactivate.

The output used to be right but is now wrong. The underlying data has drifted. Compare a recent output to an older correct one, identify what changed in the data, update your prompt to handle the new format, and test.

General Troubleshooting Steps

When you are stuck, follow these steps in order:

1. **Run the task manually** using the "Run Now" button to determine if the problem is with the schedule or the task itself.
2. **Read any error messages** and write down the exact text.
3. **Check the input data** to make sure files are where the prompt expects and look the way the prompt assumes.
4. **Simplify the prompt** — remove parts until it works, then add pieces back to find the part that breaks.
5. **Check the run history** for patterns — did it start failing on a specific date? Did something change on your computer around that time?

If none of this works, copy the prompt, the error message, and a description of what you expected, and bring that package to whoever supports your setup. That information will help them diagnose the issue far faster than "my report is broken."

What Dr. Mitchell Learned

Dr. Mitchell now has five scheduled tasks running in the Claude Desktop app: a daily appointment preview, a Monday weekly summary, a Friday billing check, and two monthly reports. Together, these save her roughly four hours per week.

Her biggest lesson: start with one task, get it running reliably for two or three cycles, and then add the next. Trying to schedule everything at once makes troubleshooting much harder because you cannot tell which task is causing which problem. Build your automation habit one scheduled task at a time.

Her second lesson: check your computer's energy settings. She lost a week of morning reports before realizing her laptop was going to sleep at midnight and not waking up until she opened it at 8 AM — an hour and a half after her 6:30 AM task was supposed to run. Once she adjusted her power settings to keep the machine awake during business hours, every task ran like clockwork.

Scaling Your Automations

From One-Off Tasks to a System

By now, you have built several automations in Cowork. You have cleaned up data, generated reports, organized files, and maybe scheduled a task or two. But right now, these automations probably exist as scattered conversations in the Claude Desktop app — some you remember, some you have already forgotten about.

This chapter is about turning your collection of one-off wins into a proper system — a personal automation library that grows more valuable over time and that you (and eventually your team) can rely on.

Dr. Mitchell hit this inflection point about six weeks in. She had built a dozen useful automations, but she could not remember the exact prompt she used for her quarterly billing report. She spent twenty minutes scrolling through old Cowork conversations trying to find it, and when she did, she realized she had improved it twice but could not tell which version was the best one. That frustration is what led her to build her personal automation library.

How to Build a Personal Automation Library

A personal automation library is a well-organized collection of your best prompts and Cowork sessions, saved where you can find them quickly. It does not require special software. A folder on your computer with clearly named text files will do — and since Cowork works directly with your local files, keeping your library on your machine means Cowork itself can reference it.

Step 1: Create a Dedicated Folder

Create a folder called "My Automations" in a place you will remember. Inside it, create subfolders by category. For example: Reports, Data Cleanup, File Management, Communications, and Analysis.

Your categories should reflect how you think about your work. A realtor might use "Listings," "Client Follow-ups," and "Market Reports." A lawyer might use "Case Summaries," "Document Review," and "Billing." The best system matches how your brain already organizes things.

Step 2: Save Each Automation as Its Own File

For each automation you build in Cowork, create a text file with these sections:

Name: A clear, descriptive title. "Weekly Patient Summary Report" is good. "Report v2" is bad.

What It Does: One or two sentences for yourself six months from now. "Pulls the past 7 days of patient visit data, groups visits by type, highlights follow-up flags, and saves a formatted PDF."

The Prompt: The exact text you use in Cowork to make this automation run. Copy it verbatim from the conversation or scheduled task where it worked — do not edit from memory.

Notes (optional): Anything you learned while building it. "Works best when the CSV has headers in row 1." "The Gmail connector needs to be authorized before this runs." These notes are gold when you come back months later.

For scheduled tasks, also record the schedule itself: "Runs every Monday at 6:30 AM" and any dependencies like "requires the daily visit log task to have run first."

Step 3: Version Your Prompts

When you improve an automation, do not overwrite the old prompt. Keep both versions and mark which is current. Add the date to filenames: "Weekly Patient Summary - 2025-03-15.txt" and "Weekly Patient Summary - 2025-04-02.txt." The most recent is the current version; older ones are your backup.

Dr. Mitchell once "improved" her billing reconciliation prompt and did not realize until month-end that the new version silently excluded a category of charges. Because she had saved the previous version, she rolled back immediately instead of reconstructing the original from memory. She updated her scheduled task with the old prompt, ran a test, and was back to correct output in under five minutes.

Stacking Automations: When One Task's Output Feeds the Next

The most powerful setups are chains where the output file from one scheduled Cowork task becomes the input for another. This is called stacking, and it is how you move from saving minutes to saving hours.

Here is an example from Dr. Mitchell's practice:

Scheduled Task 1 (Daily, 6 PM): Cowork reads today's patient visit data and saves it as a clean CSV in the "Daily Logs" folder.

Scheduled Task 2 (Weekly, Monday 6:30 AM): Cowork reads the past week's daily CSV files from the "Daily Logs" folder and generates a Weekly Patient Summary report, saved to the "Weekly Reports" folder.

Scheduled Task 3 (Monthly, 1st of month at 7 AM): Cowork reads the month's weekly summaries from the "Weekly Reports" folder and produces a Monthly Dashboard with trends and comparisons, saved to the "Monthly Reports" folder.

Each scheduled task builds on the one before it. The daily task creates clean raw material, the weekly task assembles it into something meaningful, and the monthly task synthesizes a strategic view. All of this happens automatically in the Claude Desktop app — you just review the final output.

How to Set Up a Stack

1. Start from the bottom. Build the scheduled task that collects or cleans raw data first. Get it running reliably for at least a full week before adding anything on top.

2. Define clear handoff points. Each task must save its output file in a specific location with a consistent naming convention. Be explicit in your prompt: "Save as daily-visits-[date].csv in /Documents/Practice/Daily Logs/." The next task in the chain will look for files in that exact location using that exact pattern.

3. Build the next layer once the first is running reliably. The prompt for the second task should reference the exact folder and filename pattern the previous task uses. For example: "Read all CSV files in /Documents/Practice/Daily Logs/ that are dated within the past 7 days."

4. Stagger the schedules. If your daily task runs at 6 PM and your weekly task runs Monday at 6:30 AM, there is plenty of time for the daily task to finish first. Leave generous gaps between tasks in a stack. Remember that both tasks need the computer awake and the Claude Desktop app open at their scheduled times.

5. Troubleshoot from the bottom up. When a stacked automation produces wrong results, the problem might be in an earlier task. Always check the raw output files first — open the Daily Logs folder and verify that the daily CSVs look correct. Then check each layer's output folder in order until you find where the data goes wrong.

Saving and Reusing Your Best Prompts

Some prompts are remarkably versatile. A prompt that summarizes patient data can be adapted for sales data or case files with minor changes. These are your power prompts — the ones worth saving as reusable templates.

The Template Approach

Turn your best prompts into templates by replacing specifics with placeholders:

Original: "Look at the patient visit log in /Documents/Practice/visits.csv. Pull all visits from the past 7 days. Create a summary grouped by visit type, showing totals and the three busiest days."

Template: "Look at [DATA FILE PATH]. Pull all [RECORDS] from the past [TIME PERIOD]. Create a summary grouped by [CATEGORY FIELD], showing totals and the [TOP N] busiest [TIME UNIT]."

Now you can reuse this for sales transactions, billable hours, support tickets, or anything else. Save templates alongside your specific automations, labeled clearly: "TEMPLATE - Weekly Summary by Category." When you need a new automation, check your templates first — adapting an existing one takes five minutes versus the thirty it takes to write from scratch.

Building a Prompt Playbook

Once you have ten or more saved prompts, organize them into a playbook grouped by problem type: data summaries, file organization, communication drafts, schedule management. When you face a new challenge, check the playbook first. Most new tasks are variations of something you have already solved.

You can even store your playbook in a folder that Cowork can access. When you need a new automation, tell Cowork: "Look at the templates in /Documents/My Automations/Templates/ and adapt the weekly summary template for my sales data in /Documents/Sales/transactions.csv." Cowork reads your template, understands the pattern, and produces a tailored version.

How to Share Automations with a Team

Plain-English prompts are inherently shareable. You do not need to be a programmer to understand what someone else's automation does — you just read it.

Start with a Shared Folder

Create a shared folder on your team's drive (or use a shared Google Drive folder via Cowork's connector) and organize it the same way as your personal library — by category, with clear names

and descriptions.

Write Prompts for Others

When saving a prompt you plan to share, add context:

- **Who this is for:** "For any team member who needs to generate the weekly client activity report."
- **What you need first:** "Download the latest client-activity.csv from the CRM and save it to /Documents/Shared/Data/."
- **What to change:** "Replace [YOUR NAME] with your name and update the file path to match your local folder structure."
- **What the output looks like:** "A one-page PDF with a table and three charts."

Establish Team Conventions

If multiple people build automations, agree on basics early: consistent naming formats, a shared folder structure, clear ownership of each automation, and a rule that shared prompts must be tested within the past month. Without these conventions, you end up with five versions of the same report and nobody knows which one is current.

Dr. Mitchell's practice now has a shared library of twelve automations that any staff member can use in their own Cowork setup. New hires learn to use these in their first week, and the entire office runs more efficiently because everyone follows the same tested prompts.

What to Automate Next: A Framework for Finding High-Value Tasks

You have the skills. The question is: what should you automate next?

The Automation Value Matrix

Ask two questions about any repetitive task:

1. **How much time does it take?** (Minutes per occurrence times frequency)
2. **How much does it require your judgment?** (Could a competent assistant do it with clear instructions?)

High time, low judgment — automate first. Data entry, report formatting, file organization, routine emails, invoice generation, status update compilation. These are perfect candidates for Cowork

scheduled tasks.

High time, high judgment — automate partially. Automate the data gathering and formatting, keep the analysis for yourself. You cannot automate the decision to refer a patient to a specialist, but you can have a Cowork task pull together their history and test results so you decide faster.

Low time, low judgment — automate if easy. If a five-minute setup eliminates a two-minute daily task, that is still a net gain after five days.

Low time, high judgment — leave alone. Some tasks are quick and important precisely because they require your expertise.

The 2-Hour Rule

Here is the simplest way to find your next target: if you spend more than 2 hours per week on any single repetitive task, automate it. Two hours per week is over 100 hours per year — two and a half full work weeks on something a scheduled Cowork task could handle.

To apply this, spend one week tracking your time. A sticky note on your monitor is fine. Every time you do a repetitive task, write what it was and roughly how long it took. At the end of the week, add up the time for each task. Anything over two hours goes on your automation list.

Dr. Mitchell's time audit revealed she was spending 3 hours per week on report formatting, 2.5 hours on email triage, 2 hours on chart note summaries, and 1 hour on appointment prep. She automated all four using Cowork scheduled tasks. Total time reclaimed: eight and a half hours per week — over 440 hours per year, the equivalent of eleven full work weeks.

Your Automation Roadmap

After your time audit, rank candidates by time saved and work down the list. Build one automation per week in Cowork, get it reliable, add the prompt to your library, then move on. At this pace, you will have a dozen solid automations within three months, permanently eliminating the busywork that used to dominate your evenings.

Revisit your time audit every quarter. As your role evolves, new repetitive tasks emerge. The framework stays the same: track your time, find the two-hour-plus tasks, and automate them with Cowork.

Going Further: When You Outgrow the Basics

At some point, you may want automations that go beyond what Cowork's built-in scheduling and connectors can handle — pulling data from financial APIs, coordinating multiple steps across different services, or building a custom dashboard from scratch. This is where Claude Code shines.

You do not need to learn to code. You describe what you want in plain English, and Claude Code writes the scripts, connects the APIs, and builds the infrastructure. For example, one user asked Claude Code to build a weekly financial summary that pulls transaction data from their bank, categorizes spending, and emails a formatted digest every Sunday evening. Claude Code wrote the entire pipeline — data retrieval, analysis, formatting, and email delivery — in a single conversation. The user never touched a line of code.

If you reach this level, think of it as graduating from managing one assistant (Cowork) to managing a team: Cowork handles the routine scheduled work while Claude Code builds custom tools for anything more complex. The key insight stays the same: you describe the problem, the AI builds the solution.

The Big Picture

What you have built is not a collection of tricks. It is a capability — the ability to identify repetitive work, describe it clearly, and hand it off to Cowork to execute reliably on a schedule. That capability compounds over time. Every automation frees up time and mental energy for higher-value work — the work that actually requires your expertise, your judgment, and your human relationships.

You now have every tool you need. The Claude Desktop app is installed, Cowork is set up, your connectors are linked, and your automation library is organized. The question is not whether you can automate your work — you have already proven that you can. The question is how much of your time you are willing to keep spending on tasks that do not need you.

Appendix

A. Full Prompt Template Library

Below are 25 profession-specific prompts you can copy, paste, and use immediately. Replace anything in **[BRACKETS]** with your own information. Each prompt is designed to work with Cowork or Claude directly.

Healthcare

1. Patient Research Digest

```
Review the following patient case notes and recent lab work. Summarize the key findings in plain language, flag anything outside normal ranges, and list recommended follow-up actions.

Patient ID: [PATIENT_ID_OR_INITIALS]
Condition(s) being monitored: [PRIMARY_CONDITIONS]
Recent lab results: [PASTE_LAB_VALUES_OR_ATTACH_FILE]
Current medications: [LIST_CURRENT_MEDICATIONS]
Date range to review: [START_DATE] to [END_DATE]

Format the output as:
- One-paragraph clinical summary
- Bulleted list of abnormal findings
- Recommended next steps with urgency level (routine, soon, urgent)
```

2. Lab Results Queue

```
I have [NUMBER] sets of lab results that need review. For each patient, compare the new results against their previous values and flag any significant changes.
```

```
Patient list:
- [PATIENT_1_NAME]: [TEST_TYPE], [DATE]
- [PATIENT_2_NAME]: [TEST_TYPE], [DATE]
- [PATIENT_3_NAME]: [TEST_TYPE], [DATE]

Previous baseline values are attached as [FILE_NAME_OR_PASTE_HERE].

For each patient, provide:
1. Values that moved outside normal range
2. Values that changed more than [PERCENTAGE]% from baseline
3. A one-sentence interpretation
4. Whether follow-up is needed (yes/no and why)
```

3. Prior Authorization Tracking

```
Create a prior authorization status tracker from the following information. I need to
see what is pending, what has been approved, and what needs resubmission.

Current authorizations:
- [PATIENT_NAME], [PROCEDURE/MEDICATION], submitted [DATE], insurance: [CARRIER]
- [PATIENT_NAME], [PROCEDURE/MEDICATION], submitted [DATE], insurance: [CARRIER]
- [PATIENT_NAME], [PROCEDURE/MEDICATION], submitted [DATE], insurance: [CARRIER]

For each entry, generate a row with:
| Patient | Procedure | Insurance | Date Submitted | Days Pending | Status | Next Action

Flag anything pending longer than [NUMBER] business days. Sort by urgency.
```

4. CME Tracking

```
Help me track my continuing medical education credits. My license renewal date is
[RENEWAL_DATE] and I need [TOTAL_CREDITS_REQUIRED] credits by then.

Credits completed so far:
- [COURSE_NAME], [CREDITS_EARNED], [DATE_COMPLETED], [CATEGORY]
- [COURSE_NAME], [CREDITS_EARNED], [DATE_COMPLETED], [CATEGORY]
- [COURSE_NAME], [CREDITS_EARNED], [DATE_COMPLETED], [CATEGORY]
```

```
Category requirements: [LIST_ANY_CATEGORY_MINIMUMS]

Provide:
1. Total credits earned vs. required
2. Credits remaining by category
3. Monthly pace needed to finish on time
4. A list of gaps (categories where I am behind)
```

5. Schedule Gap Analysis

```
Analyze my clinic schedule for [DATE_RANGE] and identify inefficiencies.

Schedule data: [PASTE_SCHEDULE_OR_ATTACH_CSV]
Appointment types: [LIST_TYPES, e.g., NEW_PATIENT, FOLLOW_UP, PROCEDURE]
Ideal appointment lengths: [TYPE_1: MINUTES, TYPE_2: MINUTES]
Buffer time between appointments: [MINUTES]

Report back:
- Total open slots and when they occur
- Double-bookings or overlaps
- Days that are overbooked vs. underbooked
- Suggestions for rearranging to reduce gaps
```

Legal

6. Case Research Summary

```
Research the following legal topic and provide a structured summary I can use for case
preparation.

Topic: [LEGAL_ISSUE_OR_QUESTION]
Jurisdiction: [STATE_OR_FEDERAL_JURISDICTION]
Relevant area of law: [PRACTICE_AREA]
Key facts of my case: [BRIEF_FACT_PATTERN]

Provide:
```

```
1. Overview of the relevant legal standard (2–3 paragraphs)
2. Key statutes or rules that apply, with citations
3. A list of leading cases with one-sentence summaries
4. How the law likely applies to my fact pattern
5. Any counterarguments to anticipate

Note: This is for research assistance only, not final legal advice.
```

7. Contract Clause Extractor

```
Review the attached contract and extract the following clauses. For each one, quote the
relevant language and flag any terms that are unusual or potentially unfavorable.

Contract: [PASTE_TEXT_OR_DESCRIBE_FILE]
Contract type: [LEASE / EMPLOYMENT / VENDOR / PARTNERSHIP / OTHER]

Clauses to extract:
- Termination provisions
- Indemnification
- Limitation of liability
- Non-compete or non-solicitation
- Confidentiality
- Governing law and dispute resolution
- [ANY_ADDITIONAL_CLAUSES]

For each clause, provide:
1. Exact quoted language
2. Plain-English summary
3. Risk flag (standard / review recommended / potentially unfavorable)
```

8. Billing Hours Digest

```
Summarize my billable hours for [TIME_PERIOD]. I need a client-ready report and an
internal summary.

Time entries: [PASTE_ENTRIES_OR_ATTACH_FILE]
Billing rate: $[RATE]/hour
Client: [CLIENT_NAME]
```

```
Matter: [MATTER_NAME_OR_NUMBER]

Generate:
1. Total hours and total fees
2. Breakdown by task category (research, drafting, correspondence, court, meetings)
3. A brief narrative summary suitable for including on an invoice
4. Any entries that seem too short or too long to be accurate (flag for my review)
```

9. Client Follow-Up Tracker

```
Create a follow-up tracker for my active clients. For each client, tell me what is
overdue, what is due this week, and what is coming up.

Client list:
- [CLIENT_NAME]: last contact [DATE], pending items: [DESCRIBE]
- [CLIENT_NAME]: last contact [DATE], pending items: [DESCRIBE]
- [CLIENT_NAME]: last contact [DATE], pending items: [DESCRIBE]

My follow-up policy: contact clients at least every [NUMBER] days on active matters.

Output a table:
| Client | Matter | Last Contact | Days Since Contact | Overdue? | Next Action | Due By |
```

10. Deadline Calendar Builder

```
Build a deadline calendar from the following case information. Include all statutory
deadlines, filing deadlines, and internal preparation deadlines.

Case: [CASE_NAME_AND_NUMBER]
Jurisdiction: [JURISDICTION]
Case type: [CIVIL / CRIMINAL / FAMILY / OTHER]
Key dates:
- Filing date: [DATE]
- Service date: [DATE]
- Trial date (if set): [DATE]
- [ANY_OTHER_KEY_DATES]

Applicable rules: [FEDERAL_RULES / STATE_RULES / LOCAL_RULES]
```

```
For each deadline, provide:
| Deadline | Date | Rule/Statute | Days From Today | Preparation Start Date | Notes |

Sort chronologically. Flag anything due within 14 days.
```

Finance / Accounting

11. Client Portfolio Summary

```
Summarize the current state of [CLIENT_NAME]'s portfolio for a quarterly review meeting.

Holdings: [PASTE_HOLDINGS_OR_ATTACH_FILE]
Benchmark: [S&P_500 / CUSTOM_BENCHMARK]
Review period: [START_DATE] to [END_DATE]
Client risk tolerance: [CONSERVATIVE / MODERATE / AGGRESSIVE]

Provide:
1. Total portfolio value and period return (%)
2. Performance vs. benchmark
3. Asset allocation breakdown (stocks, bonds, cash, alternatives)
4. Top 3 gainers and top 3 losers
5. Any positions that have drifted outside the target allocation by more than [PERCENTAGE
6. Recommended talking points for the client meeting
```

12. Tax Document Checklist

```
Generate a personalized tax document checklist for [CLIENT_NAME] based on their
situation.

Filing status: [SINGLE / MFJ / MFS / HOH]
Income sources: [W2 / 1099 / BUSINESS / RENTAL / INVESTMENTS / OTHER]
Deductions typically claimed: [MORTGAGE / CHARITABLE / BUSINESS_EXPENSES / OTHER]
State(s): [STATE(S)_OF_RESIDENCE]
Tax year: [YEAR]
```

```
Create a checklist with:
- [ ] Document name
- Where to get it (employer, bank, brokerage, etc.)
- Typical arrival date
- Whether we have received it yet: [YES / NO / N/A]

Group by category (income, deductions, credits, state-specific).
```

13. Monthly P&L Digest

```
Analyze the attached profit and loss statement and provide an executive summary my
client can understand without an accounting background.

P&L data: [PASTE_OR_ATTACH_FILE]
Business type: [INDUSTRY / BUSINESS_TYPE]
Comparison period: [PRIOR_MONTH / SAME_MONTH_LAST_YEAR / BOTH]

Provide:
1. Revenue: total and trend (up/down/flat vs. comparison)
2. Top 3 expense categories and whether they grew or shrank
3. Net income and margin
4. Any line items that changed more than [PERCENTAGE]% from the comparison period
5. Two or three plain-English observations (e.g., "Payroll costs rose 12% — worth reviewi
6. Questions the business owner should be asking
```

14. Invoice Aging Report

```
Create an accounts receivable aging report from the following invoice data.

Invoices: [PASTE_DATA_OR_ATTACH_CSV]
Today's date: [DATE]
Payment terms: [NET_30 / NET_60 / CUSTOM]

Generate a table:
| Client | Invoice # | Amount | Date Issued | Days Outstanding | Bucket (Current/30/60/90

Then provide:
- Total AR by aging bucket
```

```
- Clients with the largest overdue balances
- Suggested collection actions for anything over [NUMBER] days
```

15. Regulatory Update Tracker

```
Summarize recent regulatory changes that affect my [INDUSTRY] clients.

Industry: [INDUSTRY]
Jurisdiction: [STATE / FEDERAL / BOTH]
Time period: [LAST_30_DAYS / LAST_QUARTER / CUSTOM]
Topics of interest: [TAX_LAW / REPORTING / COMPLIANCE / LABOR / OTHER]

For each update, provide:
| Regulation/Rule | Effective Date | Summary | Impact (High/Med/Low) | Action Required |

Flag anything with a compliance deadline in the next 90 days.
```

Real Estate

16. Listing Research Digest

```
Compile a research digest for a property I am listing (or considering listing).

Property address: [ADDRESS]
Property type: [SINGLE_FAMILY / CONDO / MULTI_FAMILY / COMMERCIAL]
Bedrooms/Bathrooms: [BED/BATH]
Square footage: [SQFT]
Lot size: [LOT_SIZE]
Key features: [LIST_NOTABLE_FEATURES]

Provide:
1. Recent comparable sales (last [NUMBER] months, within [RADIUS] miles)
2. Active competing listings in the same area and price range
3. Average days on market for similar properties
```

```
4. Suggested list price range with justification
5. Market trend summary (buyer's market, seller's market, balanced)
```

17. Client Follow-Up Sequence

```
Write a follow-up email sequence for a [BUYER / SELLER] client at the following stage.

Client name: [NAME]
Stage: [INITIAL_INQUIRY / POST_SHOWING / POST_OFFER / UNDER_CONTRACT / POST_CLOSING]
Property (if applicable): [ADDRESS_OR_DESCRIPTION]
Key details: [ANY_RELEVANT_CONTEXT]

Write [NUMBER] emails spaced [NUMBER] days apart. Each email should:
- Have a subject line
- Be [CASUAL / PROFESSIONAL / FORMAL] in tone
- Include a clear call to action
- Be under [NUMBER] words
```

18. Market Comp Report

```
Build a comparative market analysis for [ADDRESS].

Property details: [BEDS, BATHS, SQFT, LOT_SIZE, YEAR_BUILT, CONDITION]
Comparison area: [NEIGHBORHOOD / ZIP_CODE / RADIUS]
Time frame for comps: last [NUMBER] months

Include:
- At least [NUMBER] comparable sales
- For each comp: address, sale price, price/sqft, beds/baths, days on market, sale date
- Adjusted value analysis (adjustments for size, condition, features)
- Suggested market value range
- A one-paragraph summary suitable for a client presentation
```

19. Transaction Deadline Tracker

```
Create a transaction timeline and deadline tracker for the following deal.
```

```
Property: [ADDRESS]
Contract date: [DATE]
Closing date: [DATE]
Buyer/Seller: [MY_ROLE]
Key contingency dates:
- Inspection deadline: [DATE]
- Appraisal deadline: [DATE]
- Financing contingency: [DATE]
- [OTHER_DEADLINES]

Generate a table:
| Milestone | Date | Days Away | Status (Pending/Complete/Overdue) | Notes |

Sort chronologically. Flag anything due within 7 days.
```

20. Lead Scoring Summary

```
Review my current leads and help me prioritize follow-up.

Lead list:
- [NAME], source: [REFERRAL / WEBSITE / OPEN_HOUSE / AD], first contact: [DATE],
  budget: [RANGE], timeline: [TIMEFRAME], notes: [BRIEF_NOTES]
- [NAME], source: [SOURCE], first contact: [DATE], budget: [RANGE],
  timeline: [TIMEFRAME], notes: [NOTES]

Score each lead on:
1. Readiness (1–5): how soon they are likely to transact
2. Motivation (1–5): how engaged they have been
3. Fit (1–5): does their budget match available inventory

Output a ranked list with scores and a recommended next action for each.
```

General Business

21. Weekly KPI Digest

```
Create a weekly KPI report from the following data.

Business: [BUSINESS_NAME]
Reporting period: [WEEK_START] to [WEEK_END]

Metrics:
- Revenue: $[AMOUNT] (last week: $[AMOUNT])
- New customers/clients: [NUMBER] (last week: [NUMBER])
- [KPI_3_NAME]: [VALUE] (last week: [VALUE])
- [KPI_4_NAME]: [VALUE] (last week: [VALUE])
- [KPI_5_NAME]: [VALUE] (last week: [VALUE])

Provide:
1. A 3-sentence executive summary
2. Each KPI with week-over-week change (%) and a trend arrow description (up/down/flat)
3. The single most important thing to focus on this week
4. Any metric that moved more than [PERCENTAGE]% — explain possible causes
```

22. Employee Hours Summary

```
Summarize employee hours for [TIME_PERIOD] from the attached timesheet data.

Timesheet data: [PASTE_OR_ATTACH_FILE]
Pay period: [WEEKLY / BIWEEKLY / MONTHLY]
Overtime threshold: [NUMBER] hours per [WEEK / PERIOD]
Employees to include: [ALL / LIST_SPECIFIC_NAMES]

Provide:
| Employee | Regular Hours | Overtime Hours | Total Hours | Notes |

Then summarize:
- Total labor hours across the team
- Employees approaching or exceeding overtime
- Any gaps or inconsistencies in the data (missed punches, unusual entries)
```

23. Vendor Invoice Tracker

```
Organize and track my vendor invoices for [MONTH/PERIOD].

Invoice data:
- [VENDOR_NAME], Invoice #[NUMBER], $[AMOUNT], dated [DATE], due [DATE]
- [VENDOR_NAME], Invoice #[NUMBER], $[AMOUNT], dated [DATE], due [DATE]
- [VENDOR_NAME], Invoice #[NUMBER], $[AMOUNT], dated [DATE], due [DATE]

Generate:
| Vendor | Invoice # | Amount | Date | Due Date | Days Until Due | Status (Paid/Due/Overd

Then provide:
- Total payable this period
- Cash flow note: total due in the next 7 / 14 / 30 days
- Any invoices that seem duplicated or unusual
```

24. Meeting Notes to Action Items

```
Convert the following meeting notes into a structured action item list.

Meeting: [MEETING_NAME_OR_TOPIC]
Date: [DATE]
Attendees: [LIST_NAMES]

Notes:
[PASTE_RAW_MEETING_NOTES_HERE]

For each action item, extract:
| # | Action Item | Assigned To | Due Date | Priority (High/Med/Low) | Status |

Also provide:
- A 3-sentence meeting summary
- Key decisions that were made
- Open questions that still need answers
```

25. Competitor Monitoring Digest

```
Create a competitor monitoring summary based on the following information.

My business: [YOUR_BUSINESS_DESCRIPTION]
Competitors to track:
- [COMPETITOR_1_NAME]: [WEBSITE_OR_NOTES]
- [COMPETITOR_2_NAME]: [WEBSITE_OR_NOTES]
- [COMPETITOR_3_NAME]: [WEBSITE_OR_NOTES]

Areas to monitor: [PRICING / PRODUCT_LAUNCHES / HIRING / MARKETING / REVIEWS / ALL]
Time period: [LAST_WEEK / LAST_MONTH / CUSTOM]

For each competitor, provide:
1. Notable recent activity
2. Potential impact on my business (high/medium/low)
3. Suggested response or action

End with a one-paragraph competitive landscape summary.
```

B. Cowork Quick Reference

Everything you need to get started with Cowork, the desktop app for knowledge workers.

Starting a Session

Open the Cowork app on your computer. Click **New Session** to begin. Type your prompt in the chat window and press Enter. Cowork will process your request and display the results in the same window. You can continue the conversation by sending follow-up messages.

Connecting a Local Folder

Click the **Connect Folder** button in the sidebar (or use the settings menu). Browse to the folder on your computer that contains the files you want Cowork to access. Once connected, Cowork can read and write files in that folder. Reference files by name in your prompts — for example, "read the file march-expenses.csv."

Connecting Google Drive

Open **Settings > Connectors** and select **Google Drive**. Sign in with your Google account and grant Cowork permission to access your files. Once connected, you can reference Google Drive documents directly in your prompts.

Scheduling a Task

Use the `/schedule` command in any session to set up a recurring or one-time task. For example:

- `/schedule daily 9am` — runs the task every day at 9 AM
- `/schedule every friday 4pm` — runs every Friday at 4 PM
- `/schedule once tomorrow 8am` — runs one time tomorrow morning

After entering the command, Cowork will ask you to confirm the prompt and schedule. You can view and manage all scheduled tasks in **Settings > Scheduled Tasks**.

Other Useful Actions

Action	How to Do It
Resume a previous session	Open the **Sessions** panel in the sidebar and click on the session you want to continue.
Change the model	Open **Settings > Model** and select Sonnet (faster) or Opus (more capable).
Stop a running task	Click the **Stop** button in the active session, or close the session.
View scheduled tasks	Go to **Settings > Scheduled Tasks** to see all active and paused schedules.
Export results	Copy the output from the session window, or use the **Export** button to save results to your connected folder.

Tip: You can type `/help` in any session to see all available commands.

C. Glossary

Fifteen terms you will encounter in this guide, explained without jargon.

API (Application Programming Interface) A way for two pieces of software to talk to each other. When Cowork connects to Claude, it uses an API. You do not need to build an API yourself — you

just need an account, which handles authentication automatically.

Automation Setting up a process so it runs on its own, without you clicking buttons or doing manual work every time. In this guide, automation means using Cowork to handle repetitive tasks like sending summaries, processing files, or generating reports.

Claude Code Anthropic's web-based coding tool for building software, automating code-heavy workflows, and working with GitHub repositories. Use Claude Code when your task involves writing or modifying code. For everything else, use Cowork.

Claude Desktop App The application you install on your Mac or Windows computer to run Cowork. Download it from Anthropic's website, sign in, and you are ready to go. No other software is required.

Cloud Environment A computing space hosted on remote servers rather than your local machine. Claude Code runs in a cloud environment, meaning your code projects are processed on Anthropic's servers. Your local files stay on your computer and are accessed through Cowork.

Connector A link between Cowork and an external service like Google Drive, a local folder, or another application. Connectors let Cowork read and work with your data wherever it lives. You set them up once in Settings.

Cowork Anthropic's desktop app for knowledge workers. You type prompts in a chat-style interface, connect your files and folders, and Cowork handles tasks like generating reports, analyzing data, drafting documents, and running scheduled jobs. No technical background required.

CSV (Comma-Separated Values) A simple file format for spreadsheet data. Each line is a row, and commas separate the columns. You can export CSVs from Excel, Google Sheets, QuickBooks, and most business software. Cowork can read and analyze CSV files directly.

Model The AI brain behind Claude. Different models have different capabilities and speeds. Claude Sonnet is faster and cheaper; Claude Opus is more capable and thorough. You choose which one to use based on your task.

Prompt The instructions you give to Claude. A good prompt is specific, includes context, and tells Claude exactly what format you want the answer in. The templates in this appendix are all prompts.

Scheduled Task A job that runs automatically at a set time. In Cowork, you can create scheduled tasks using the `/schedule` command — daily, weekly, or at whatever interval you choose. No technical knowledge required.

Session A single conversation thread in Cowork. Each session keeps its own context and history. You can have multiple sessions open and return to previous ones from the Sessions panel in the sidebar.

Script A file containing a set of instructions that a computer follows step by step. In this guide, scripts are usually short text files that tell Cowork what to do. You write them once and reuse them.

Token The unit Claude uses to measure text. Roughly speaking, one token is about three-quarters of a word. Tokens determine how much a request costs and how much text Claude can process at once. A typical business email is around 200-400 tokens.

D. Troubleshooting Quick Reference

Ten common problems and how to fix them.

Problem 1: Cowork will not open or crashes on launch.

Solution: Make sure you are running the latest version of the Cowork desktop app. Check for updates in the app menu or download the latest version from Anthropic's website. If the app still will not open, try restarting your computer. On Mac, you can also try deleting the app's cache by going to ~/Library/Caches and removing the Cowork folder, then relaunching.

Problem 2: "Authentication failed" or sign-in issues.

Solution: Your account credentials may have expired. Sign out of Cowork and sign back in. If you use a team or enterprise account, check with your administrator to make sure your access is still active. Make sure your computer is connected to the internet, as Cowork needs an active connection to authenticate.

Problem 3: Claude gives a generic or unhelpful response.

Solution: Your prompt is too vague. Add more context: tell Claude your role, what the data is, what format you want, and what the output is for. Compare "summarize this" (vague) with "summarize

this CSV of client invoices into a table showing who owes what, sorted by amount, and flag anything overdue" (specific). Use the templates in Section A of this appendix as a starting point.

Problem 4: A scheduled task did not run at the expected time.

Solution: Check three things. First, make sure your computer was turned on and awake at the scheduled time — if your computer was asleep or shut down, the task will not fire until the next scheduled time. Second, open **Settings > Scheduled Tasks** and verify the task is enabled (not paused). Third, confirm the schedule is set correctly — check whether the time is AM or PM, and make sure the correct days are selected.

Problem 5: Cowork cannot find or read my files.

Solution: Make sure you have connected the correct folder. Go to **Settings > Connected Folders** and verify the folder path. If the folder is on an external drive, make sure the drive is connected. Check that the files you are referencing actually exist in the connected folder — Cowork can only access files in folders you have explicitly connected. Also check file permissions: right-click the file in Finder, select Get Info, and make sure your user account has read access.

Problem 6: Claude's response gets cut off in the middle.

Solution: The response exceeded the maximum token limit. Break your request into smaller pieces. Instead of asking Claude to process a 500-row spreadsheet all at once, process it in batches of 100 rows. You can also ask Claude to be more concise: add "keep the response under 1000 words" to your prompt.

Problem 7: Google Drive connector is not working.

Solution: Go to **Settings > Connectors > Google Drive** and check the connection status. If it shows as disconnected, click **Reconnect** and sign in again with your Google account. Make sure you have granted Cowork the necessary permissions to access your files. If you recently changed your Google password or enabled two-factor authentication, you may need to reconnect.

Problem 8: The output format is not what I expected (e.g., I wanted a table but got paragraphs).

Solution: Be explicit about the format in your prompt. Add a line like "Format the output as a markdown table with these columns: Name, Amount, Status" or "Return the result as a numbered list." Claude follows formatting instructions well when they are stated clearly. Including an example of the desired output in your prompt helps even more.

Problem 9: "Context window exceeded" or "input too long" error.

Solution: You are sending more text than Claude can process in a single request. Reduce the input size by summarizing the data before sending it, or split the work into multiple smaller requests. For large files, extract only the relevant sections rather than sending the entire document.

Problem 10: Cowork is running slowly.

Solution: Several things can cause slowness. If you are using the Opus model, switch to Sonnet for faster responses on simpler tasks (change this in **Settings > Model**). Close other heavy applications to free up system resources. If you are processing large files, consider whether you can filter the data down before sending it to Claude. A poor internet connection can also cause delays since Cowork communicates with Anthropic's servers over the internet.